



Universidad de las Ciencias
Informáticas

Facultad 4

**Módulo para la integración de un servidor FTP con el
Sistema de Gestión para el Ingreso a la Educación
Superior (SIGIES)**

**Trabajo de Diploma para optar por el Título de Ingeniero en
Ciencias Informáticas**

Autor:

Jeanne Morejón Cruz

Tutores:

Ing. Jorge Luis Piña González

Ing. Susana Becerra Rodríguez

La Habana, 2017

“Año 58 de la Revolución”

Declaración de autoría

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas (UCI) a que haga el uso que estime pertinente con el mismo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Autor:

Jeanne Morejón Cruz

Tutor(es):

Ing. Susana Becerra Rodriguez

Ing. Jorge Luis Piña González

Pensamiento



“En la vida tienes que ir tras las oportunidades para las cuales estás mejor equipado.”

Soichiro Honda

Dedicatoria

A mis padres: Mercedes y Andres por estar siempre brindándome su amor y apoyo incondicional, desde el primer día que vine al mundo, hoy estoy súper agradecido por la forma que me criaron, por los valores, que como flores en mi pecho sembraron.

A mi hermana Llilianne por estar siempre a mi lado en las buenas y en las malas, mostrándome que la familia es lo más importante y que debemos siempre estar unidos.

A mis sobrinos: Yidiel y Yilier que saben cuánto los amo y el día que le toque a ustedes yo seré la persona más feliz del mundo.

A toda la familia en general que de una forma u otra siempre se han preocupado por mis estudios, hoy se recoge el fruto de tantos años de esfuerzo.

A todas aquellas personas que me han brindado su apoyo y ayuda durante estos años, sin ustedes no lo hubiese logrado.

Agradecimientos

A mi mamá agradecerle, por todo su esfuerzo y dedicación, por su amor infinito, gracias por todo, por existir y formar parte de mi vida te quiero mima.

A mi papá, no sé si me alcance la vida para agradecerte por todo lo que has hecho por mí, todos esos sacrificios para que yo pueda ser realidad mis mayores sueños, lo que he logrado hoy te lo debo a ti, gracias papá.

A mi hermana, por formar parte de mi vida, sin duda lo más valioso de ella, te quiero tata.

A toda la familia, mis sobrinos, que los quiero muchísimos.

A toda la familia mi abuela Marta, mis tías y tíos, mis primos, a todos en general.

A los amigos del barrio Yoni, Yuniel, Jose y todos los que siempre me apoyaron.

A mis amigos de la universidad que durante estos cinco años he podido contar con su apoyo y amistad.

A los que se han convertido en parte de mi familia Oscar, Adrian, Javier, Manuel, Ernesto, Carrion, Carlos, Yuniel y Daniel.

A los tutores y a todos aquellos que de una manera u otra me apoyaron en la realización de este trabajo.

Resumen

La informatización es un hecho innegable, que avanza de manera ordenada en Cuba sobre bases seguras y sostenibles. Cuba está consciente de que una sociedad para ser más eficaz, eficiente y competitiva debe aplicar la informatización en todas sus esferas y procesos. En este proceso hacia la informatización, la Universidad de las Ciencias Informáticas (UCI), encargada de producir aplicaciones y servicios informáticos, se sitúa a la vanguardia. Actualmente en esta institución se desarrolla el Sistema para la Gestión del Proceso de Ingreso a la Educación Superior (SIGIES). Este sistema presenta la necesidad de contar con un espacio en el cual se pueda almacenar la información generada durante la realización del proceso. Por esta razón, la presente investigación propone desarrollar un módulo para integrar un servidor FTP al sistema SIGIES. Con esta propuesta se pretende que los usuarios, mediante técnicas FTP, accedan de forma ágil y segura a la información almacenada en el servidor. Para guiar el desarrollo del módulo en cuestión se utilizó la metodología AUP-UCI, generándose los artefactos fundamentales que propone la metodología para cada etapa de trabajo. Se utilizó los lenguajes HTML 5, CSS 3 y JavaScript apoyado de herramientas como Visual Paradigm 8.0 para el modelado UML y PhpStorm 8.0 como entorno integrado de desarrollo. Finalmente desarrollado el producto, las pruebas aplicadas evidenciaron que existe una correspondencia satisfactoria entre el objetivo y los resultados obtenidos, así como una alta satisfacción por parte del cliente.

Palabras clave: integración, servidor FTP, Sistema de Gestión.

Índice

Introducción.....	1
Capítulo 1: Fundamentación teórica.....	5
Introducción.....	5
1.1 Conceptos asociados.....	5
Servidor.....	5
FTP (<i>File Transfer Protocol</i>).....	5
Servidor FTP.....	5
Cliente FTP.....	5
Cliente FTP basado en Web.....	6
1.2 Funcionamiento del protocolo FTP.....	6
1.2.1 Tipos de acceso al servicio FTP.....	7
1.2.2 Modos de conexión del cliente.....	8
1.2.3 Funcionalidades estándares del servicio FTP.....	10
1.3 Estudio de sistemas similares.....	10
1.4.1 Metodología de desarrollo de software.....	12
1.4.2 Lenguajes de desarrollo.....	15
1.4.3 Framework de desarrollo.....	18
1.4.4 Sistema Gestor de Bases de Datos.....	19
1.4.5 Cliente de base de datos.....	20
1.4.6 Servidor Web.....	20
1.4.7 Entorno integrado de desarrollo.....	21
1.4.8 Herramientas CASE.....	22
Conclusiones parciales.....	23
Capítulo 2: Análisis y diseño.....	24
Introducción.....	24
2.1 Modelo de negocio.....	24
2.2 Modelo de dominio.....	25
2.2.1 Conceptos del dominio.....	25
2.3 Descripción de la propuesta solución.....	26
2.4 Requerimientos del sistema.....	28
2.4.1 Requerimientos funcionales.....	28
2.4.2 Requerimientos no funcionales.....	29
2.5 Descripción de requisitos por procesos.....	30
2.6 Modelo de análisis.....	32

2.6.1 Diagrama de clases del análisis.....	32
2.6.2 Diagrama de colaboración del análisis.....	33
2.7 Modelo de diseño.....	34
2.7.1 Patrón de arquitectura.....	34
2.7.2 Patrones de diseños.....	35
2.8 Modelo de datos.....	38
2.8.1 Diseño de la base de datos.....	38
2.8.2 Descripción de las tablas en la base de datos.....	38
Conclusiones parciales.....	39
Capítulo 3: Implementación y prueba.....	40
Introducción.....	40
3.1 Modelo de implementación.....	40
3.1.1 Diagrama de componente.....	40
3.2 Pruebas de software.....	41
3.2.1 Niveles de Prueba.....	41
3.2.2 Métodos de prueba.....	42
3.2.3 Diseño de casos de prueba.....	43
3.2.4 Resultado de las pruebas.....	45
Conclusiones parciales.....	48
Conclusiones generales.....	49
Referencias bibliográficas.....	51
Anexo 1: Descripción de requisitos por procesos.....	57
Anexo 2: Diagramas de clases del análisis.....	72
Anexo 3: Diagramas de colaboración del análisis.....	75
Anexo 4: Diagramas de clases del diseño.....	79
Anexo 5: Diagramas de secuencias del diseño.....	84
Anexo 6: Diseños de casos de pruebas.....	88

Introducción

Introducción

Hoy en día no se puede hablar de eficiencia o novedad si no existe una aplicación y correcta utilización de la tecnología moderna, y es que se puede asegurar que las Tecnologías de la Información y las Comunicaciones (TIC) tienen la respuesta efectiva a la mano.

Las nuevas tecnologías han afectado todas las áreas del desarrollo social y se han convertido en elementos inherentes al desarrollo en todas las esferas de la vida. En el campo de la educación, su impacto cubre numerosas áreas de aplicación, desde la educación a distancia hasta la utilización de terminales portátiles y herramientas informáticas en las escuelas. Otro efecto muy importante es el impacto de internet, el cual puede compensar problemas como el bajo nivel de capacitación de los educadores o la falta de material educativo (Katz, 2009).

Tomando en consideración esa realidad mundial, se han desarrollado en Cuba múltiples programas encaminados a lograr la informatización del país. Estos están relacionados con el proceso de utilización ordenada y masiva de las TIC para satisfacer las necesidades de información y conocimiento de la sociedad. En este proceso hacia la informatización del país, la Universidad de las Ciencias Informáticas (UCI) se sitúa a la vanguardia, formando profesionales comprometidos con su Patria y altamente calificados en la rama de la Informática.

En la Universidad existen varios centros de producción, como el Centro de Tecnologías para la Formación (FORTES), perteneciente a la facultad 4. Este centro tiene como misión: desarrollar tecnologías que permitan ofrecer servicios y productos con el fin de brindar soluciones de formación aplicando las TIC, a todo tipo de instituciones con diferentes modelos de formación y condiciones tecnológicas.

Actualmente, en este centro se está desarrollando un Sistema de Gestión para el Ingreso a la Educación Superior (SIGIES) para el Ministerio de Educación Superior (MES). Dicho sistema, permite gestionar los subprocesos de Organización, Exámenes, Asignación y Otorgamiento comprendidos en el proceso de Ingreso a la Educación Superior. Durante la realización de estos subprocesos, el sistema genera reportes personalizados que permiten obtener un mayor control y seguimiento del Ingreso a la Educación Superior. Estos reportes deben ser almacenados por un periodo de tiempo definido por la institución. Por ejemplo, si un estudiante que vive en la ciudad de Pinar del Río, por motivos personales culmina los estudios en la provincia de Villa Clara, durante el proceso de Ingreso a la Educación Superior estas comisiones necesitan estar en un constante intercambio de información. Es necesario que la comisión de Pinar del Río le reporte a la de Villa Clara todos los estudiantes que se encuentran en esa situación y a la vez la comisión

Introducción

de Villa Clara les brinde la información referente a los estudiantes en el proceso de Exámenes y Asignación. Hoy la limitante que posee el sistema es que no cuenta con un espacio en el cual una comisión tenga acceso de manera ágil y segura a la información generada por las demás comisiones del país, por lo que se emplean algunos mecanismos como el correo electrónico. Esto dificulta el proceso de comunicación, pues cada vez que ocurra una modificación o se actualice la información se envían varios correos para las comisiones implicadas. Además, esta alternativa atenta contra la seguridad y confiabilidad de la información.

Al analizar la problemática anteriormente planteada se define el siguiente **problema científico**: Inexistencia de un espacio dentro del sistema en el cual se pueda compartir la información relacionada con los diferentes procesos del Ingreso a la Educación Superior.

Se define como **objeto de estudio**, integración de servicios FTP a sistemas web quedando enmarcado el **campo de acción** en la integración de un servidor FTP al Sistema de Gestión para el Ingreso a la Educación Superior.

Se establece como **objetivo general**: desarrollar un módulo para integrar un servidor FTP al Sistema de Gestión para el Ingreso a la Educación Superior.

Para lograr el objetivo general propuesto se han definido los siguientes **objetivos específicos**:

- Analizar elementos teóricos y principales tendencias del desarrollo web en la actualidad en el campo de la integración de sistemas.
- Implementar las funcionalidades del módulo para la integración de los servicios FTP al SIGIES.
- Realizar las pruebas necesarias para garantizar la funcionalidad y calidad del módulo.

A partir de los objetivos definidos, se derivan las siguientes **tareas** a cumplir:

- Análisis de soluciones similares existentes.
- Revisión bibliográfica para fundamentar la selección de la metodología de desarrollo.
- Selección de las técnicas y tecnologías a emplear en el desarrollo de la solución.
- Análisis y diseño de la solución propuesta.
- Identificación de requisitos funcionales y no funcionales.
- Definición de los patrones de diseño a utilizar en el desarrollo del módulo.

Introducción

- Elaboración de los artefactos necesarios según la metodología seleccionada para el desarrollo de la propuesta de solución.
- Implementación de la solución propuesta.
- Estudio de los diferentes tipos de pruebas para validar la calidad y el correcto funcionamiento del módulo.

La presente investigación sustenta la **hipótesis** siguiente:

Si se integra al Sistema de Gestión para el Ingreso a la Educación Superior un servidor FTP, permitirá obtener un espacio para que los usuarios compartan la información referente a los procesos realizados durante el Ingreso a la Educación Superior.

Obteniendo como **posible resultado**:

- Un espacio dentro del sistema en el cual se pueda tener acceso de manera rápida a los recursos generados en el proceso de Ingreso a la Educación Superior.
- La correcta documentación del proceso y del módulo a desarrollarse, de manera que pueda continuarse con el mantenimiento de este, así como el desarrollo de futuras versiones.

Métodos científicos:

La investigación científica es la búsqueda intencionada de conocimientos o de soluciones a problemas de carácter científico. El método científico indica el camino que se ha de transitar en esa indagación y las técnicas precisan la manera de recorrerlo. A continuación, se citan los métodos teóricos y empíricos seleccionados:

Métodos teóricos:

- **Analítico-sintético:** permitirá estudiar y realizar el análisis de documentos y bibliografías que sustenten tanto de forma teórica como práctica los elementos asociados a la integración del sistema SIGIES, identificando los elementos más importantes y necesarios para dar solución al problema planteado.
- **Inductivo-deductivo:** se utilizará para arribar a conclusiones que permitan implementar la solución, partiendo de la información adquirida en el estudio del estado del arte.
- **Modelación:** se utilizará para realizar los diagramas necesarios que modelarán la solución.

Métodos empíricos:

Introducción

- **Entrevista:** se utilizará para identificar las necesidades presentes en el sistema SIGIES y de esta manera definir el objetivo que estará dirigido a la propuesta de solución.

El presente trabajo está estructurado de la siguiente forma:

Capítulo 1: Fundamentación Teórica

En este capítulo se exponen los elementos teóricos utilizados en la investigación. Se realiza un análisis del estado del arte y se describe las tecnologías, metodologías, herramientas y los lenguajes de programación utilizados en el desarrollo de la solución, así como los principales conceptos involucrados.

Capítulo 2: Análisis y diseño

En este capítulo se describen el proceso de análisis y diseño, se exponen los elementos que permiten describir la propuesta de solución para lograr una mayor comprensión de las funcionalidades a desarrollar. También se detalla la arquitectura de la solución y sus principales características.

Capítulo 3: Implementación y prueba

Se realiza la implementación de todas las funcionalidades identificadas, logrando un módulo que satisface las principales necesidades del cliente. Además, se detallan las pruebas realizadas una vez finalizada la implementación, para asegurar que este cumple con las especificaciones, calidad y eficiencia requeridas.

Capítulo 1: Fundamentación teórica

Capítulo 1: Fundamentación teórica

Introducción

Para la fundamentación del presente trabajo en este capítulo se exponen los conceptos necesarios para una mayor comprensión del contenido. Se realiza un estudio que permita conocer las funcionalidades que sirvan de aporte a la solución. Se explica la metodología, los lenguajes y las herramientas a utilizar.

1.1 Conceptos asociados

A continuación, se definen los términos más significativos relacionados con el objeto de estudio.

Servidor

Los servidores son máquinas que actúan como "almacenes" de información. Esta información es solicitada por los ordenadores-cliente, y el servidor responde a tales peticiones devolviendo los datos solicitados (Morera, 2000).

FTP (*File Transfer Protocol*)

Protocolo de Transferencia de Archivos, es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP/IP, basado en la arquitectura cliente-servidor. Desde un equipo cliente se puede conectar a un servidor para descargar archivos desde él o para enviarle archivos, independientemente del sistema operativo utilizado en cada equipo, esta conexión se realiza a través de los puertos 20 y 21 (Noite.pl, 2010).

Servidor FTP

Un servidor FTP es un programa que se ejecuta en un servidor conectado normalmente en Internet (aunque puede estar conectado en otros tipos de redes, LAN¹, MAN² y WAN³). La función del mismo es permitir el desplazamiento de datos entre diferentes servidores u ordenadores (Morera, 2000).

Cliente FTP

Un cliente FTP es un programa que se instala en el ordenador del usuario, que emplea el protocolo FTP para conectarse a un servidor FTP y transferir archivos, ya sea para descargarlos o para subirlos. Para utilizar un cliente FTP, se necesita conocer el nombre del archivo, el ordenador en que reside (servidor, en el caso de descarga de archivos), el ordenador al que se

1 LAN (Red de Área Local)

2 MAN (Red de Área Metropolitana)

3 WAN (Wide Area Network)

Capítulo 1: Fundamentación teórica

quiere transferir el archivo (en caso de querer subir un archivo propio al servidor), y la carpeta en la que se encuentra (Andreu, 2011).

Cliente FTP basado en Web

Un "cliente FTP basado en WEB" no es más que un Cliente FTP al cual se puede acceder a través del Navegador Web, sin necesidad de tener otra aplicación para ello. El usuario accede a un servidor web que lista los contenidos de un servidor FTP. El usuario se conecta mediante el protocolo http a un servidor web, el cual se conecta mediante el protocolo FTP al servidor FTP. El servidor web actúa de intermediario haciendo pasar la información desde el servidor FTP en los puertos 20 y 21 hacia el puerto 80 http que ve el usuario (Andreu, 2011).

1.2 Funcionamiento del protocolo FTP

El protocolo FTP permite definir la manera en que los datos deben ser transferidos a través de una red TCP/IP. El objetivo principal de este servicio es permitir que equipos remotos puedan compartir información, posibilitando de esta forma, la independencia entre los sistemas de archivos del equipo cliente y los del servidor. Durante una conexión FTP, se encuentran abiertos dos canales de transmisión: un canal de comandos (canal de control) y el de datos como se muestra en la siguiente imagen.



Fig 1: Funcionamiento del protocolo FTP

Capítulo 1: Fundamentación teórica

Por tanto, el cliente y el servidor cuentan con dos procesos que permiten la administración de estos dos tipos de información:

- El Proceso Transferencia de Datos (DTP, por sus siglas en inglés *Process Transfer of Data*), es el encargado de establecer la conexión y administrar el canal de datos. El DTP del lado del servidor se denomina Servidor de DTP y del lado del cliente se denomina Usuario de DTP.
- El Intérprete de Protocolo (PI, por sus siglas en inglés *Protocol Interpreter*), interpreta el protocolo y permite que el DTP pueda ser controlado mediante los comandos recibidos a través del canal de control.

Estos procesos son diferentes en el cliente y en el servidor:

- El Servidor PI es responsable de escuchar los comandos que provienen de un Usuario PI a través del canal de control en un puerto de datos. Además, se encarga de establecer la conexión para el canal de control, recibir los comandos FTP del Usuario PI a través de este, emitir una respuesta y ejecutar el Servidor DTP.
- El Usuario PI es responsable de establecer la conexión con el servidor FTP, enviar los comandos FTP, recibir respuestas del Servidor PI y controlar al Usuario DTP, si fuera necesario.

De forma general, cuando un cliente FTP se conecta con un servidor FTP, el Usuario PI inicia la conexión con el servidor de acuerdo con el protocolo Telnet. El cliente envía comandos FTP al servidor, este los interpreta, ejecuta su DTP y después envía una respuesta. Una vez que se establece la conexión, el Servidor PI proporciona el puerto a donde se enviarán los datos al Usuario DTP y este escucha por el puerto especificado, los datos provenientes del servidor.

1.2.1 Tipos de acceso al servicio FTP

- Acceso anónimo:

Los servidores FTP anónimos ofrecen sus servicios libremente a todos los usuarios, permiten acceder a sus archivos sin necesidad de tener una cuenta que los identifique. Es la manera más cómoda fuera del servicio web de permitir que todo el mundo tenga acceso a cierta información sin que para ello el administrador de un sistema tenga que crear una cuenta para cada usuario (Gómez 2010).

Sin embargo, al acceder de esta forma se tendrán menos privilegios que al acceder como un usuario normal. Lo recomendable es que al acceder de forma anónima al servidor solo se puedan leer y copiar los archivos que sean públicos.

Capítulo 1: Fundamentación teórica

➤ Acceso de usuario:

Si se desea tener privilegios de acceso a cualquier parte del sistema de archivos del servidor FTP, de modificación de datos existentes y de posibilidad de subir archivos propios, generalmente se suele realizar mediante una cuenta de usuario. En el servidor se guarda la información de las distintas cuentas que pueden acceder a él, de manera que para iniciar una sesión FTP se debe introducir un nombre de usuario y una contraseña que identifique unívocamente (Gómez 2010).

➤ Acceso de invitado:

El acceso sin restricciones al servidor que proporcionan las cuentas de usuario implica problemas de seguridad, lo que ha dado lugar a un tercer tipo de acceso FTP denominado invitado (guest). Se puede contemplar como una mezcla de los dos anteriores, le permite a cada usuario conectarse a la máquina mediante su usuario y su contraseña. Evita que se tenga acceso a partes del sistema de archivos que no necesita para realizar su trabajo. Esto es muy similar a lo que sucede en los accesos anónimos, pero con más privilegios (Gómez 2010).

1.2.2 Modos de conexión del cliente

➤ Modo activo

En modo activo, el servidor siempre crea el canal de datos en su puerto 20, mientras que en el lado del cliente el canal de datos se asocia a un puerto aleatorio mayor que el 1024. Para ello, el cliente manda un comando PORT al servidor por el canal de control indicándole ese número de puerto, de manera que el servidor pueda abrirle una conexión de datos por donde se transferirán los archivos, en el puerto especificado. Esto trae como consecuencia un grave problema de seguridad, es que la máquina cliente debe estar dispuesta a aceptar cualquier conexión de entrada en un puerto superior al 1024. De hecho, los cortafuegos que se instalen en el equipo para evitar ataques seguramente rechazarán esas conexiones aleatorias. Para solucionar esto se desarrolló el modo pasivo (Gómez 2010).

Capítulo 1: Fundamentación teórica

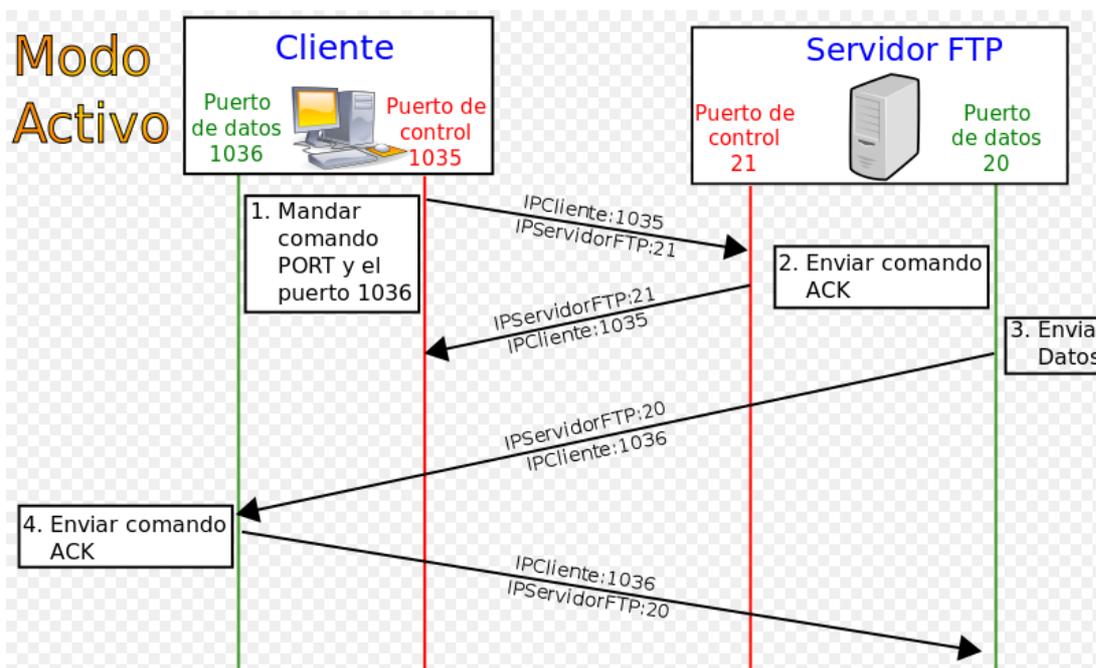


Fig 2: Conexión de forma activa al servidor FTP

➤ Modo pasivo

Cuando el cliente envía un comando PASV 2 sobre el canal de control, el servidor FTP indica por dicho canal, el puerto (mayor a 1023) al que debe conectarse el cliente. El cliente inicia una conexión desde el puerto siguiente al de control, hacia el puerto del servidor especificado anteriormente. La siguiente figura modela el proceso.

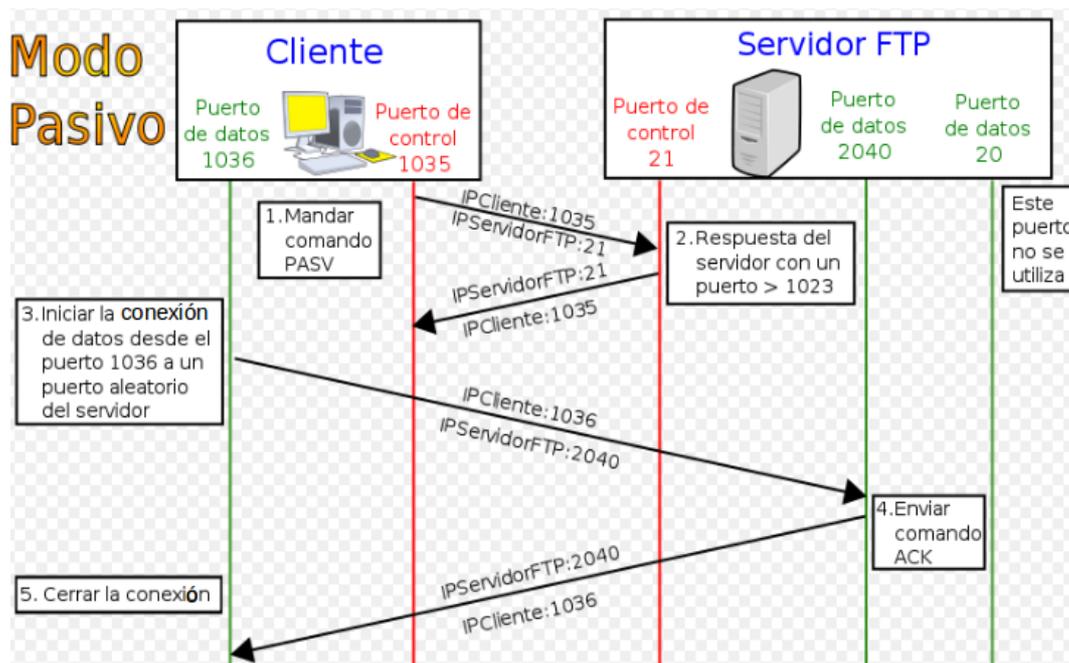


Fig 3: Conexión de forma pasiva al servidor FTP

Capítulo 1: Fundamentación teórica

Antes de cada transferencia tanto en el modo activo como en el pasivo, el cliente debe enviar otra vez un comando de control (PORT o PASV, según el modo en el que haya conectado). El servidor recibirá esa conexión de datos en un nuevo puerto aleatorio (si está en modo pasivo) o por el puerto 20 (si está en modo activo) (Gómez 2010).

1.2.3 Funcionalidades estándares del servicio FTP

- Permitir o no el acceso de usuarios locales al servidor.
- Permitir o no el acceso de usuarios anónimos al servidor.
- Establecer permisos por tipos de usuarios.
- Restringir el ancho de banda para la transferencia de datos.
- Restringir la concurrencia en el servidor teniendo en cuenta la cantidad de clientes e IP conectados de forma simultánea.
- Restringir la cantidad de intentos fallidos por sesión.
- Brindar soporte opcional para una capa de cifrado SSL/TLS.
- Gestionar registros en el servidor.

Para la gestión del contenido existente en el servidor FTP, hay programas específicos con muchas opciones, pero realmente es mucho más sencillo usar un Navegador. En dependencia del lenguaje utilizado estará disponible un conjunto de funciones que implementan el acceso por parte del cliente mediante técnicas FTP.

1.3 Estudio de sistemas similares

Con el objetivo de obtener conocimiento y hallar una solución al problema existente fue necesario realizar un estudio de las tendencias actuales sobre integración de sistemas que existen. A continuación, se expone el resultado del estudio realizado.

Una de las carencias más graves que tenían los grupos de **Facebook**⁴, uno de los sitios web más visitados en internet, es que no se podían compartir archivos. Hoy es posible debido a la integración con Dropbox, compañía que brinda servicios de almacenamiento online.

Es evidente que la integración de servicios FTP a un sistema web ha resultado una solución factible para algunas de las grandes compañías en el mundo. En la actualidad existen muchos sistemas que permiten gestionar el contenido alojado en un servidor FTP, algunos de ellos implementados en el lenguaje PHP.

⁴ <https://www.facebook.com/>

Capítulo 1: Fundamentación teórica

NET2FTP es un cliente de FTP basado en Web, es decir, puede ser utilizado mediante cualquier navegador. Este cuenta con todas las funciones típicas de cualquier otro cliente FTP, tales como subir, copiar, renombrar, cambiar permisos, crear directorios, entre otras. El sistema centra toda su atención en lo que constituye su función principal, instalar en su servidor un script que servirá para instalar algunos de los CMS más utilizados (Joomla, WordPress, Moodle, Mambo, TikiWiki y otros). De esta manera el proceso resultará más sencillo sin tener que buscar y subir usted mismo los paquetes de instalación. A la vez esto resulta muy interesante porque ningún cliente FTP de escritorio lo hace. La principal limitante que posee este cliente-FTP basado en la web es que no permite establecer conexión con varios FTP para la búsqueda y almacenamiento de la información.

NgExplorer agrupa todos los sitios FTP que conozcas en un solo lugar, teniendo así la capacidad de buscar información en cada uno de ellos (o en todos al mismo tiempo) de una forma sencilla y agradable.

Este cliente-FTP permite navegar por los directorios de los FTP registrados como un explorador de archivos habitual. Permite personalizar el modo de vista de los directorios, ir al directorio superior, recargar el directorio, guardar el historial de navegación por cada pestaña, selección de los archivos a una lista para descarga. También se calcula de ser posible el tamaño y la última modificación de un directorio si fue visitado. Además, busca coincidencias mediante filtros en uno o varios proveedores a la vez, accede al directorio donde se encuentra el archivo y realizar la descarga del mismo. También permite gestionar los proveedores en el sistema (adicionar, eliminar, modificar, escanear y sincronizar proveedores), se establecen las configuraciones del proxy de ser necesario y gestión de la cuenta de usuario administrador del sistema.

Como se puede observar, el ngExplorer es un sistema que solo se centra en la búsqueda y descarga de la información en cada uno de los FTP registrados, no permitiendo, si los usuarios poseen los privilegios, subir un archivo propio al servidor FTP seleccionado.

En conclusión, una vez analizados los sistemas anteriores se determina que aunque presentan características y funcionalidades similares al objetivo buscado, no utilizar ninguno de ellos, pues no cubren las necesidades específicas detectadas para el SIGIES. Como estos sistemas no dan solución al problema planteado en esta investigación, se decide implementar un módulo que cubra las necesidades detectadas y sea capaz de satisfacer los requerimientos del negocio exigidos por el cliente.

Capítulo 1: Fundamentación teórica

Hoy en día, es fácil encontrar bibliotecas que brinden ciertas facilidades a la hora de implementar una aplicación. Para la conexión con un servidor FTP se puede hacer uso de varias bibliotecas o bundles que se encargan de establecer dicha conexión. A continuación, se mencionan algunas a tener presente para el desarrollo del módulo.

IJanki/FtpBundle: Un paquete de Symfony2 para envolver la funcionalidad de extensión PHP FTP. La versión estable 2.0 requiere de PHP 5.5.9 o superior. Permite establecer la conexión vía ssl o no, mediante el modo pasivo o activo utilizando el puerto 21 por defecto. Posee un conjunto de funcionalidades predefinidas que permiten ahorrar tiempo durante el desarrollo de un producto de software, dentro de las cuales se puede destacar la función *connectUrl()* que permite interpretar y validar los datos de una conexión. Otras funcionalidades permiten conocer si un recurso es un directorio o un archivo, el tamaño del mismo y cuando fue modificado por última vez.

Touki653/FTPBundle: este es un paquete de Symfony2 para envolver el componente FTP. Requiere PHP 5.3 o superior. Aunque no posee un conjunto de funcionalidades tan amplia como el anterior, este permite configurar los parámetros de conexión: host, usuario, password, modo pasivo y el puerto mediante el cual se establece la conexión.

Para el desarrollo de la propuesta solución se hará uso del IJanki/FtpBundle. Este cuenta con un conjunto de funcionalidades básicas para el trabajo con el servidor FTP, ya implementadas, lo que permite acortar el tiempo de desarrollo del módulo a desarrollar.

1.4 Metodología, tecnologías y herramientas

A continuación, se describe la metodología, las tecnologías y las herramientas a emplear en el desarrollo e implementación de la solución propuesta.

1.4.1 Metodología de desarrollo de software

Las metodologías de desarrollo de software abarcan todo el ciclo de vida del mismo, y se definen como “un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo software” (Castellanos, 2009). El objetivo de las metodologías es guiar a un equipo de proyecto durante la creación de un nuevo software. Estas se clasifican en dos grandes grupos:

Metodologías tradicionales: son orientadas al control de los procesos; establecen rigurosamente las actividades a desarrollar, herramientas a utilizar y notaciones que se usarán (Falgueras, 2002).

Capítulo 1: Fundamentación teórica

Metodologías ágiles: son orientadas a la interacción con el cliente y el desarrollo incremental del software; mostrando versiones parcialmente funcionales del software al cliente en determinados intervalos de tiempo, para que pueda evaluar y sugerir cambios en el producto (Cobb, 2011).

Entre las metodologías fundamentales con enfoque ágil utilizadas en el mundo para el proceso de desarrollo de software se encuentra Programación Extrema o *Extreme Programming* (XP), *Microsoft Solution Framework* (MSF) y el Proceso Unificado Ágil o *Agile Unified Process* (AUP).

AUP constituye una versión simplificada del Proceso Unificado Racional (*Rational Unified Process*, RUP), desarrollada por Scott Ambler. Esta metodología combina procesos propios del concepto unificado tradicional con técnicas ágiles, con el objetivo de mejorar la productividad. Permitiendo así describir de manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio. A esta metodología la Universidad de las Ciencias Informáticas (UCI) decidió hacerle una variación y ahora es AUP-UCI. Con esta adaptación de AUP se logra estandarizar el proceso de desarrollo de software, dando cumplimiento además a las buenas prácticas que define CMMI-DEV v1.3. También permite que se adapte el ciclo de vida definido para la actividad productiva de la institución.

Siguiendo la política de desarrollo de software de la institución, se define como metodología a emplear la AUP-UCI en el escenario número tres. Por ser la metodología y el escenario empleados en el desarrollo del sistema al cual será integrado el módulo en cuestión. A continuación, se describe la metodología.

AUP-UCI

Al no existir una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, entre otros) exigiéndose así que el proceso sea configurable. La universidad decide hacer una variación de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI.

Fases que propone AUP

- Inicio
- Elaboración
- Construcción
- Transición

Fases que propone AUP-UCI

Capítulo 1: Fundamentación teórica

Se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, pero modificando el objetivo de la misma. Se unifican las restantes 3 fases de AUP en una sola, llamada Ejecución y se agrega la fase de Cierre.

Disciplinas que propone AUP

- Modelo
- Implementación
- Prueba
- Despliegue
- Gestión de configuración
- Gestión de proyecto
- Entorno

Disciplinas que propone AUP-UCI

Se decide para el ciclo de vida de los proyectos de la UCI tener 7 disciplinas también, pero a un nivel más atómico que el definido en AUP. Los flujos de trabajos: Modelado de negocio, Requisitos y Análisis y diseño en AUP están unidos en la disciplina Modelo, en la variación para la UCI, se consideran a cada uno de ellos disciplinas. Se mantiene la disciplina Implementación, en el caso de Prueba se desagrega en 3 disciplinas: Pruebas Internas, de Liberación y Aceptación. Las restantes 3 disciplinas de AUP asociadas a la parte de gestión para la variación UCI se cubren con las áreas de procesos que define CMMI- DEV v1.3 para el nivel 2, serían CM (Gestión de la configuración), PP (Planeación de proyecto) y PMC (Monitoreo y control de proyecto).

Roles que propone AUP

- Administrador de proyecto
- Ingeniero de procesos
- Desarrollador
- Administrador de Bases de Datos
- Modelador ágil
- Administrador de la configuración
- Stakeholder
- Administrador de pruebas
- Probador

Roles que propone AUP-UCI

Capítulo 1: Fundamentación teórica

Se decide para el ciclo de vida de los proyectos de la UCI tener 11 roles, manteniendo algunos de los propuestos por AUP y unificando o agregando otros.

Esta versión AUP-UCI, define cuatro escenarios en los que se puede ubicar el desarrollo de una aplicación de acuerdo a sus características, los cuales son:

- Escenario 1: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que puedan modelar una serie de interacciones entre los trabajadores del negocio/actores del sistema (usuario), similar a una llamada y respuesta.
- Escenario 2: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que no es necesario incluir las responsabilidades de las personas que ejecutan las actividades, de esta forma modelarían exclusivamente los conceptos fundamentales del negocio.
- Escenario 3: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio con procesos muy complejos, independientes de las personas que los manejan y ejecutan, proporcionando objetividad, solidez, y su continuidad.
- Escenario 4: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio muy bien definido. El cliente estará siempre acompañando al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos. Se recomienda en proyectos no muy extensos.

1.4.2 Lenguajes de desarrollo

Los lenguajes de programación fueron creados por el hombre con el fin de modelar y hacer en las computadoras programas que simulen la realidad en la que se desarrolla la sociedad. Actualmente existen diferentes lenguajes de programación para desarrollar aplicaciones de escritorio como aplicaciones para la web (Pierce, 2002). A continuación, se describen los lenguajes usados, que han sido definidos por el proyecto al cual se integrará el módulo en cuestión.

Lenguajes del lado del cliente

Los lenguajes del lado del cliente son aquellos que se utilizan para la creación de páginas web y son totalmente independientes del servidor. Son muy utilizados en la capa de presentación o las vistas de los usuarios (Vázquez, 2006).

Capítulo 1: Fundamentación teórica

HTML: es el lenguaje con el que se escriben las páginas web (*Hyper Text Markup Language* por sus siglas en inglés HTML). Es un estándar que define una estructura básica y un código (denominado código HTML) para la definición de contenido en una página web, como texto, imágenes, videos y juegos. Los navegadores se encargan de interpretar el código HTML de los documentos, y de mostrar a los usuarios las páginas web resultantes del código interpretado (Lancker,2012).

En el desarrollo de la solución propuesta se emplea HTML5, el cual establece una serie de nuevos elementos y atributos que reflejan el uso típico de los sitios web modernos. Algunos de ellos son técnicamente similares a las etiquetas <div> y pero tienen un significado semántico como por ejemplo la etiqueta <nav>(bloque de navegación del sitio). También incorpora nuevos tipos de datos (ejemplo: email, number, url y datetime) y facilidades para validar sin tener que usar javascript en la realización de formularios. En general contiene un conjunto más amplio de tecnologías que permiten a las aplicaciones ser más diversas y de gran alcance.

JavaScript: es un lenguaje de programación utilizado principalmente para crear páginas web dinámicas. Incorpora efectos tales como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. JavaScript es un lenguaje interpretado, por tanto, no es necesario compilar programas para ejecutarlo (McFarland, 2008).

CSS: es un lenguaje de hojas de estilos, creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. Es la mejor forma de separar los contenidos y su presentación, y es indispensable para crear páginas web complejas (García, 2012). CSS3, versión utilizada para el desarrollo del módulo, posee como principales características mayor control sobre el estilo de los elementos de la página web y mayor número de efectos visuales (CSS3, 2016).

Ventajas de usar CSS3:

- Código más simple para muchas tareas.
- Mayores opciones de gráfica.
- CSS3 es más utilizado por los usuarios.

Lenguaje de programación del lado del servidor

Los lenguajes son ejecutados por el servidor y lo que se envía al cliente es la respuesta o el resultado de dicha ejecución.

Capítulo 1: Fundamentación teórica

PHP: es uno de los lenguajes más utilizados en el desarrollo de aplicaciones web. Lenguaje gratuito, rápido e independiente de plataforma, con una gran biblioteca de funciones y mucha documentación. PHP se escribe dentro del código HTML, esto lo hace realmente fácil de utilizar, brinda muchas ventajas como su independencia de plataforma, rapidez y seguridad (Muñoz, 2013).

Más de 250 millones de sitios web se han realizado en PHP, entre ellos algunos tan famosos como: Facebook.com, Yahoo INC, Wikipedia.org, Friendster.com, Digg.com, Sourceforge.org, Flickr.com y Meneame.net. También es utilizado por CMS como WordPress y Drupal. PHP está desarrollado en política de código abierto, esto lo convierte en el idóneo para usar como lenguaje de programación al desarrollar el módulo que integre los servicios FTP con SIGIES. A continuación, se ofrecen algunas de las mejoras presentes en la versión 7.0:

- Importante reducción del uso de memoria.
- Árbol sintáctico abstracto.
- Soporte consistente a los 64-bits.
- Un buen número de errores fatales convertidos en excepciones.
- Mejora en la jerarquía de estas excepciones.
- Generador de números aleatorio más seguro.
- Operador de coalescencia con null.
- Clases anónimas.

Funciones presentes en PHP para conectarse al servidor FTP

- ftp_connect — Abre una conexión FTP.
- ftp_ssl_connect — Abre una conexión segura SSL-FTP.
- ftp_close — Cierra una conexión FTP.
- ftp_login — Inicia sesión en una conexión FTP.
- ftp_put — Carga un archivo al servidor FTP.
- ftp_pwd — Devuelve el nombre del directorio actual.
- ftp_get — Descarga un archivo desde el servidor FTP.
- ftp_delete — Elimina un archivo en el servidor FTP.
- ftp_rmdir — Elimina un directorio.
- ftp_rename — Renombra un archivo o un directorio en el servidor FTP.
- ftp_size — Devuelve el tamaño del archivo dado.

Capítulo 1: Fundamentación teórica

Lenguaje de modelado

UML (*Unified Modeling Language*) o Lenguaje Unificado de Modelado, es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo). Incluye aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. Se utiliza para definir un sistema de software y detallar sus artefactos y documentos (Rumbaugh, 2007).

Este lenguaje de modelado visual posibilita identificar los sistemas, y puede ser utilizado para identificar "lo que" se necesita de un sistema, y el "cómo", podría ponerse en marcha o aplicarse. Además, ayuda a planear un programa antes que la programación se lleva a cabo. Algunas de las herramientas utilizadas para modelar UML, generan su código basado en las clases establecidas en el modelo. Esto puede ayudar a reducir los gastos en la etapa de ejecución de cualquier programa. Además, un diagrama de modelo UML es fácil de cambiar, esto ayuda considerablemente en el desarrollo de cualquier software actual. Es utilizado en el desarrollo de la investigación para el modelado de algunos diagramas que permiten una mejor comprensión de la misma (Fowler y Scott, 1999).

1.4.3 Framework de desarrollo

Un framework es un esquema o patrón que provee funcionalidades genéricas. Estas pueden ser utilizadas con el fin de desarrollar aplicaciones de manera rápida, fácil, modular y sencilla, ahorrando tiempo y esfuerzo. Es posible que esta tecnología defina una estructura para una aplicación completa, o solo se centre en un aspecto de la misma (Hernández, 2016). A continuación, se describe el framework utilizado para la implementación de la solución propuesta.

Symfony: es uno de los frameworks PHP más utilizado entre los usuarios y las empresas. Permitiéndole a los programadores ser más productivos y a la vez crear código de calidad y fácil de mantener. Este framework está diseñado para optimizar el desarrollo de las aplicaciones, separando la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo, automatizando las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

Capítulo 1: Fundamentación teórica

Symfony está desarrollado completamente con PHP y ha sido probado con éxito en sitios como Yahoo, Answers, Delicious y DailyMotion. Es compatible con la mayoría de los gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas *nix (Unix y Linux.) como en plataformas Windows (Hernández, 2016). Para el desarrollo de la solución propuesta se utiliza la versión 2.7, esta es la empleada en el desarrollo del sistema al cual se le integrará el módulo en cuestión. Además, es lo suficientemente estable como para desarrollar aplicaciones a largo plazo.

1.4.4 Sistema Gestor de Bases de Datos

Un Sistema Gestor de Bases de Datos (SGBD) o *Data Base Management System* (DBMS), se define como una colección de datos relacionados entre sí, estructurados y organizados, permitiéndole a un conjunto de programas acceder y gestionar esos datos. Esta colección se denomina Base de Datos, (DB por sus siglas en inglés *Data Base*) (MIGUEL, 2011). A continuación, se describe el sistema gestor utilizado en el desarrollo del módulo.

PostgreSQL: desarrollado por *PostgreSQL Global Development Group* (PGDG), es multiplataforma y posee una licencia libre permisiva. Permite no solo la implementación de bases de datos relacionales, también simula las multidimensionales. Se puede manejar mediante un cliente de entorno gráfico PGAdmin o mediante un cliente en forma de terminal de texto llamado PSQL, el cual es muy efectivo para servidores con pocos recursos de hardware (Cornelio, Rivas y Hernández 2004). Además, puede ser usado perfectamente por sistemas donde exista alta concurrencia de usuarios accediendo y donde se manejen grandes volúmenes de datos.

Características:

- Incluye varios métodos para el manejo de índices.
- Es multiplataforma y permite a los usuarios definir su propio tipo de datos, así como la ejecución de consultas complejas, consultas sobre vistas, subconsultas y uniones de gran tamaño.
- Utiliza nativamente el lenguaje procedural PL/PGSQL para implementar funciones, además de soportar otros, tales como: C, C++, Java PL, Java web y PL/PHP.
- Utiliza un método de almacenamiento llamado Control de Concurrencia para Múltiples Versiones (MVCC) que permite la eficiencia en ambientes donde se manejan grandes volúmenes de datos.

Para el desarrollo del módulo se utiliza la versión 9.5, esta se caracteriza por su estabilidad, potencia, robustez, facilidad de administración e implementación de estándares. Posee una gran

Capítulo 1: Fundamentación teórica

documentación en español en su sitio web oficial y foros de internet. Es un software con licencia libre, utilizado en el desarrollo del SIGIES.

1.4.5 Cliente de base de datos

PgAdmin III es una aplicación de diseño y manejo de bases de datos para su uso con PostgreSQL. La aplicación se puede utilizar para manejar PostgreSQL 7.3 o superiores y funciona sobre casi todas las plataformas. Este software fue diseñado para responder a las necesidades de todos los usuarios, desde la escritura de simples consultas SQL a la elaboración de bases de datos complejas (Date, 2001). La interfaz gráfica es compatible con todas las características de PostgreSQL y facilita la administración. La aplicación también incluye un editor de la sintaxis SQL, un editor de código del lado del servidor, un agente para la programación de tareas «SQL/batch/shell», soporte para el motor de replicación Slony-I. La conexión del servidor se puede realizar mediante TCP/IP o Unix Domain Sockets (en plataformas *nix), y puede ser cifrado mediante SSL por seguridad.

En pgAdmin III se puede ver y trabajar con casi todos los objetos de la base de datos, examinar sus propiedades y realizar tareas administrativas. Otra característica interesante es que, cada vez que se realiza alguna modificación en un objeto, escribe la(s) sentencia(s) SQL correspondiente(s), esto permite que además de una herramienta muy útil, sea a la vez didáctica. También incorpora funcionalidades para realizar consultas, examinar su ejecución (como el comando explain) y trabajar con los datos.

1.4.6 Servidor Web

Un servidor web es un programa que implementa el protocolo HTTP (*Hypertext Transfer Protocol*, en español Protocolo de Transferencia de Hipertexto), que está diseñado para transferir hipertextos, páginas web o páginas HTML. Este programa que se ejecuta continuamente en un ordenador, se mantiene a la espera de peticiones por parte de un cliente (navegador web). Al responder o ejecutar estas peticiones, se exhibe una página web en el navegador o se muestra el respectivo mensaje, si detectó algún error (Morera, 2000). A continuación, se describe el servidor web utilizado en el desarrollo de la propuesta solución, definido por el proyecto al cual se le integrará el módulo en cuestión.

NGINX: es un servidor web HTTP de código abierto que también incluye servicios de correo electrónico con acceso al Internet Message Protocol (IMAP) y al servidor Post Office Protocol (POP). Además, NGINX está listo para ser utilizado como un proxy inverso. En este modo,

Capítulo 1: Fundamentación teórica

NGINX se utiliza para equilibrar la carga entre los servidores back-end, o para proporcionar almacenamiento en caché para un servidor back-end lento (Nedelcu, 2015).

La principal ventaja de Nginx como servidor web es que consume muchos menos recursos al servir contenido estático. Esto convierte a Nginx en una excelente opción para funcionar como proxy inverso o como balanceador de carga para otros servidores como Apache, optimizando la entrega de contenidos.

Nginx permite responder a millones de peticiones por segundo aprovechando al máximo los núcleos o hilos de ejecución del servidor con una configuración muy simple. A continuación, se ofrecen otras características a tener muy en cuenta.

Características:

- Es ligero: reduce el consumo de RAM.
- Es multiplataforma y fácil de instalar.
- Caché: puedes usar Nginx como caché, con algo de configuración, permite mejorar la eficiencia de la aplicación sin tocar la programación de esta.
- Balanceador de carga: este servidor web puede funcionar como balanceador de carga, distribuyendo el tráfico entre varios servidores, permitiendo mayor escalabilidad.
- Compatibilidad con las aplicaciones web más populares: es compatible con una gran cantidad de CMS existentes en el mercado. Existe una amplia documentación para instalar estos bajo Nginx, como por ejemplo: Wordpress, Joomla, Drupal, phpBB y otros.

En el desarrollo de SIGIES se emplea como servidor Nginx combinado con **PHP-FPM**. Esta es una alternativa a FastCGI como intérprete de PHP que consume mucha menos memoria al procesar código PHP. Además, funciona de forma independiente al servidor web, mediante el uso de procesos externos y sockets para funcionar (PHP-FPM, 2015).

Nginx y PHP-FPM proporcionan un rendimiento impresionante, incluso al procesar código complejo. Los tiempos de respuesta son imposibles de conseguir con cualquier otra configuración de servidor web e intérprete. PHP-FPM ofrece muchas más opciones de configuración que cualquier otro intérprete.

1.4.7 Entorno integrado de desarrollo

Un entorno de desarrollo integrado (IDE por sus siglas en inglés) es una herramienta informática que aporta funcionalidades al desarrollador durante todas las etapas del ciclo de vida del desarrollo de software. Este ciclo comienza con el análisis y diseño y continúa hasta la distribución y su mantenimiento del producto, de ahí el término "integrado". En resumen, un IDE

Capítulo 1: Fundamentación teórica

es una interfaz visual que permite al desarrollador trabajar con comodidad, se encarga de dialogar internamente con el compilador, el enlazador y demás herramientas (Kyrnin, 2014). A continuación, se describe el IDE empleado en el desarrollo de la propuesta solución.

PhpStorm: es un IDE de programación desarrollado por JetBrains, empresa de desarrollo de software cuyas herramientas están dirigidas hacia los desarrolladores y administradores de proyectos. Este IDE tiene como propósito ayudar a mejorar la calidad del código. Es una herramienta rápida teniendo en cuenta su tamaño, soporta muchos idiomas y marcos a través de plugins, además, es multiplataforma. PhpStorm se destaca por la ejecución del código en la misma interfaz del IDE. Este editor permite además la gestión de proyectos fácilmente, proporciona un fácil autocompletado de código, soporta el trabajo con PHP 5.5 y posee una sintaxis abreviada.

Para el desarrollo de la solución se hará uso de este IDE en su versión 8.0, ya que este es el utilizado en el proyecto al cual va dirigido la presente investigación. Además, es compatible con los principales marcos de trabajo como Symfony, Drupal, WordPress, Zend Framework, Laravel, Magento, CakePHP y Yii. Soporta diferentes sintaxis del lenguaje PHP, pero al mismo tiempo también es compatible con HTML, CSS y JavaScript. Por tanto, proporciona un entorno impresionante para una amplia gama de lenguajes de scripting que un desarrollador PHP puede necesitar durante el desarrollo de una aplicación web estándar. A su vez, proporciona un soporte completo para todas estas secuencias de comandos con la terminación inteligente de código, la inspección de código, la detección de errores y depuración inteligente para su aplicación web.

1.4.8 Herramientas CASE

Las herramientas de Ingeniería de Software Asistida por Computadora (en inglés, *Computer Aided Software Engineering*) comprende un amplio abanico de diferentes tipos de programas que se utilizan para ayudar a las actividades del proceso del software (Sommerville, 2005). Estas pueden ser el análisis de requerimientos, el modelado del sistema, la depuración y las pruebas.

Estas herramientas están destinadas a aumentar la productividad en el desarrollo de software y reducir el costo del mismo en términos de tiempo y dinero. Permiten representar un producto de software de forma completa, a través de diagramas que se desarrollan durante los ciclos de vida del proyecto. Una de las más usadas actualmente por las facilidades que ofrece es el Visual Paradigm, a continuación, se describe la herramienta antes mencionada.

Visual Paradigm es un software de modelado UML ideal para los interesados en la construcción de sistemas a gran escala, que necesiten confiabilidad y estabilidad en el desarrollo orientado a

Capítulo 1: Fundamentación teórica

objetos. Este permite analizar, diseñar, codificar, probar, desplegar, dibujar todo tipo de diagramas UML, generar código fuente a partir de dichos diagramas y también posibilita la elaboración de documentos (Oscar, 2013).

La herramienta permite aumentar la calidad del software y mejora la productividad en el desarrollo y mantenimiento del software. Aumenta el conocimiento informático de una empresa ayudando así a la búsqueda de soluciones para los requisitos. También permite la reutilización del software, portabilidad y estandarización de la documentación, además del uso de las distintas metodologías propias de la Ingeniería de Software.

Para el desarrollo de la solución se debe emplear Visual Paradigm en su versión 8.0 debido a que es el empleado en el proyecto al cual va dirigido la presente investigación. Además, es multiplataforma y cumple con las políticas de migración a software libre en Cuba. Proporciona excelentes facilidades de interoperabilidad con otras aplicaciones, soporta el ciclo de vida completo del desarrollo de software, lo que facilita la captura de requisitos, análisis, diseño e implementación.

Conclusiones parciales

Una vez elaborado el marco teórico de la investigación se obtuvo una mayor comprensión de la propuesta solución. A partir del estudio de soluciones similares se pudo determinar las características genéricas de un cliente FTP, que serán utilizadas como base para el desarrollo de la aplicación. Para desarrollar dicha propuesta se definió el uso de la metodología AUP-UCI la cual guiará el proceso de desarrollo de software. Para su construcción se utilizará como framework PHP Symfony v2.7 sobre el entorno de desarrollo PhpStorm v8.0, el cual trabajará en paralelo con el servidor Nginx v1.10.3. Como gestor de base de datos se empleará PostgreSQL en su versión 9.5 y como herramienta de modelado Visual Paradigm 8.0 utilizando el lenguaje de modelado UML en su versión 2.0.

Capítulo 2: Análisis y diseño

Capítulo 2: Análisis y diseño

Introducción

En el desarrollo de un software, es de vital importancia definir los procesos que intervienen en este para lograr así una mejor comprensión del sistema a desarrollar entre clientes y desarrolladores. En este capítulo se exponen artefactos que permiten describir la propuesta de solución: modelo de dominio, la especificación de requisitos funcionales y no funcionales para el desarrollo de los componentes personalizables. Se realiza la descripción de requisitos por procesos, que incluyen a su vez los prototipos de interfaz de usuario, el modelo de análisis (donde se identifican las clases del análisis y se realizan los diagramas de clases y de colaboración) y el modelo de diseño (encargado de realizar los diagramas de clases y de secuencia con estereotipos web). Otro aspecto que se define es la arquitectura y los patrones de diseño a utilizar, así como el modelo de entidad relación de la base de datos y el diagrama de despliegue.

2.1 Modelo de negocio

Según (Rosas, 2009), el modelado del negocio persigue como objetivos:

- Entender los problemas que la organización desea solucionar e identificar mejoras potenciales.
- Medir el impacto del cambio organizacional.
- Asegurar que clientes, usuarios finales, desarrolladores y los otros participantes tengan un entendimiento compartido del problema.
- Derivar los requerimientos del sistema de software, necesarios para dar soporte a los objetivos de la organización.
- Entender como el sistema a ser desarrollado entra dentro de la organización.

Teniendo en cuenta lo anterior, el SIGIES gestiona los subprocesos: Organización, Exámenes, Asignación y Otorgamiento, comprendidos en el proceso de Ingreso a la Educación Superior. En cada etapa de los subprocesos se generan un conjunto de datos resultante del flujo realizado. La propuesta de solución de la presente investigación no responde a los procesos identificados, partiendo de que está enfocada a permitir el almacenamiento y gestión de los datos resultantes

Capítulo 2: Análisis y diseño

del proceso en general. Teniendo en cuenta que no se puede modelar un proceso de negocio, se hace necesario identificar conceptos asociados al dominio de la investigación y la relación de ellos.

2.2 Modelo de dominio

El modelo de dominio es utilizado como base para analizar los requisitos del usuario en el flujo de la ingeniería del software (Pressman, 2006). Este permite observar los distintos conceptos que forman parte del dominio de interés, y modela la interacción de estos conceptos con la finalidad de obtener un entendimiento del negocio.

2.2.1 Conceptos del dominio

Para lograr una mejor comunicación entre usuarios, desarrolladores y clientes, se establece un vocabulario común que permita entender el funcionamiento del sistema.

SIGIES: Sistema de Gestión para el Ingreso a la Educación Superior.

Reportes: reportes personalizados generados por cada uno de los procesos.

Usuario: persona que interactúa con el sistema.

Procesos: procesos para el Ingreso a la Educación Superior.

Organización: subprocesos para el Ingreso a la Educación Superior.

Examen: subprocesos para el Ingreso a la Educación Superior.

Asignación: subprocesos para el Ingreso a la Educación Superior.

Otorgamiento: subprocesos para el Ingreso a la Educación Superior.

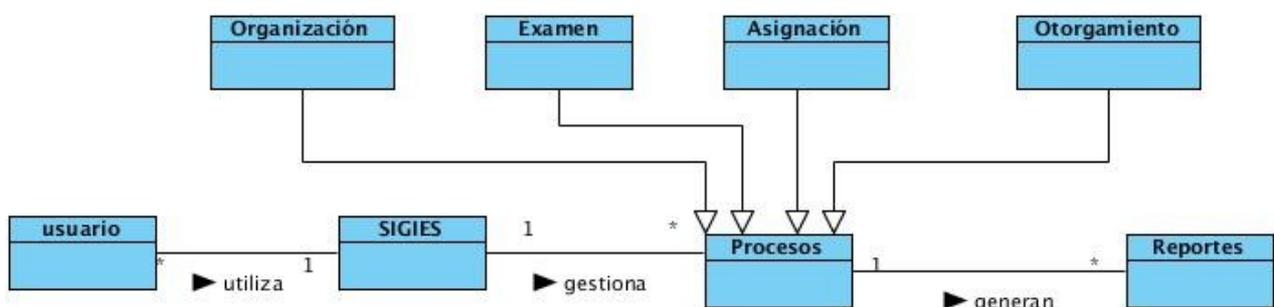


Fig 4: Diagrama de dominio

Capítulo 2: Análisis y diseño

2.3 Descripción de la propuesta solución

Con el objetivo de brindar un espacio dentro del sistema en el cual se pueda tener acceso de manera ágil y segura se propone realizar un cliente-ftp basado en la web. Este permitirá establecer la conexión a través del modo pasivo a un servidor FTP, utilizando el tipo de acceso de usuario. Este usuario debe tener privilegios de acceso a cualquier parte del sistema de archivos del servidor FTP, podrá crear, renombrar y eliminar directorios, editar, ver, descargar y subir archivos propios.

Para proporcionar la seguridad y el acceso a la información por parte del personal autorizado, se establecen varios niveles o roles que limitan las acciones a realizar sobre el contenido existente en el servidor FTP. La siguiente tabla describe los roles establecidos.

Tabla 1: Roles definidos

Roles	Descripción
Básico	Permite navegar por los directorios existentes en el servidor FTP y descargar el archivo seleccionado
Medio	Permite navegar por los directorios existentes en el servidor FTP y realizar las siguientes acciones sobre el contenido: <ul style="list-style-type: none">• Descargar archivos.• Subir archivos propios al servidor FTP.• Crear o modificar directorios en el servidor.
Avanzado	Permite navegar por los directorios existentes en el servidor FTP y realizar las siguientes acciones sobre el contenido: <ul style="list-style-type: none">• Descargar archivos.• Subir archivos propios al servidor FTP.• Crear y modificar directorios en el servidor.• Eliminar archivos y directorios.
Administrativo	Permite navegar por los directorios existentes en el servidor FTP y realizar las siguientes acciones sobre el contenido: <ul style="list-style-type: none">• Descargar archivos.• Subir archivos propios al servidor FTP.• Crear y modificar directorios en el servidor.• Eliminar archivos y directorios.

Capítulo 2: Análisis y diseño

Roles	Descripción
	<ul style="list-style-type: none">• Modificar el rol asignado a un usuario.• Agregar una conexión a un servidor FTP.• Eliminar una conexión con un servidor FTP.

La siguiente imagen ilustra el flujo de la información entre las comisiones descritas en la situación problemática una vez implementado el módulo en cuestión.

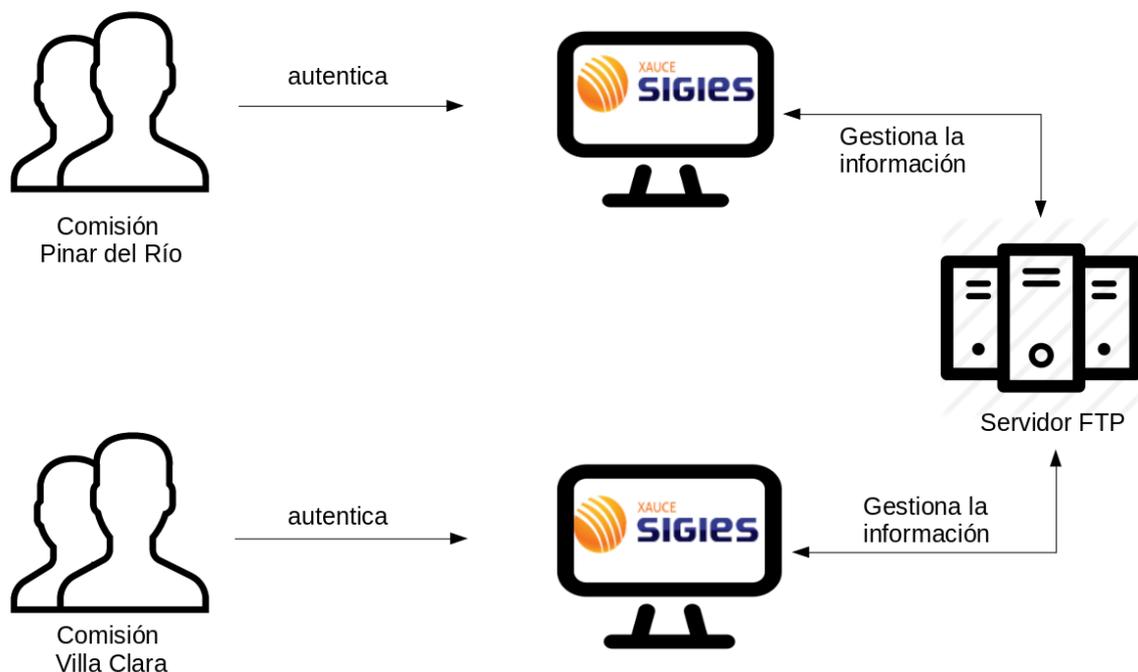


Fig 5: Flujo de información una vez integrado el módulo al SIGIES

El usuario con rol Funcionario Diul de la comisión de Pinar del Río accede al Sistema de Gestión para el Ingreso a la Educación Superior, genera los reportes pertinentes y los almacena en el servidor FTP. El usuario con rol Funcionario Diul de la comisión de Villa Clara al acceder al sistema puede establecer la conexión con el servidor FTP, donde encontrará los reportes generados por la comisión de Pinar del Río. Además, almacenará los propios para que todas las comisiones del resto del país puedan acceder a la información generada durante el proceso en esta comisión.

Capítulo 2: Análisis y diseño

2.4 Requerimientos del sistema

Los requerimientos del sistema establecen con detalles las funciones, servicios y restricciones operativas del sistema. Ellos deben ser claros y precisos porque definen exactamente lo que se va a implementar (Sommerville, 2005). Estos se dividen en dos grupos: requerimientos funcionales y requerimientos no funcionales. A continuación, se exponen cada uno de estos requisitos.

2.4.1 Requerimientos funcionales

Los requerimientos funcionales son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que este debe reaccionar a entradas particulares y como el sistema debe comportarse en distintas situaciones. En algunos casos estos requerimientos también pueden declarar explícitamente lo que el sistema no debe hacer (Sommerville, 2005). La propuesta de solución de la presente investigación debe satisfacer los requisitos funcionales siguientes:

1. Gestionar la conexión con el servidor FTP.
 - I. Crear la conexión con el servidor FTP:
Establece una conexión desde el sistema con el servidor FTP.
 - II. Listar las conexiones abiertas con el servidor FTP:
Muestra un listado con todas las conexiones establecidas.
 - III. Eliminar la conexión al servidor FTP:
Elimina la conexión establecida con el servidor FTP.
2. Listar el contenido que se encuentran en el servidor FTP:
Muestra todo el contenido alojado en un directorio situado en el servidor FTP.
3. Subir archivo al servidor FTP:
Permite al usuario transferir un archivo desde su pc hacia el servidor FTP.
4. Eliminar archivos del servidor FTP:
Permite al usuario eliminar el archivo deseado del servidor FTP.
5. Modificar archivos en el servidor FTP:
Permite al usuario modificar el nombre de un archivo deseado ubicado en el servidor FTP.
6. Descargar archivos del servidor FTP:
Permite al usuario transferir un archivo alojado en el servidor ftp hacia su pc.
7. Gestionar directorios en el servidor FTP.

Capítulo 2: Análisis y diseño

- I. Crear directorios en el servidor FTP:
Permite al usuario crear un directorio nuevo con el nombre deseado en el servidor FTP.
 - II. Eliminar directorios en el servidor FTP:
Permite al usuario eliminar un directorio alojado en el servidor FTP.
 - III. Modificar directorios en el servidor FTP:
Permite al usuario asignar un nuevo nombre a un directorio existente en el servidor FTP.
8. Asignar roles FTP:
Permite asignar a los usuarios las posibles acciones a realizar sobre los archivos mostrados.
 9. Mostrar permisos para la conexión.
Muestra el conjunto de acciones que puede realizar ese usuario sobre los recursos almacenados en la conexión actual.
 10. Mostrar información de los recursos en el servidor FTP:
Muestra la información referente al tamaño, la última fecha de modificación de un directorio y el tipo de recurso.

2.4.2 Requerimientos no funcionales

Los requerimientos no funcionales son restricciones de los servicios o funciones ofrecidas por el sistema. Estos se refieren a la fiabilidad, al tiempo de respuesta y la capacidad de almacenamiento que proporciona el sistema. A menudo son aplicados al sistema en su totalidad, aunque normalmente se aplican a características o servicios individuales del sistema (Sommerville, 2005).

La propuesta a desarrollar debe cumplir con los siguientes requisitos:

Accesibilidad:

- El sistema podrá ser usado en una red WAN (Wide Area Network) corporativa o una Red Privada Virtual(VPN).
- El sistema debe visualizarse y ejecutarse correctamente en un navegador web moderno, especialmente en Firefox (v20.x en adelante) y Google Chrome (v26.x en adelante), que son dos de los navegadores que mejor funcionan con HTML5 y CSS3.

Capítulo 2: Análisis y diseño

Usabilidad:

- El diseño de interfaz debe tener una iconografía que represente la acción a realizar y una navegación sugerente.
- El sistema mostrará sus mensajes y textos en el idioma español.

Seguridad:

- El sistema debe permitir la conexión vía SSL (del inglés Secure Sockets Layer) si el servidor lo requiere.
- El sistema debe contar con diferentes niveles de acceso a la información almacenada para garantizar la protección de información de accesos no autorizados.
- El sistema debe garantizar el acceso a las funcionalidades definidas para los usuarios de acuerdo a los roles que posean.
- Mantener el sistema disponible evitando que los mecanismos de seguridad impidan el acceso a la información requerida por los usuarios autorizados.

2.5 Descripción de requisitos por procesos

Con el objetivo de brindar una descripción detallada del comportamiento del sistema por cada requisito se realizarán 14 descripciones de requisitos por procesos. A continuación, se muestra el perteneciente al requisito Gestionar conexión sección Crear conexión con el servidor FTP. Los demás se pueden observar en el [Anexo 1](#).

Tabla 2: Crear conexión con el servidor FTP.

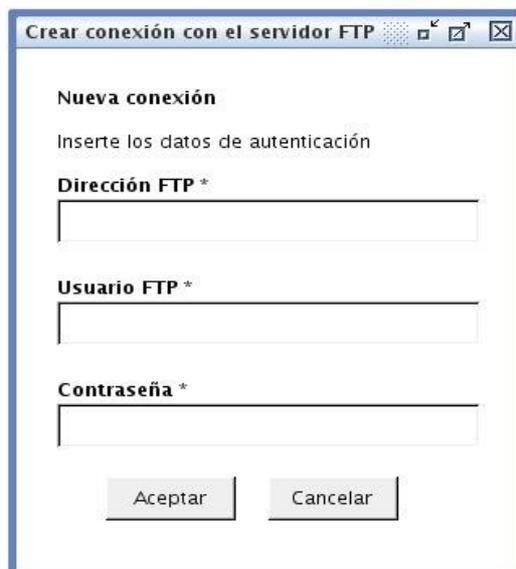
Precondiciones	El usuario debe estar autenticado en el sistema con el rol Administrador DIUL. El usuario debe poseer los permisos para crear conexión con el servidor FTP
Flujo de eventos	
Flujo básico crear conexión con el servidor FTP	
1.	El usuario selecciona la acción Nueva Conexión
2.	El sistema debe permitir introducir los siguientes datos para crear una conexión con el servidor FTP: <ul style="list-style-type: none">• (*) Dirección FTP• (*) Usuario FTP• (*) Contraseña Permite además, realizar las siguientes opciones: <ul style="list-style-type: none">• Aceptar

Capítulo 2: Análisis y diseño

	<ul style="list-style-type: none"> • Cancelar 		
3.	Se introducen los datos para crear una conexión con el servidor FTP.		
4.	El usuario selecciona la opción Aceptar.		
5.	El sistema valida los datos.		
6.	El sistema crea la nueva conexión con el servidor FTP y muestra un listado de los recursos alojados en el servidor FTP.		
7.	Concluye así el requisito.		
Pos-condiciones			
1.	Se creó la conexión con el servidor FTP satisfactoriamente.		
Flujo alternativo 5 a			
1.	Datos incorrectos		
2.	El sistema muestra un mensaje de información.		
Flujo alternativo 5 b			
1.	Campos vacíos		
2.	El sistema muestra un mensaje de información		
Flujo alternativo * Cancelar			
1.	El usuario selecciona la opción Cancelar.		
2.	El sistema regresa a la interfaz anterior. Ver documento DRP_Listar contenidos.odt		
3.	Concluye así el requisito.		
Pos-condiciones			
1.	No se crea la nueva conexión con el servidor FTP.		
Validaciones			
Modelo conceptual/Servidor FTP			
Conceptos	<table border="1" style="width: 100%;"> <tr> <td style="width: 30%;">Servidor FTP</td> <td> Visibles en la interfaz: <ul style="list-style-type: none"> • (*) Dirección FTP • (*) Usuario FTP • (*) Contraseña Uso interno: <ul style="list-style-type: none"> • id </td> </tr> </table>	Servidor FTP	Visibles en la interfaz: <ul style="list-style-type: none"> • (*) Dirección FTP • (*) Usuario FTP • (*) Contraseña Uso interno: <ul style="list-style-type: none"> • id
Servidor FTP	Visibles en la interfaz: <ul style="list-style-type: none"> • (*) Dirección FTP • (*) Usuario FTP • (*) Contraseña Uso interno: <ul style="list-style-type: none"> • id 		
Requisitos especiales	Seguridad: Acceso restringido		
Asuntos pendientes	NA		

Prototipo elemental de interfaz gráfica de usuario

Capítulo 2: Análisis y diseño



Crear conexión con el servidor FTP

Nueva conexión

Inserte los datos de autenticación

Dirección FTP *

Usuario FTP *

Contraseña *

Aceptar Cancelar

Fig 6: PIU_Crear conexión con el servidor FTP

2.6 Modelo de análisis

El modelo de análisis es la primera representación técnica de un sistema. Utiliza textos y diagramas para representar los requisitos del software, sus funciones y comportamiento. Además, permite examinar los requisitos desde diferentes puntos de vista aumentando la probabilidad de encontrar errores y de que se descubran nuevas debilidades a tiempo. De manera general, el objetivo de este flujo de trabajo es generar una arquitectura de objetos que sirva como base para el diseño posterior del sistema (Falgueras, 2002).

2.6.1 Diagrama de clases del análisis

Los diagramas de clases del análisis (DCA en lo adelante) representan la relación entre las clases que intervienen en los casos de uso. Son utilizados para comprender de forma general la estructura del sistema y sirve como entrada para la etapa de diseño (Salavert y Pérez, 2000). Se realizarón 10 diagramas, a continuación, se muestra el perteneciente al requisito Gestionar conexión con el servidor FTP. Los demás se pueden observar en el [Anexo 2](#).

Capítulo 2: Análisis y diseño

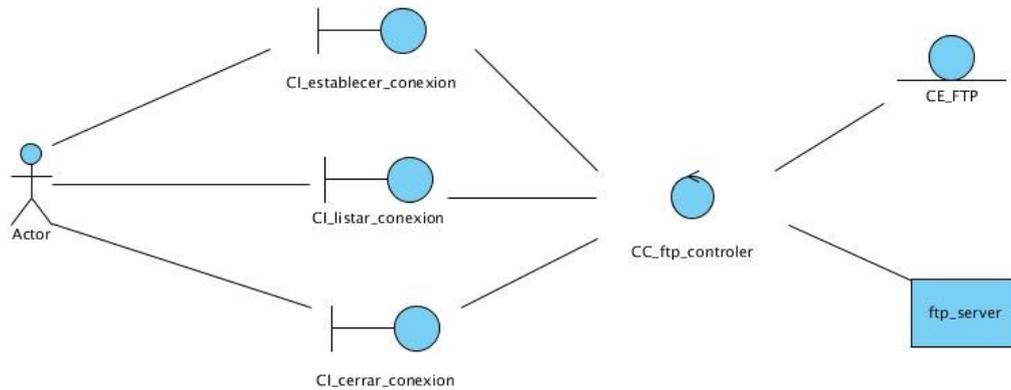


Fig 7: Diagrama de clases del análisis del requisito Gestionar conexión con el servidor FTP.

2.6.2 Diagrama de colaboración del análisis

Los diagramas de colaboración del análisis (DC en lo adelante) son una forma de representar la interacción entre objetos. Además, muestran como las instancias específicas de las clases trabajan juntas para conseguir un objetivo común (Salavert y Pérez, 2000). A continuación, se muestra el DC del requisito Gestionar conexión sección Crear conexión con el servidor FTP. El resto se detallan en los anexos. Se realizarón 14 diagramas, a continuación, se muestra el perteneciente al requisito Gestionar conexión sección Crear conexión con el servidor FTP. Los demás se pueden observar en el [Anexo 3](#).

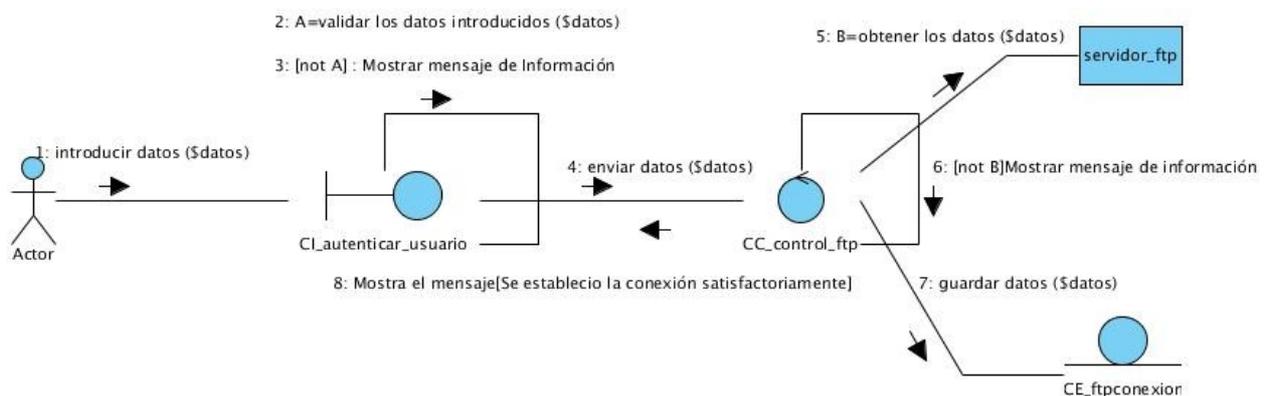


Fig 8: Diagrama de colaboración del análisis del requisito Gestionar conexión sección Crear conexión con el servidor FTP.

Capítulo 2: Análisis y diseño

2.7 Modelo de diseño

El modelo de diseño, es una abstracción de la implementación del sistema que describe la realización física de los casos de uso, centrado en cómo los requisitos funcionales y no funcionales impactan en el desarrollo de la aplicación. Representa todas las clases del diseño, subsistemas, paquetes, colaboraciones y las relaciones entre ellos, constituyendo la entrada principal a las actividades de la fase de implementación (Amo y Normand, 2005).

2.7.1 Patrón de arquitectura

Un patrón arquitectónico expresa un esquema de organización estructural esencial para un sistema de software, que consta de subsistemas que poseen responsabilidades e interrelaciones. Algunos patrones representan soluciones a problemas de rendimiento y otros pueden ser utilizados con éxito en sistemas de alta disponibilidad (Muñoz, 2010).

En el desarrollo de la propuesta solución se ha empleado el patrón Modelo Vista Controlador (MVC) que es el utilizado por el framework Symfony. Este framework toma lo mejor de la arquitectura MVC y la realiza de modo que el desarrollo de la aplicación sea de manera rápida y sencilla. Este patrón de arquitectura de software separa los datos de una aplicación, la interfaz de usuario y la lógica de negocio en tres componentes distintos:

El **modelo** contiene los datos y las funcionalidades esenciales. Es la representación específica de la información con que opera el sistema, es decir la representación de los datos y reglas del negocio. Es el encargado de manejar un registro de las vistas y de los controladores que existen en el sistema asegurando la integridad y disponibilidad de los datos. En Symfony el acceso y la modificación de los datos que se almacenan en la base de datos, se realiza mediante objetos. Debido a esto en el modelo se encuentran las clases, que son generadas de forma automática según la estructura de la BD.

La **vista** es la encargada de originar las páginas que son mostradas como resultado de las acciones, donde se encuentra el layout, que es común para todas las páginas de la aplicación. La vista en Symfony está conformada por varias partes preparadas para ser fácilmente transformable por la persona que trabaja con cada aspecto del diseño de las aplicaciones.

En el **controlador** se encuentran las acciones, las cuales son el núcleo de la aplicación, pues contienen toda la lógica. Estas acciones utilizan el modelo y precisan las variables para la vista.

Capítulo 2: Análisis y diseño

Al realizarse una petición web en una aplicación Symfony, la URL define una acción y los parámetros de la petición.

2.7.2 Patrones de diseños

Teniendo en cuenta las tecnologías y herramientas de desarrollo seleccionadas y el patrón de arquitectura antes elegido, se decide emplear para estructurar el diseño del sistema los siguientes patrones:

- Patrones GRASP (*General Responsibility Assignment Software Patterns*, por sus siglas en inglés).
- Patrones GOF (*Gang of four*, por sus siglas en inglés)

Los patrones **GRASP** describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones (Larman, 2003). Dentro de los patrones GRASP utilizados para el diseño del sistema y que incluye por defecto en su arquitectura el framework de desarrollo seleccionado, se destacan los siguientes:

Experto: consiste en asignar una responsabilidad al experto en información, en otras palabras, se asigna la responsabilidad a la clase que cuenta con la información necesaria para cumplirla. Se evidencia en las clases que extienden de la clase Entidad, las cuales son expertas en su propia información, tales como la clase Ftp.

Creador: asigna responsabilidades relacionadas con la creación de objetos o instanciación de nuevos objetos o clases. El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento. El empleo de este patrón será muy útil para aquellas clases encargadas de registrar datos de entidades que manejará el sistema, y este se evidencia en las clases: ftpConexionManager.

Controlador: se basa en asignar la responsabilidad de todos los eventos realizados a una clase específica que constituye el único punto de entrada para cada evento. En la solución propuesta este patrón está evidenciado en las clases controladoras encargadas de realizar las operaciones de gestión sobre las entidades y en la clase ftpController.

Los patrones **GOF** describen 23 patrones comúnmente utilizados y de gran aplicabilidad en problemas de diseño usando modelado UML. Se clasifican en tres categorías basadas en su propósito: creacionales, estructurales y de comportamiento. Para el diseño de la propuesta solución se han seleccionado los siguientes:

Capítulo 2: Análisis y diseño

Decorator: el objetivo de este patrón es permitir agregar dinámicamente funcionalidades suplementarias a un objeto (DEBRAUWER y EVAIN, 2015). Este se evidencia en la clase EstandarLayout.html.twig la cual contiene el código html que es usual en todas las plantillas del sistema para no tener que repetirlo en cada página. Desde otro punto de vista el layout decora la plantilla.

2.7.3 Diagrama de clases del diseño

Los diagramas de clases del diseño (DCD en lo adelante) son una representación más concreta y detallada que los diagramas de clases del análisis. También representan la parte estática del sistema conteniendo las clases y sus relaciones (Larman, 2003). Se realizaron 10 diagramas, a continuación, se muestra el perteneciente al requisito Gestionar conexión con el servidor FTP. Los demás se pueden observar en el [Anexo 4](#)

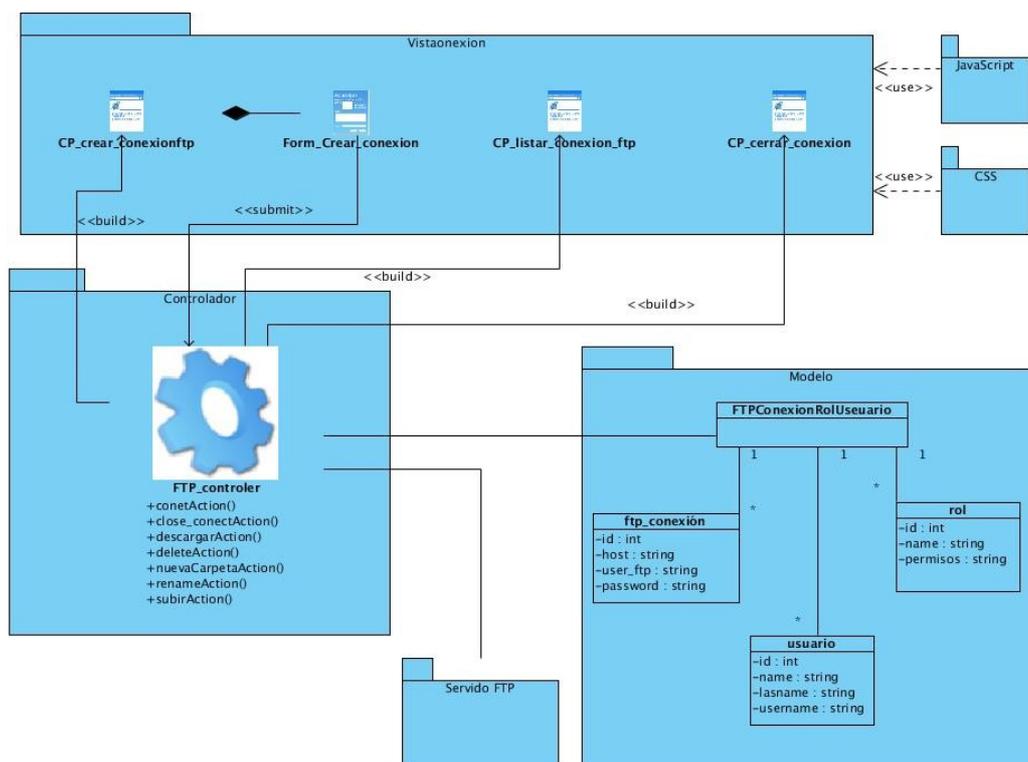


Fig 9: Diagrama de clases del diseño del requisito Gestionar conexión con el servidor ftp.

2.7.4 Diagrama de secuencia del diseño

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo. Este describe el comportamiento del sistema y las operaciones que se realizan

Capítulo 2: Análisis y diseño

representando los objetos y los mensajes que se intercambian. Tales interacciones suceden con el tiempo que se asigna, es decir, el diagrama de secuencias de UML es una mecánica de interacción en base a los tiempos (Larman, 2003). Se realizarón 14 diagramas, a continuación, se muestra el perteneciente al requisito Gestionar conexión sección Crear conexión con el servidor FTP. Los demás se pueden observar en el [Anexo 5](#).

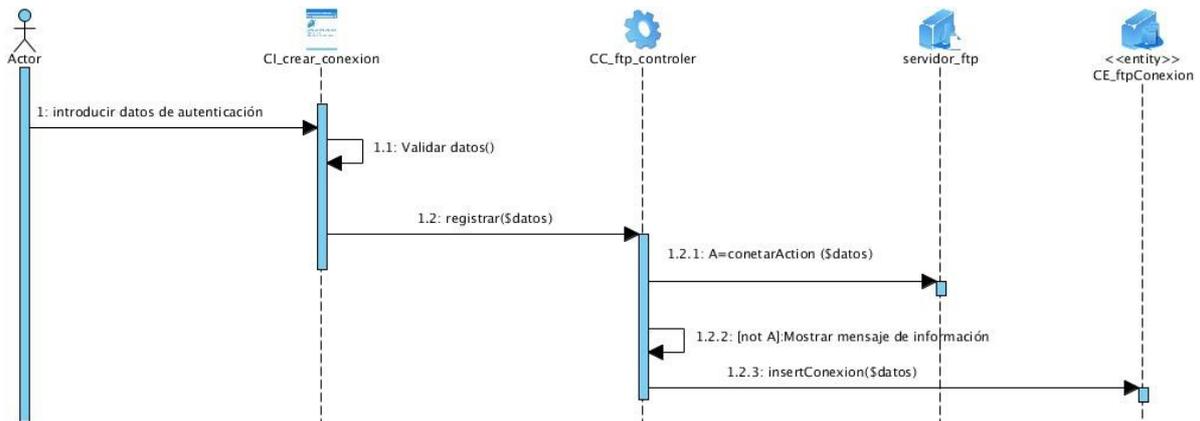


Fig 10: Diagrama de secuencia del requisito Gestionar conexión sección Crear conexión con el servidor ftp

2.7.5 Diagrama de despliegue

Un diagrama de despliegue muestra como el sistema se asentará físicamente en el entorno de hardware que lo acompaña. Su propósito es mostrar donde los componentes del sistema se ejecutarán y como se comunicarán entre ellos (Palomo y Gil, 2014).

Los elementos usados por este tipo de diagrama son nodos (representados como un prisma), componentes (representados como una caja rectangular con dos protuberancias del lado izquierdo) y asociaciones. El diagrama de despliegue que se muestra a continuación representa la distribución física del sistema en cuestión.

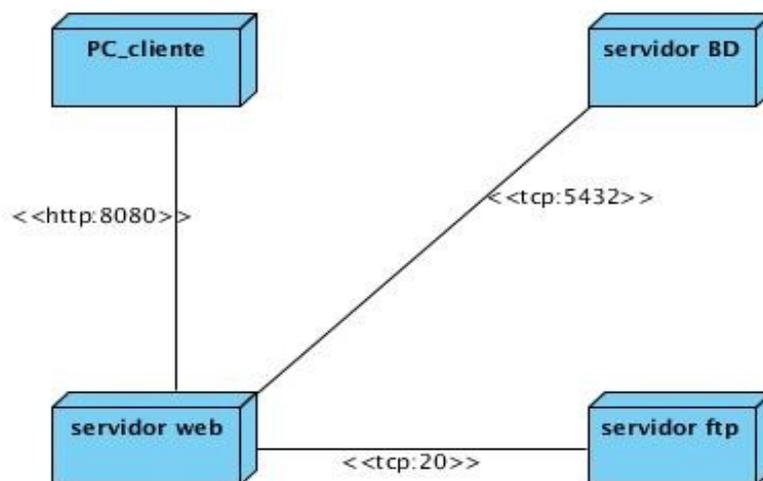


Fig 11: Diagrama de despliegue.

Capítulo 2: Análisis y diseño

2.8 Modelo de datos

Un modelo de datos es un lenguaje orientado a hablar de una Base de Datos. Típicamente un modelo de datos permite describir:

- Las estructuras de datos de la base: el tipo de los datos que hay en la base y la forma en que se relacionan.
- Las restricciones de integridad: un conjunto de condiciones que deben cumplir los datos para reflejar la realidad deseada.
- Operaciones de manipulación de los datos: típicamente, operaciones de agregado, borrado, modificación y recuperación de los datos de la base

2.8.1 Diseño de la base de datos

Un modelo de base de datos es un tipo de modelo de datos que determina la estructura lógica de una base de datos. También determina el modo de almacenar, organizar y manipular los datos (Revesz, 2010).

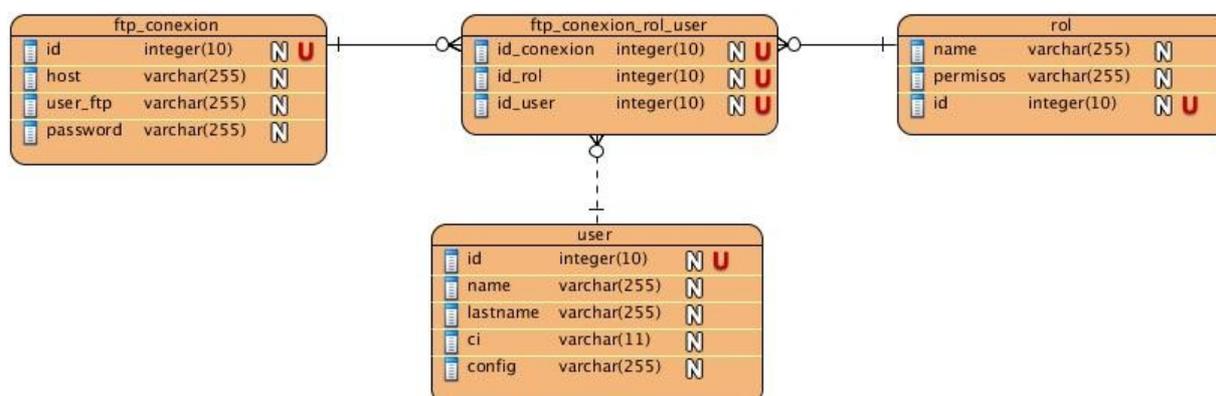


Fig 12: Diseño de la base de datos

2.8.2 Descripción de las tablas en la base de datos

Ftp_conexion: en esta tabla se almacena la información necesaria para establecer la conexión con el servidor ftp.

Rol: en esta tabla se almacena la información relacionada con las acciones que puede desarrollar un usuario en dependencia del tipo de rol que posee en el sistema.

User: en esta tabla se almacena la información relacionada con los distintos usuarios autenticados en el sistema

Capítulo 2: Análisis y diseño

Ftp_conexion_rol_user: esta tabla porta la relación que existe entre una conexión abierta y el rol desempeñado por el usuario autenticado.

Conclusiones parciales

La identificación de los principales conceptos asociados a la solución, reflejados en el modelo de dominio permitió definir los requisitos funcionales y no funcionales que sustentan la propuesta de solución representados a través de las descripciones de requisitos por proceso. El desarrollo del análisis y diseño del sistema, permitió definir las bases necesarias para la implementación y permitió que se obtuvieran de una forma más concreta y detallada las relaciones que se establecen entre clases. Además, los patrones de diseño que se emplean en el desarrollo de las funcionalidades quedaron bien definidos y argumentados. También se realizó el modelo de base de datos, donde quedaron plasmadas todas las relaciones de las tablas que serán objetos de persistencia en la BD para las entidades que se manejan en el sistema y el diagrama de despliegue. Así, se dejan creadas las bases para comenzar con la construcción de la propuesta, velando por el cumplimiento de los requerimientos identificados.

Capítulo 3: Implementación y prueba

Capítulo 3: Implementación y prueba

Introducción

En este capítulo se documenta el proceso de implementación de los elementos identificados durante la realización del diseño. Para ello se modela el diagrama de componentes. Además, se incluyen los resultados de las pruebas y las validaciones realizadas al módulo. De esta manera, se verificará que el módulo desarrollados cumplan con los requisitos establecidos y puedan ser integrados al Sistema de Gestión para el Ingreso a la Educación Superior.

3.1 Modelo de implementación

El modelo de implementación describe cómo los elementos del diseño se implementan en componentes. El propósito principal de este flujo de trabajo es desarrollar la arquitectura y el sistema como un todo. Describe también la organización de los componentes según los mecanismos de estructuración y modularización disponibles en el entorno de desarrollo, el lenguaje de programación utilizado, y la dependencia entre componentes. Como parte del modelo de implementación se obtienen los diagramas de componentes que a continuación se presentan (Burgués, 2016).

3.1.1 Diagrama de componente

El diagrama de componentes representa la estructura física de la implementación. Este diagrama permite visualizar la estructura de alto nivel del sistema y el comportamiento del servicio que estos proporcionan y usan a través de interfaces (Burgués, 2016). Además, organiza los subsistemas de implementación en capas, donde los componentes constituyen su elemento central. Los componentes pueden ser archivos, tablas, bibliotecas o documentos, estos son representados mediante un estereotipo. A continuación, se muestra el diagrama de componente de la propuesta de solución:

Capítulo 3: Implementación y prueba

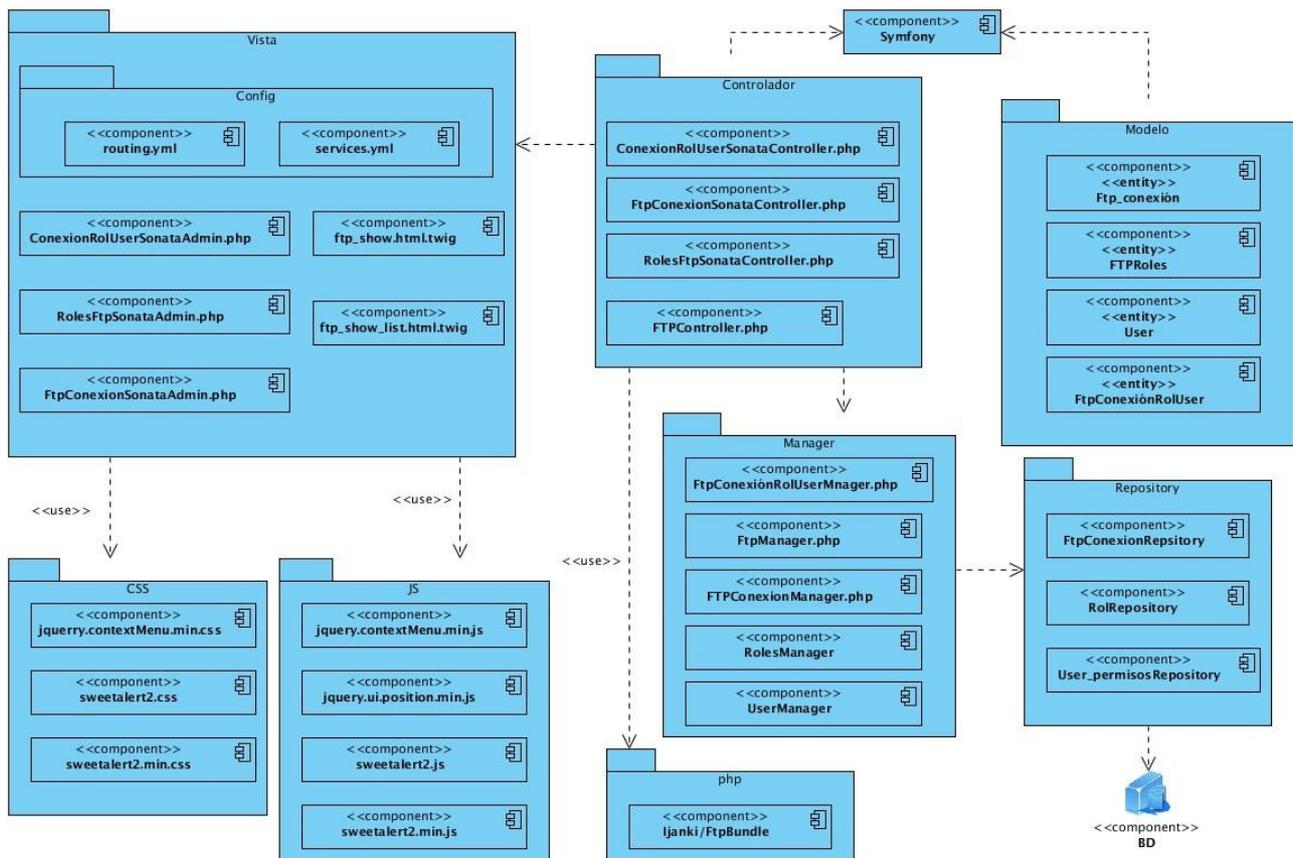


Fig 13: Diagrama de componentes

3.2 Pruebas de software

Las pruebas son un elemento importante dentro del ciclo de vida de un proyecto que proporciona una medida de la calidad del software. Estas son actividades en las cuales un sistema o componente es ejecutado bajo condiciones o requerimientos especificados, evaluando algún aspecto del sistema o componente. Los resultados son observados y registrados (Román y Cosín, 2007).

Dentro de esta fase pueden desarrollarse varios tipos de pruebas en función de los objetivos a lograr. Algunos tipos son pruebas funcionales, pruebas de usabilidad, pruebas de rendimiento, pruebas de seguridad, entre otras.

3.2.1 Niveles de Prueba

Los niveles de prueba son diferentes formas de verificar y validar un producto de software. A continuación se distinguen los siguientes niveles de prueba empleados para la identificación de los errores.

Capítulo 3: Implementación y prueba

Pruebas de integración del sistema: consiste en identificar grupos de componentes que proporcionan alguna funcionalidad del sistema e integrar estos añadiendo código para hacer que funcionen conjuntamente. Los componentes a integrar pueden ser: componentes comerciales, componentes reutilizables que han sido adaptados a un sistema particular o componentes nuevos desarrollados (Sommerville, 2005).

Pruebas de sistema: son las pruebas que se realizan cuando el software está funcionando como un todo. Es la actividad de prueba dirigida a verificar el programa final, después que todos los componentes de software y hardware han sido integrados (Román y Cosín, 2007). De manera general, este nivel de prueba es preparado y ejecutado por un grupo independiente al desarrollador. Estas consisten en validar que el software cumpla con los requerimientos especificados por el cliente. Para su ejecución se utiliza la técnica de caja negra, con el objetivo de detectar fallas en el cubrimiento de los requerimientos.

Pruebas de aceptación: en ocasiones son responsabilidad de los clientes o usuarios de un sistema, a pesar de que pueden participar otras partes interesadas. Estas pruebas consisten en evaluar la buena disposición de un sistema para su despliegue y uso (Román y Cosín, 2007).

Las pruebas de aceptación pueden darse en distintos momentos del ciclo de vida como:

- Un producto de software puede ser objeto de pruebas de aceptación una vez instalado o integrado.
- Las pruebas de aceptación de la usabilidad de un componente pueden realizarse durante las pruebas de componente.
- Las pruebas de aceptación de una nueva mejora funcional pueden realizarse antes de las pruebas de sistema.

3.2.2 Métodos de prueba

Un método de prueba es un enfoque sistemático, independiente del nivel en que se enmarque la prueba, que ayuda a encontrar buenos conjuntos de casos de prueba para detectar diferentes tipos de errores. Se puede probar cualquier producto de ingeniería de dos formas: conociendo la función específica para la cual fue diseñado el mismo y conociendo el funcionamiento del producto. Al primer enfoque se le denomina prueba de caja negra y al segundo, prueba de caja blanca (Pressman, 2006).

Capítulo 3: Implementación y prueba

- **Pruebas de caja blanca:** también conocidas como pruebas estructurales o pruebas de caja transparente se basan en un minucioso examen de los detalles procedimentales del código a evaluar. Estas pruebas aseguran que la operación interna se ajusta a las especificaciones y que todos los componentes internos se han comprobado de forma adecuada. Se rigen por un examen detallado de los procedimientos y caminos lógicos del sistema para determinar si el estado real coincide con el esperado.
- **Pruebas de caja negra:** también son denominadas pruebas funcionales o de comportamiento y se centran en los requisitos funcionales del software. Según lo definido por Pressman en el 2006, se llevan a cabo sobre la interfaz del software buscando errores en cada una de las funcionalidades. Con la aplicación de estas, se trata de demostrar que las funciones del software son completamente operativas, que las entradas se manejan de forma adecuada y que se produce el resultado esperado.

3.2.3 Diseño de casos de prueba

El diseño de casos de prueba consiste en probar el sistema, incluyendo los datos de entrada y los resultados esperados. Estos se derivan de las descripciones de requisitos por procesos y su objetivo fundamental es encontrar la mayor cantidad de defectos en las funcionalidades implementadas. Muestran si el sistema en realidad satisface las necesidades del cliente (Sommerville, 2005).

Para detectar la mayor cantidad de defectos en las funcionalidades del módulo se realizarón 14 diseño de casos de pruebas. A continuación, se presenta el diseño del caso de prueba perteneciente a la descripción de requisitos por procesos Crear la conexión con el servidor. Los demás se pueden consultar en el [Anexo 6](#).

Tabla 3: Crear la conexión con el servidor.

Descripción general: permitir establecer una conexión con el servidor FTP.						
Condiciones de ejecución: para crear una conexión hay que estar autenticado en el sistema.						
Nombre de la sección: SC1 Crear conexión con el servidor FTP						
Escenarios	Descripción	Dirección FTP	Usuario FTP	Contra seña	Respuesta del sistema	Flujo central
EC1.1 Crear la conexión con el	El usuario selecciona la opción nueva conexión.				El sistema debe permitir introducir los siguientes datos:	Principal/Conexiones/Nueva conexión

Capítulo 3: Implementación y prueba

servidor FTP.					dirección FTP usuario FTP contraseña. Y además, permite seleccionar las siguientes opciones: Conectar Cancelar	
EC1.2 Opción conectar	El usuario selecciona los datos de la conexión y ejecuta la opción conectar	V	V	V	El sistema valida los datos, crea una nueva conexión, la agrega al listado de conexiones abiertas y muestra el contenido de FTP.	Galería de Principal/Conexiones/Nueva conexión/Conectar
EC1.3 Opción cancelar	El usuario selecciona la opción de Cancelar.				El sistema regresa al listado de conexiones	Principal/Conexiones/Nueva conexión/Cancelar
EC1.4 Datos incompletos	Existen datos incompletos.	I V V	V I V	V V I	El sistema muestra un mensaje de información.	Principal/Conexiones/Nueva conexión/Conectar
EC1.5 Datos incorrectos	Existen datos incorrectos.	I V V	V I V	V V I	El sistema muestra un mensaje de información.	Principal/Conexiones/Nueva conexión/Conectarse

Capítulo 3: Implementación y prueba

Descripción de variables

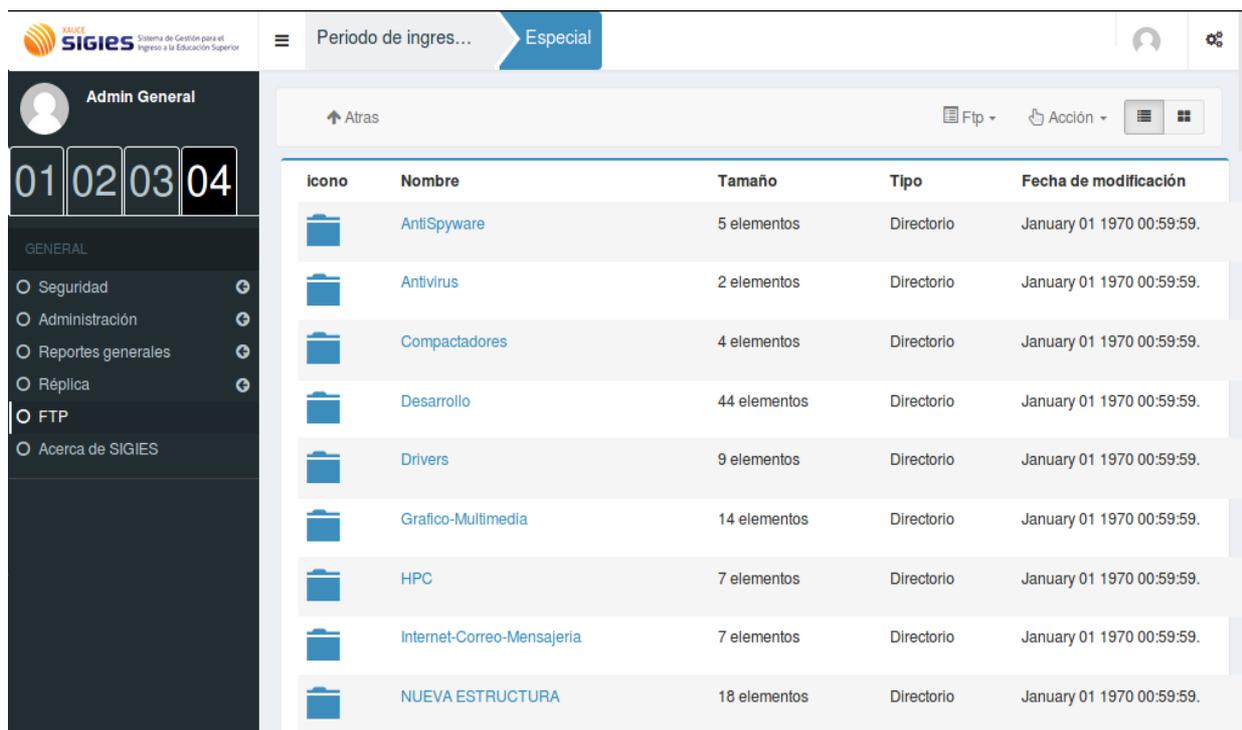
No	Nombre de campo	Clasificación	Valor nulo	Descripción
1	Dirección FTP	Campo de texto	No	Campo de texto que representa la ubicación del servidor FTP al cual desea conectarse. Ejemplo 1: misitioftp.cu Ejemplo 2: 10.0.0.1
2	Usuario FTP	Campo de texto	No	Campo de texto que representa el usuario a autenticar en el servidor FTP al cual desea conectarse. Ejemplo: miusuario
3	Contraseña	Campo de texto tipo password	No	Campo de texto que representa la clave de autenticación para el usuario que desea autenticarse.

3.2.4 Resultado de las pruebas

Resultados de las pruebas de integración:

Durante la implementación de las funcionalidades, se realizaron las pruebas de integración donde para evaluar la solución desarrollada se concibieron dos iteraciones. En las pruebas se obtuvo como resultado para una primera iteración una no conformidad, la cual fue resuelta. En una segunda iteración se obtuvieron resultados favorables, se logró que el módulo funcionara correctamente dentro del SIGIES. En resumen, las pruebas de integración validaron que las funcionalidades establecidas y los elementos del sistema operan satisfactoriamente, ajustándose a los requisitos especificados. A continuación, se muestra la imagen evidenciándose la integración.

Capítulo 3: Implementación y prueba



The screenshot displays the SIGIES system interface. At the top, there is a header with the SIGIES logo and the text 'Sistema de Gestión para el Ingreso a la Educación Superior'. Below the header, there is a navigation bar with 'Período de ingres...' and 'Especial'. On the left, there is a sidebar with a user profile 'Admin General' and a menu with options: 'GENERAL', 'Seguridad', 'Administración', 'Reportes generales', 'Réplica', 'FTP', and 'Acerca de SIGIES'. The main content area shows a table with the following data:

Icono	Nombre	Tamaño	Tipo	Fecha de modificación
	AntiSpyware	5 elementos	Directorio	January 01 1970 00:59:59.
	Antivirus	2 elementos	Directorio	January 01 1970 00:59:59.
	Compactadores	4 elementos	Directorio	January 01 1970 00:59:59.
	Desarrollo	44 elementos	Directorio	January 01 1970 00:59:59.
	Drivers	9 elementos	Directorio	January 01 1970 00:59:59.
	Gráfico-Multimedia	14 elementos	Directorio	January 01 1970 00:59:59.
	HPC	7 elementos	Directorio	January 01 1970 00:59:59.
	Internet-Correo-Mensajería	7 elementos	Directorio	January 01 1970 00:59:59.
	NUEVA ESTRUCTURA	18 elementos	Directorio	January 01 1970 00:59:59.

Fig 14: Módulo integrado al SIGIES

Resultados de las pruebas de sistema

En las principales pruebas pertenecientes al nivel de sistema que se realizaron se empleó el método de caja negra sobre las interfaces gráficas del sistema. La aplicación de este método tiene como objetivo verificar el cumplimiento de los requisitos funcionales establecidos para la presente investigación. Este método se utilizó teniendo en cuenta la técnica de partición por equivalencia, que permite examinar los valores válidos e inválidos de las entradas existentes en el software y descubrir de forma inmediata una clase de errores.

Se hace uso de los casos de prueba generados durante este flujo de trabajo con el fin de detectar la mayor cantidad de no conformidades posibles en las funcionalidades del módulo realizándose tres iteraciones de prueba. Las no conformidades fueron clasificadas según su complejidad por lo que se definen de la siguiente forma:

- Alta son los errores en el código o errores de funcionalidad.
- Media son los errores de ortografía o de validación.

Capítulo 3: Implementación y prueba

- Baja son los errores de interfaz.

Para el seguimiento de todo el proceso de corrección de no conformidades se realiza una gráfica.

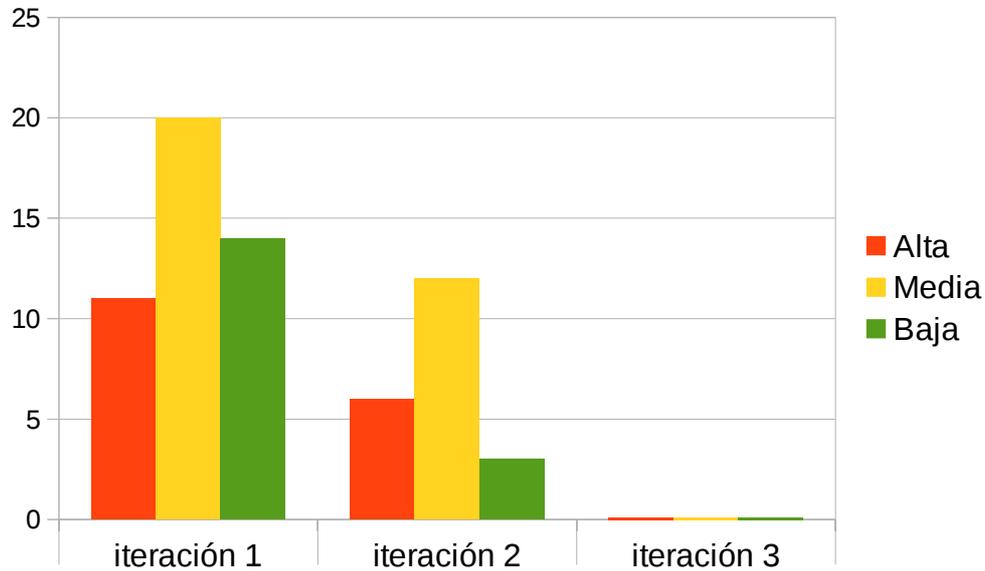


Fig 15: Gráfica de no conformidades

En la gráfica anterior se muestra la ejecución de tres iteraciones, donde en cada iteración fue resuelta cada una de las no conformidades identificadas, contribuyendo al mejoramiento de la propuesta para obtener un producto de calidad libre de errores y satisfacer las necesidades de los usuarios finales.

Resultados de las pruebas de aceptación:

En la realización de las pruebas de aceptación se efectuaron tres iteraciones, en las cuales se detectaron 19 no conformidades; de estas 5 se clasificaron de complejidad alta, 7 de complejidad media y 7 de complejidad baja.

En el siguiente gráfico se muestra la ejecución de tres iteraciones, donde en cada iteración fue resuelta cada una de las no conformidades identificadas.

Capítulo 3: Implementación y prueba

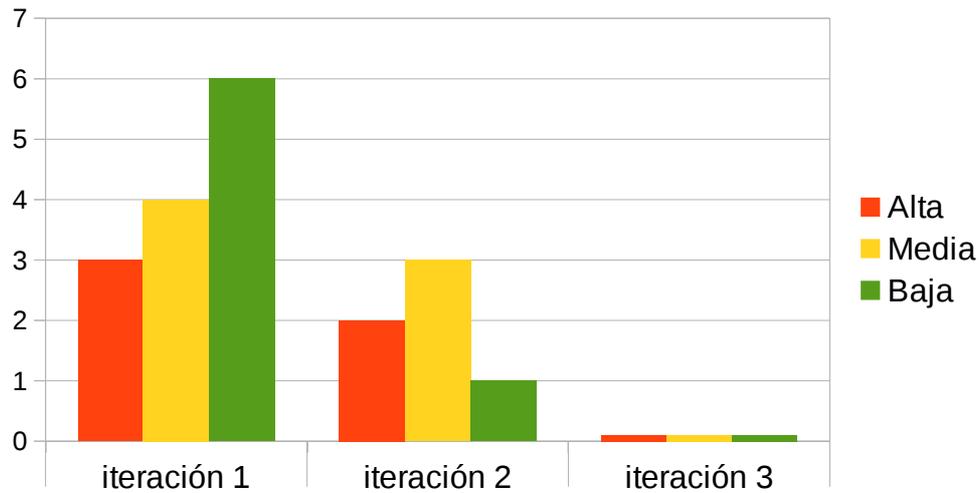


Fig 16: Gráfica de no conformidades

Conclusiones parciales

En esta etapa del desarrollo quedan implementadas en su totalidad las funcionalidades definidas al inicio de la investigación. El modelo de implementación, conformado por el diagrama de componentes obtenido, describió cómo se organizan y se implementan en términos de componentes los elementos del modelo del diseño. Además, las pruebas efectuadas al producto permitieron corregir errores para mejorar la calidad de la solución y garantizar la aceptación final por parte del cliente.

Conclusiones generales

Conclusiones generales

Al término de esta investigación se arriban a las siguientes conclusiones:

- La realización de un estudio del estado del arte posibilitó identificar los conceptos relacionados con el objeto de estudio.
- Mediante el análisis de las soluciones similares se determinaron las principales características a tener en cuenta para el desarrollo del módulo.
- Apoyado en la metodología de desarrollo seleccionada, se generan los artefactos ingenieriles que permiten describir la propuesta de solución.
- Con la aplicación de las pruebas de integración, de sistema y aceptación se detectaron varias no conformidades, las cuales fueron resueltas.
- Se logró implementar e integrar un módulo al SIGIES que permite, a través de un servidor FTP, brindar un espacio en el cual los usuarios tengan acceso a la información generada durante el proceso de Ingreso a la Educación Superior.

Recomendaciones

Recomendaciones

Con el objetivo que el usuario obtenga mayor facilidad y rapidez durante el proceso de búsqueda o transferencia de información entre los distintos servidores FTP accesibles desde el Sistema de Gestión para el Ingreso a la Educación Superior se recomienda:

- Implementar una funcionalidad que permita realizar la búsqueda en uno o varios servidores FTP a la vez.
- Implementar una funcionalidad que permita copiar archivos y directorios desde un servidor FTP hacia otro.

Referencias bibliográficas

Referencias bibliográficas

ALLMAN, M. y OSTERMANN, S., [sin fecha]. FTP Security Considerations. [en línea]. [Consulta: 21 mayo 2017]. Disponible en: <https://tools.ietf.org/html/rfc2577.html>.

AMO, F.A., NORMAND, L.M. y PÉREZ, F.J.S., 2005a. *Introducción a la ingeniería del software*. S.I.: Delta Publicaciones. ISBN 978-84-96477-00-1.

ARMAND, S., 2014. *Extending Symfony2 Web Application Framework*. S.I.: Packt Publishing Ltd. ISBN 978-1-78328-720-8.

BANSAL, A.K., 2013. *Introduction to Programming Languages*. S.I.: CRC Press. ISBN 978-1-4665-6514-2.

BURGUÉS, E.G., 2016. *Aprende a Modelar Aplicaciones con UML: 2ª Edición*. S.I.: IT Campus Academy. ISBN 978-1-5234-9853-6.

CENTER FOR HISTORY AND NEW MEDIA, [sin fecha]. Guía rápida. [en línea]. Disponible en: http://zotero.org/support/quick_start_guide.

COBB, C.G., 2011. *Making Sense of Agile Project Management: Balancing Control and Agility*. S.I.: John Wiley & Sons. ISBN 978-1-118-01570-4.

CORNELIO, E.R., RIVAS, C.G. y HERNÁNDEZ, J.C.R., 2004. *Bases de datos relacionales: diseño físico*. S.I.: Univ Pontifica Comillas. ISBN 978-84-8468-138-0.

CROCKFORD, D., 2008. *JavaScript: The Good Parts: The Good Parts*. S.I.: O'Reilly Media, Inc. ISBN 978-0-596-55487-3.

DATE, C.J., 2001. *Introducción a los sistemas de bases de datos*. S.I.: Pearson Educación. ISBN 978-968-444-419-5.

DEBRAUWER, L. y EVAÏN, Y., 2015. *Patrones de diseño en PHP: Los 23 modelos de diseño: descripciones y soluciones ilustradas en UML2 y PHP*. S.I.: Ediciones ENI. ISBN 978-2-7460-9837-4.

Referencias bibliográficas

- DURANGO, A., 2015. *Diseño de Software: 2ª Edición*. S.I.: IT Campus Academy. ISBN 978-1-5196-2073-6.
- FALGUERAS, B.C., 2002a. *Ingeniería del software*. S.I.: Universitat Oberta de Catalunya. ISBN 978-84-8429-793-2.
- FUENTES, J.R.L., 2015. *Desarrollo de Software ÁGIL: Extreme Programming y Scrum*. S.I.: IT Campus Academy. ISBN 978-1-5029-5222-6.
- GALINDO, M.J.M., 2010. *Escaneando la informática*. S.I.: Editorial UOC. ISBN 978-84-9788-110-4.
- GAMMA, E., 2002. *Patrones de diseño: elementos de software orientado a objetos reutilizable*. S.I.: Pearson Educación. ISBN 978-84-7829-059-8.
- GARCÍA, M.Á.A., 2012. *CSS3*. S.I.: Anaya Multimedia. ISBN 978-84-415-3114-7.
- GAUCHAT, J.D., 2012. *El gran libro de HTML5, CSS3 y Javascript*. S.I.: Marcombo. ISBN 978-84-267-1782-5.
- HERNÁNDEZ, M.T., 2016. *Symfony Framework: Desarrollo Rápido de Aplicaciones Web. 2ª Edición*. S.I.: IT Campus Academy. ISBN 978-1-5404-2002-2.
- HOYOS, L.G., 2001. *Instalación de un servidor Web. Ventajas e inconvenientes de las aplicaciones open source*. S.I.: s.n.
- KYRNIN, J., [sin fecha]. Here's What an «IDE» Is (Building Web Apps). *ThoughtCo* [en línea]. [Consulta: 25 mayo 2017]. Disponible en: <https://www.thoughtco.com/what-is-an-ide-3471199>.
- MCFARLAND, D.S., 2008. *JavaScript: The Missing Manual: The Missing Manual*. S.I.: O'Reilly Media, Inc. ISBN 978-0-596-55530-6.
- MCFARLAND, D.S., 2012. *CSS3: The Missing Manual*. S.I.: O'Reilly Media, Inc. ISBN 978-1-4493-3948-7.
- MITCHELL, J.C., 2003. *Concepts in Programming Languages*. S.I.: Cambridge University Press. ISBN 978-0-521-78098-8.

Referencias bibliográficas

- MORERA, J. de D.M., 2000. *Implementación de un servidor ftp utilizando el modelo cliente/servidor mediante el uso de sockets en lenguaje C UNIX con el fin de mejorar los tiempos de respuesta en la red. Uniciencia. 24(1), 2010.* S.I.: Red Universidad Nacional de Costa Rica.
- MUÑOZ, C.C., VELTHUIS, M.G.P. y RUBIA, M.Á.M. de la, 2010. *Calidad del producto y proceso software.* S.I.: Editorial Ra-Ma.
- NEDELCO, C., 2015. *Nginx HTTP Server.* S.I.: Packt Publishing. ISBN 978-1-78528-591-2.
- NOITE.PL, [sin fecha]. *PostgreSQL database administration tools: Linux Services. AL3-056.* S.I.: NOITE S.C.
- ORTEGA-ARJONA, J.L., [sin fecha]. Wiley: *Patterns for Parallel Software Design - Jorge Luis Ortega-Arjona.* [en línea]. [Consulta: 11 mayo 2017]. Disponible en: <http://www.wiley.com/WileyCDA/WileyTitle/productCd-0470697342.html>.
- OSCAR, S., 2013. *Visual Paradigm for Uml.* S.I.: International Book Market Service Limited. ISBN 978-613-9-16653-4.
- PALOMO, S.R.G. y GIL, E.M., 2014. *Aproximación a la ingeniería del software.* S.I.: Editorial Universitaria Ramon Areces. ISBN 978-84-9961-093-1.
- PIERCE, B.C., 2002. *Types and Programming Languages.* S.I.: MIT Press. ISBN 978-0-262-16209-8.
- POTENCIER, F., [sin fecha]. *Practical symfony - Doctrine edition.* S.I.: Lulu.com. ISBN 978-2-918390-01-5.
- PRESSMAN, R.S., 2006. *Ingeniería del software: un enfoque práctico.* S.I.: McGraw-Hill. ISBN 978-970-10-5473-4.
- SALAVERT, I.R. y PÉREZ, M.D.L., 2000. *Ingeniería del software y bases de datos: tendencias actuales.* S.I.: Univ de Castilla La Mancha. ISBN 978-84-8427-077-5.
- SOMMERVILLE, I., 2005. *Ingeniería del software.* S.I.: Pearson Educación. ISBN 978-84-7829-074-1.

Referencias bibliográficas

- TECNOLÓGICA, C. de I., 1997. *Información Tecnológica*. S.I.: Centro de Información Tecnológica.
- TSANG, C.H.K., LAU, C.S.W. y LEUNG, Y.K., 2005. *Object-oriented Technology: From Diagram to Code with Visual Paradigm for UML*. S.I.: McGraw-Hill. ISBN 978-0-07-124046-8.
- TUYA, J., ROMÁN, I.R. y COSÍN, J.J.D., 2007. *Técnicas cuantitativas para la gestión en la ingeniería del software*. S.I.: Netbiblo. ISBN 978-84-9745-204-5.
- VÁZQUEZ, P., 2006. *Creación De Sitios Web/ Creation of Websites*. S.I.: USERSHOP. ISBN 978-987-22995-9-0.
- AMO, F.A., NORMAND, L.M. y PÉREZ, F.J.S., 2005. *Introducción a la ingeniería del software*. S.I.: Delta Publicaciones. ISBN 978-84-96477-00-1.
- ANDREU, J., 2011a. *Servicios de acceso remoto (Servicios en red)*. S.I.: Editex. ISBN 978-84-9003-085-1.
- ANDREU, J., 2011b. *Servicios FTP (Servicios en red)*. S.I.: Editex. ISBN 978-84-9003-086-8.
- BERTINO, E. y MARTINO, L., 1995. *Sistemas de bases de datos orientadas a objetos: conceptos y arquitecturas*. S.I.: Ediciones Díaz de Santos. ISBN 978-0-201-65356-4.
- CASTAÑO, J.J. y JURADO, S., [sin fecha]. *Publicación y mantenimiento de la web corporativa (Comercio electrónico)*. S.I.: Editex. ISBN 978-84-9078-966-7.
- DÁVILA, M.R., CALLEJO, M.L. y GONZÁLEZ, M.E., 2004. *Las TIC, un reto para nuevos aprendizajes: Usar información, comunicarse y utilizar recursos*. S.I.: Narcea Ediciones. ISBN 978-84-277-1460-1.
- ESTRADA, A.C., 2011. *Seguridad por niveles*. S.I.: Alejandro Corletti. ISBN 978-84-615-3814-0.
- FOWLER, M. y SCOTT, K., 1999. *UML gota a gota*. S.I.: Pearson Educación. ISBN 978-968-444-364-8.
- GAMMA, E., 2002. *Patrones de diseño: elementos de software orientado a objetos reutilizable*. S.I.: Pearson Educación. ISBN 978-84-7829-059-8.
- GÓMEZ, J.A., 2010. *Servicios en Red*. S.I.: Editex. ISBN 978-84-9771-760-1.

Referencias bibliográficas

- HARE, C., 1995. *Internet y seguridad en redes*. S.I.: Prentice-Hall Hispanoamericana. ISBN 978-968-880-605-0.
- HERNÁNDEZ, M.T., 2016. *Symfony Framework: Desarrollo Rápido de Aplicaciones Web*. 2ª Edición. S.I.: IT Campus Academy. ISBN 978-1-5404-2002-2.
- JACOBSON, I., BOOCH, G. y RUMBAUGH, J., 2000. *El proceso unificado de desarrollo de software*. S.I.: Pearson Educación. ISBN 978-84-7829-036-9.
- KATZ, R., 2009. *El Papel de las TIC en el Desarrollo*. S.I.: Raul Katz. ISBN 978-84-08-09116-5.
- KENDALL, K.E. y KENDALL, J.E., 2005. *Análisis y diseño de sistemas*. S.I.: Pearson Educación. ISBN 978-970-26-0577-5.
- LANCKER, L.V., 2012. *HTML5: Los fundamentos del lenguaje*. S.I.: Ediciones ENI. ISBN 978-2-7460-7516-0.
- LARMAN, C., 2003. *UML y patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado*. S.I.: Pearson Educación. ISBN 978-84-205-3438-1.
- LÓPEZ, E.T., RAMON, A.O., SARROCA, E.M. y SEONE, C.G., 2004. *Diseño de sistemas software en UML*. S.I.: Univ. Politèc. de Catalunya. ISBN 978-84-9880-075-3.
- LÓPEZ, P.A. y FERNÁNDEZ, M.M., 2008. *Informática 4º ESO*. S.I.: Editex. ISBN 978-84-9771-351-1.
- MONTERO, S., ZARRAONADÍA, T., DÍAZ, P., AEDO, I., (DIRS.), P.S., Antonio, AURELIO, L.P., GRAFS.), A.C., May Ann (Diss, MAQ.), C.G., Ana (Dis y AN, E.F., 2011. *Patrones de diseño aplicados al desarrollo de objetos digitales educativos (ODE)*. S.I.: Ministerio de Educación. ISBN 978-84-369-5083-0.
- MUÑOZ, V.J.E., 2013. *El nuevo PHP paso a paso*. S.I.: Vicente Javier Eslava Muñoz. ISBN 978-84-686-4110-2.
- NOITE.PL, 2010. *FTP Protocol: Network Basic*. AL0-033. S.I.: NOITE S.C.
- RAFAEL BARZANALLANA, 2012. ¿Qué son los lenguajes de marcado? Rafael Menéndez-Barzanallana Asensio. Universidad de Murcia. [en línea]. [Consulta: 23 enero 2017].

Referencias bibliográficas

Disponible en: <http://www.um.es/docencia/barzana/DIVULGACION/INFORMATICA/Que-son-lenguajes-marcado.html>.

REVESZ, P., 2010. *Introduction to Databases: From Biological to Spatio-Temporal*. S.l.: Springer Science & Business Media. ISBN 978-1-84996-094-6.

RODRÍGUEZ, J.G., [sin fecha]. *Guía completa de HTML5 y CSS3: Todo lo necesario para aprender a programar páginas web, con ejercicios resueltos y explicados*. S.l.: Javier García.

RUEDA, R.A.S., 2016. *DISEÑO Y ANÁLISIS DE UN SISTEMA WEB EDUCATIVO CONSIDERANDO LOS ESTILOS DE APRENDIZAJE*. S.l.: 3Ciencias. ISBN 978-84-945785-4-0.

RUMBAUGH, J., BOOCH, G. y JACOBSON, I., 2007. *El Lenguaje unificado de modelado: manual de referencia*. S.l.: Pearson Educación. ISBN 978-84-7829-087-1.

SERGIO AGUDO, 2016. ¿Qué es un servidor FTP y cómo se utiliza? [en línea]. [Consulta: 22 enero 2017]. Disponible en: <http://rootear.com/web/servidor-ftp-que-es-como-se-utiliza>.

ROSAS, U., 2009. Ingeniera de Software: Disciplinas del modelo RUP. Ingeniera de Software [en línea]. [Consulta: 23 junio 2017]. Disponible en: <http://isw-newbe.blogspot.com/2009/12/disciplinas-del-modelo-rup.html>. Ograficas

MIGUEL, S.E.Ó.Y.M.V., 2011. Aplicaciones informáticas de bases de datos. S.l.: Editorial Paraninfo. ISBN 978-84-283-3244-6. ón Del Lado Servidor. Google Docs [en línea], [sin fecha]. [Consulta: 21 mayo 2017]. Disponible en: https://docs.google.com/presentation/d/1ZQgIsW4KCylfH8NyiQH9ueanM83Ym7bjitaxyQKS0hs/edit?pli=1&usp=embed_facebook.

Anexo 1: Descripción de requisitos por procesos.

DRP_Listar conexiones abiertas con el servidor FTP.

Precondiciones	El usuario debe estar autenticado en el sistema con el rol Administrador DIUL. El usuario debe poseer los permisos para Listar las conexiones abiertas con el servidor FTP.
Flujo de eventos	
Flujo básico Listar conexiones abiertas con el servidor FTP.	
1.	El usuario selecciona la opción FTP
2.	El sistema muestra el listado de las conexiones creadas.
3.	Concluye así el requisito.
Pos-condiciones	
1.	Se mostró el listado de conexiones abiertas con el servidor FTP.
Validaciones	
	FORTES_SIGIES_Modelo_conceptual/Servidor FTP
Conceptos	Visibles en la interfaz: Uso interno: <ul style="list-style-type: none"> id
Requisitos especiales	Seguridad: Acceso restringido
Asuntos pendientes	NA

Prototipo elemental de interfaz gráfica de usuario

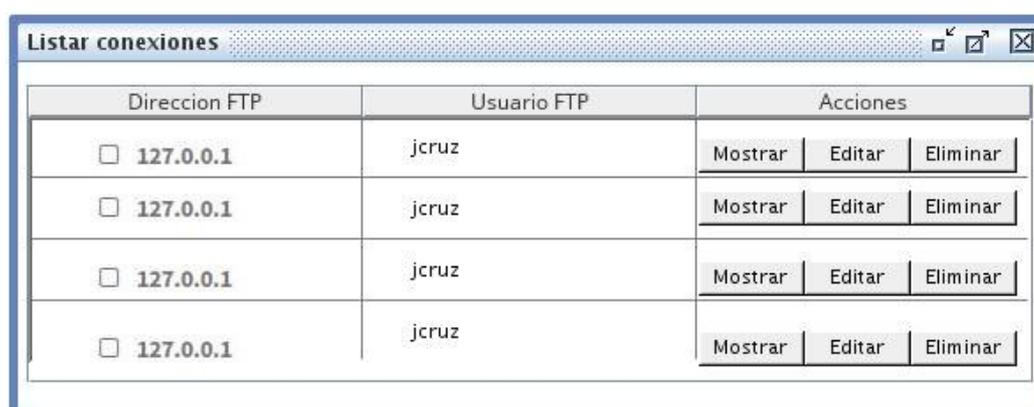


Fig. 1: PIU_Listar conexiones abiertas con el servidor FTP

DRP_Eliminar conexiones con el servidor FTP.

Precondiciones	El usuario debe estar autenticado en el sistema con el rol Administrador DIUL. El usuario debe poseer los permisos para Eliminar conexiones con el servidor FTP.
-----------------------	---

Flujo de eventos

Flujo básico Eliminar conexiones con el servidor FTP.

1.	El usuario selecciona la opción Eliminar conexión
2.	El sistema debe permitir seleccionar las conexiones con el servidor FTP a ser eliminadas. Además, permite realizar las siguientes opciones: <ul style="list-style-type: none">• Aceptar• Cancelar
3.	El usuario selecciona la opción Aceptar.
4.	El sistema elimina las conexiones del listado de conexiones abiertas.
5.	Concluye así el requisito.

Pos-condiciones

1. Se eliminaron las conexiones satisfactoriamente.

Flujo alternativo * Cancelar

1.	El usuario selecciona la opción Cancelar.
2.	El sistema regresa a la interfaz anterior. Ver documento DRP_Listar contenidos.odt
3.	Concluye así el requisito.

Pos-condiciones

1. No eliminan las conexiones con el servidor FTP

Validaciones

FORTES_SIGIES_Modelo_conceptual/Servidor FTP

Conceptos	Visibles en la interfaz: Uso interno: <ul style="list-style-type: none">• id
------------------	---

Requisitos especiales	Seguridad: Acceso restringido
Asuntos pendientes	NA

Prototipo elemental de interfaz gráfica de usuario

Direccion FTP	Usuario FTP	Acciones
<input type="checkbox"/> 127.0.0.1	jcruz	Mostrar Editar Eliminar
<input type="checkbox"/> 127.0.0.1	jcruz	Mostrar Editar Eliminar
<input type="checkbox"/> 127.0.0.1	jcruz	Mostrar Editar Eliminar
<input type="checkbox"/> 127.0.0.1	jcruz	Mostrar Editar Eliminar

Fig. 1: PIU_Eliminar conexiones con el servidor FTP

DRP_Listar contenido alojado en el servidor FTP

Precondiciones	El usuario debe estar autenticado en el sistema El usuario debe poseer los permisos para Listar el contenido alojado en el servidor FTP.
Flujo de eventos	
Flujo básico Listar el contenido alojado en el servidor FTP.	
1.	El usuario selecciona la opción la conexión con el servidor FTP
2.	El sistema debe mostrar los recursos alojados el servidor FTP.
3.	El sistema debe permitir seleccionar un directorio y mostrar el contenido de este.
4.	Concluye así el requisito.
Pos-condiciones	
1.	Se mostró el contenido satisfactoriamente.
Validaciones	
FORTES_SIGIES_Modelo_conceptual/Servidor FTP	
Conceptos	Visibles en la interfaz: Uso interno: <ul style="list-style-type: none">• id
Requisitos especiales	Seguridad: Acceso restringido
Asuntos pendientes	NA

Prototipo elemental de interfaz gráfica de usuario

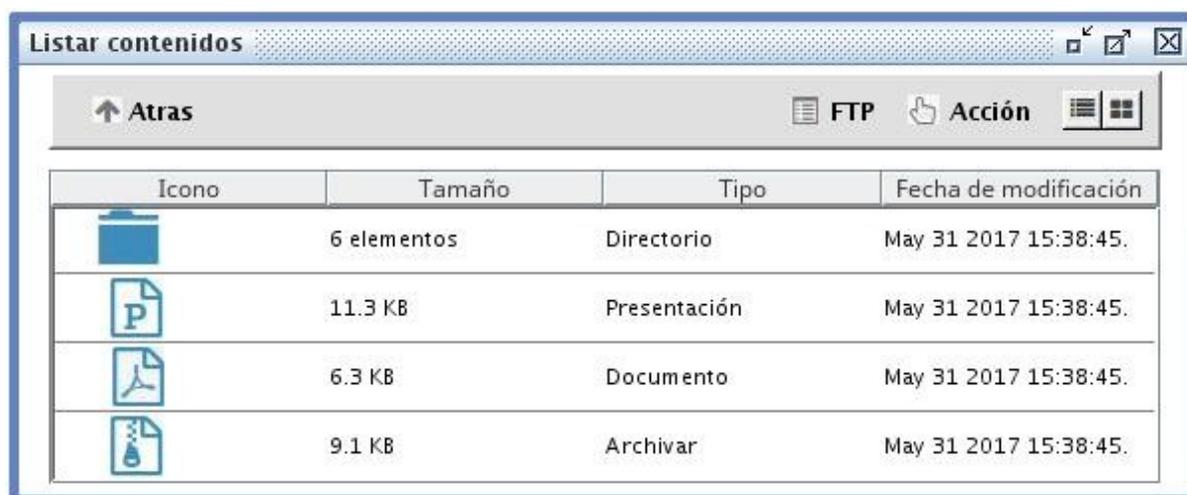


Fig. 1: PIU_Listar contenido alojado en el servidor FTP

DRP_Crear directorio en el servidor FTP

Precondiciones	El usuario debe estar autenticado en el sistema. El usuario debe poseer los permisos para Crear carpeta en el servidor FTP
-----------------------	---

Flujo de eventos

Flujo básico Crear directorio en el servidor FTP

1.	El usuario selecciona la opción Agregar carpeta
2.	El sistema debe permitir introducir siguientes datos para crear una carpeta. <ul style="list-style-type: none"> (*) Nombre Además, permite realizar las siguientes opciones: <ul style="list-style-type: none"> Aceptar Cancelar
3.	Se introducen los datos para incluir la nueva carpeta.
4.	El usuario selecciona la opción Aceptar.
5.	El sistema valida los datos.
6.	El sistema crea la nueva carpeta y regresa a la interfaz anterior.
7.	Concluye así el requisito.

Pos-condiciones

1.	Se creó la nueva carpeta satisfactoriamente.
----	--

Flujo alternativo * Cancelar

1.	El usuario selecciona la opción Cancelar.
2.	El sistema regresa la interfaz anterior. Ver documento

	FORTES_SIGIES_DRP_Listar contenidos.odt
3.	Concluye así el requisito.

Pos-condiciones

1.	No se crea la nueva carpeta.
----	------------------------------

Validaciones

	FORTES_SIGIES_Modelo_conceptual/Servidor FTP
--	--

Conceptos		Visibles en la interfaz: <ul style="list-style-type: none"> (*) Nombre Uso interno: <ul style="list-style-type: none"> id
------------------	--	---

Requisitos especiales	Seguridad: Acceso restringido
------------------------------	-------------------------------

Asuntos pendientes	NA
---------------------------	----

Prototipo elemental de interfaz gráfica de usuario



Fig. 1: PIU_Crear directorio en el servidor FTP

DRP_Modificar directorios en el servidor FTP

Precondiciones	El usuario debe estar autenticado en el sistema. El usuario debe poseer los permisos para Modificar directorios en el servidor FTP.
-----------------------	--

Flujo de eventos

Flujo básico Modificar directorios en el servidor FTP.

1.	El usuario selecciona la opción Renombrar
2.	El sistema debe permitir introducir los siguientes datos: <ul style="list-style-type: none"> (*) Nombre Además, permite realizar las siguientes opciones: <ul style="list-style-type: none"> Aceptar

Anexos

	<ul style="list-style-type: none">• Cancelar
3.	Se introducen los datos para modificar el directorio en el servidor FTP.
4.	El usuario selecciona la opción Aceptar
5.	El sistema valida los datos.
6.	El sistema modifica el directorio en el servidor FTP, regresa al listado del contenido.
7.	Concluye así el requisito.

Pos-condiciones

1.	Se modificó el directorio satisfactoriamente.
----	---

Flujo alternativo * Cancelar

1.	El usuario selecciona la opción Cancelar.
2.	El sistema regresa a la interfaz anterior. Ver documento <u>FORTES_SIGIES_DRP_Listar ccontenido.odt</u>
3.	Concluye así el requisito.

Pos-condiciones

1.	No se modifica el directorio en el servidor FTP.
----	--

Validaciones

	FORTES_SIGIES_Modelo_conceptual.odt/vinculación territorial
--	---

Conceptos		Visibles en la interfaz: <ul style="list-style-type: none">• Nombre Uso interno: <ul style="list-style-type: none">• id
------------------	--	---

Requisitos especiales	Seguridad: Acceso restringido
------------------------------	-------------------------------

Asuntos pendientes	NA
---------------------------	----

Prototipo elemental de interfaz gráfica de usuario

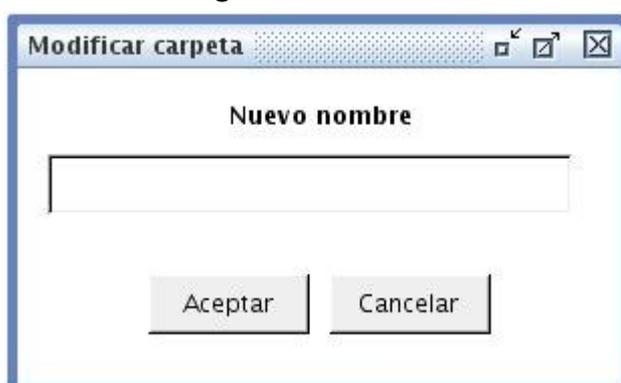


Fig. 1: PIU_Modificar directorio en el servidor FTP

DRP_Eliminar directorios en el servidor FTP

Anexos

Precondiciones	El usuario debe estar autenticado en el sistema. El usuario debe poseer los permisos para Eliminar directorios del servidor FTP.
-----------------------	---

Flujo de eventos

Flujo básico Eliminar directorios del servidor FTP.

1.	El usuario selecciona la opción Borrar
2.	El sistema debe mostrar un mensaje de alerta. Además, permite realizar las siguientes opciones: <ul style="list-style-type: none">• Aceptar• Cancelar
3.	El usuario selecciona la opción Aceptar.
4.	El sistema elimina el directorio y regresa al listado.
5.	Concluye así el requisito.

Pos-condiciones

1.	Se eliminó el directorio del servidor FTP satisfactoriamente.
----	---

Flujo alternativo * Cancelar

1.	El usuario selecciona la opción Cancelar.
2.	El sistema regresa a la interfaz anterior. Ver documento FORTES_SIGIES_DRP_Listar contenido.odt
3.	Concluye así el requisito.

Pos-condiciones

1.	No se elimina el directorio del servidor FTP
----	--

Validaciones

	FORTES_SIGIES_Modelo_conceptual.odt/vinculación territorial
--	---

Conceptos	Visibles en la interfaz: Uso interno: <ul style="list-style-type: none">• id
------------------	---

Requisitos especiales	Seguridad: Acceso restringido
------------------------------	-------------------------------

Asuntos pendientes	NA
---------------------------	----

Prototipo elemental de interfaz gráfica de usuario



Fig. 1: PIU_Eliminar carpeta en el servidor FTP

DRP_Subir archivos al servidor FTP

Precondiciones	El usuario debe estar autenticado en el sistema. El usuario debe poseer los permisos para Subir un archivo al servidor FTP
Flujo de eventos	
Flujo básico subir archivos al servidor FTP	
1.	El usuario selecciona la opción Agregar nuevo
2.	El sistema debe permitir seleccionar el archivo deseado para subir al servidor FTP Además, permite realizar las siguientes opciones: <ul style="list-style-type: none"> • Aceptar • Cancelar
3.	El usuario selecciona la opción Aceptar regresar al listado.
4.	El sistema incluye en el servidor FTP el archivo.
5.	Concluye así el requisito.
Pos-condiciones	
1.	Se incluyó el archivo en el servidor FTP satisfactoriamente.
Flujo alternativo * Cancelar	
1.	El usuario selecciona la opción Cancelar.
2.	El sistema regresa a la interfaz anterior. Ver documento FORTES_SIGIES_DRP_Listar contenidos.odt
3.	Concluye así el requisito.
Pos-condiciones	
1.	No se incluye el archivo en el servidor FTP
Validaciones	
	FORTES_SIGIES_Modelo_conceptual/Servidor FTP
Conceptos	Visibles en la interfaz: Uso interno: <ul style="list-style-type: none"> • id
Requisitos	Seguridad: Acceso restringido

especiales	
Asuntos pendientes	NA

Prototipo elemental de interfaz gráfica de usuario



Fig. 1: PIU_Subir archivos al servidor FTP

DRP_Modificar archivo en el servidor FTP

Precondiciones	El usuario debe estar autenticado en el sistema. El usuario debe poseer los permisos para Modificar archivos en el servidor FTP.
-----------------------	---

Flujo de eventos

Flujo básico Modificar archivos en el servidor FTP.

1.	El usuario selecciona la opción Renombrar
2.	El sistema debe permitir introducir los siguientes datos: <ul style="list-style-type: none"> • (*) Nombre Además, permite realizar las siguientes opciones: <ul style="list-style-type: none"> • Aceptar • Cancelar
3.	Se introducen los datos para modificar el archivo en el servidor FTP.
4.	El usuario selecciona la opción Aceptar
5.	El sistema valida los datos.
6.	El sistema modifica los archivos en el servidor FTP, regresa al listado del contenido.
7.	Concluye así el requisito.

Pos-condiciones

1.	Se modificó el archivo satisfactoriamente.
----	--

Flujo alternativo * Cancelar

1.	El usuario selecciona la opción Cancelar.
----	---

2.	El sistema regresa a la interfaz anterior.Ver documento FORTES_SIGIES_DRP_Listar ccontenido.odt
3.	Concluye así el requisito.

Pos-condiciones

1.	No se modifica el archivo en el servidor FTP.
----	---

Validaciones

	FORTES_SIGIES_Modelo_conceptual.odt/vinculación territorial
--	---

Conceptos		Visibles en la interfaz: <ul style="list-style-type: none"> • Nombre Uso interno: <ul style="list-style-type: none"> • id
------------------	--	---

Requisitos especiales	Seguridad: Acceso restringido
------------------------------	-------------------------------

Asuntos pendientes	NA
---------------------------	----

Prototipo elemental de interfaz gráfica de usuario

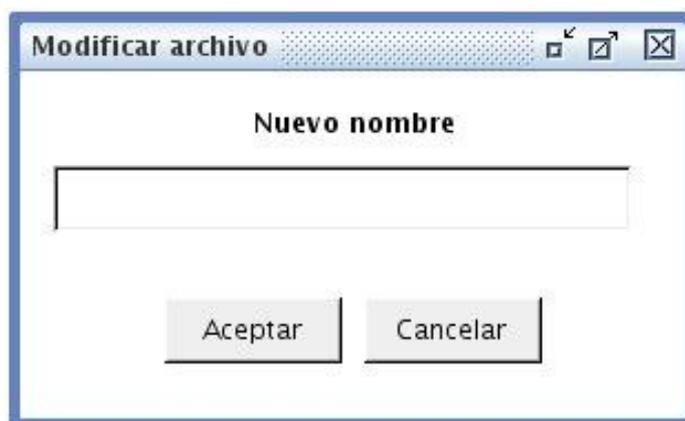


Fig. 1: PIU_Modificar archivo en el servidor FTP

DRP_Eliminar archivo del servidor FTP

Precondiciones	El usuario debe estar autenticado en el sistema. El usuario debe poseer los permisos para Eliminar archivos del servidor FTP.
-----------------------	--

Flujo de eventos

Flujo básico Eliminar archivos del servidor FTP.

1.	El usuario selecciona la opción Borrar
2.	El sistema debe mostrar un mensaje de alerta. Además, permite realizar las siguientes opciones: <ul style="list-style-type: none"> • Aceptar

Anexos

	<ul style="list-style-type: none">• Cancelar
3.	El usuario selecciona la opción Aceptar.
4.	El sistema elimina el archivo y regresa al listado.
5.	Concluye así el requisito.
Pos-condiciones	
1.	Se eliminó el archivo del servidor FTP satisfactoriamente.
Flujo alternativo * Cancelar	
1.	El usuario selecciona la opción Cancelar.
2.	El sistema regresa a la interfaz anterior.Ver documento <u>FORTES_SIGIES_DRP_Listar contenido.odt</u>
3.	Concluye así el requisito.
Pos-condiciones	
1.	No se elimina el archivo del servidor FTP
Validaciones	
	FORTES_SIGIES_Modelo_conceptual.odt/vinculación territorial
Conceptos	Visibles en la interfaz: Uso interno: <ul style="list-style-type: none">• id
Requisitos especiales	Seguridad: Acceso restringido
Asuntos pendientes	NA

Prototipo elemental de interfaz gráfica de usuario



Fig. 1: PIU_Eliminar archivos en el servidor FTP

DRP_Descargar archivos del servidor FTP

Precondiciones	El usuario debe estar autenticado en el sistema. El usuario debe poseer los permisos para Descargar archivos del servidor FTP.
Flujo de eventos	
Flujo básico Descargar archivos del servidor FTP.	
1.	El usuario selecciona el archivo a descargar
2.	El sistema realiza la descarga del archivo seleccionado.
3.	Concluye así el requisito.
Pos-condiciones	
1.	Se descargó el archivo satisfactoriamente.
Validaciones	
	FORTES_SIGIES_Modelo_conceptual/Servidor FTP
Conceptos	<p>Visibles en la interfaz:</p> <p>Uso interno:</p> <ul style="list-style-type: none"> • id
Requisitos especiales	Seguridad: Acceso restringido
Asuntos pendientes	NA

Prototipo elemental de interfaz gráfica de usuario

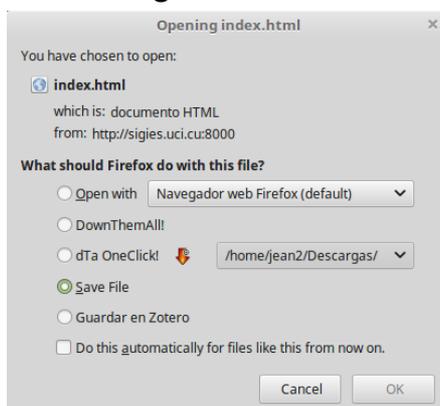


Fig. 1: PIU_Descargar archivos en el servidor FTP

DRP_Asignar rol_ FTP

Precondiciones	El usuario debe estar autenticado en el sistema con el rol Administrador DIUL. El usuario debe poseer los permisos para Asignar roles_ FTP.
-----------------------	--

Anexos

Flujo de eventos

Flujo básico Asignar rol_FTP.

1.	El usuario selecciona la acción Editar permisos
2.	El sistema debe permitir introducir y seleccionar los siguientes datos: <ul style="list-style-type: none"> • (*) Usuario FTP • (*) Tipo de navegación Además, permite realizar las siguientes opciones: <ul style="list-style-type: none"> • Aceptar • Cancelar
3.	Se introducen los datos.
4.	El usuario selecciona la opción Aceptar.
5.	El sistema valida los datos.
6.	El sistema asigna a los usuarios introducidos las acciones definidas por el tipo de navegación seleccionado.
7.	Concluye así el requisito.

Pos-condiciones

- | | |
|----|---|
| 1. | Se creó la conexión con el servidor FTP satisfactoriamente. |
|----|---|

Flujo alternativo * Cancelar

1.	El usuario selecciona la opción Cancelar.
2.	El sistema regresa a la interfaz anterior. Ver documento DRP_Listar contenidos.odt
3.	Concluye así el requisito.

Pos-condiciones

- | | |
|----|---|
| 1. | No se crea la nueva conexión con el servidor FTP. |
|----|---|

Validaciones

Modelo_conceptual/Servidor FTP

Conceptos	Servidor FTP	Visibles en la interfaz: <ul style="list-style-type: none"> • (*)Usuario FTP • (*) Tipo de navegación Uso interno: <ul style="list-style-type: none"> • id
------------------	--------------	---

Requisitos especiales	Seguridad: Acceso restringido
------------------------------	-------------------------------

Asuntos pendientes	NA
---------------------------	----

Prototipo elemental de interfaz gráfica de usuario

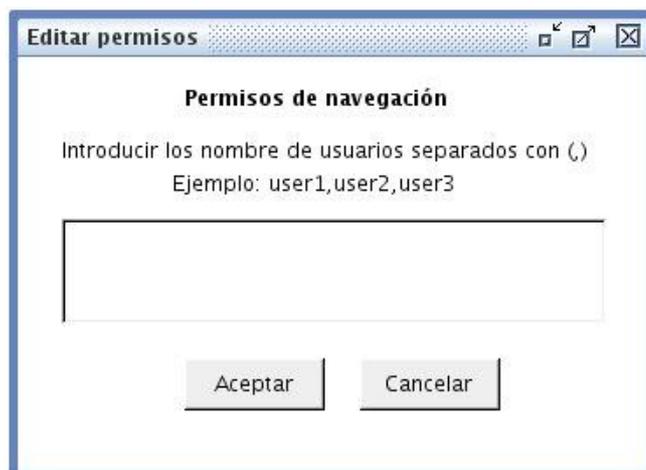


Fig. 1: PIU_Asignar rol_FTP

DRP_Mostrar permisos para la conexión

Precondiciones	El usuario debe estar autenticado en el sistema. El usuario debe poseer los permisos para ver acciones a realizar sobre el contenido alojado en el servidor FTP.	
Flujo de eventos		
Flujo básico Mostrar permisos para la conexión		
1.	El usuario selecciona la acción Mostrar permisos	
2.	El sistema debe mostrar la lista de acciones posibles para el usuario autenticado. Además, permite realizar las siguientes opciones: <ul style="list-style-type: none"> • Salir 	
3.	El usuario selecciona la opción Salir.	
4.	Concluye así el requisito.	
Pos-condiciones		
1.	Se mostró las acciones realizar por el usuario autenticado para la conexión actual satisfactoriamente.	
Validaciones		
	Modelo_conceptual/Servidor FTP	
Conceptos	Servidor FTP	Visibles en la interfaz: Uso interno: <ul style="list-style-type: none"> • id
Requisitos especiales	Seguridad: Acceso restringido	
Asuntos pendientes	NA	

Prototipo elemental de interfaz gráfica de usuario

DRP_Mostrar información de los recursos en el servidor FTP

Precondiciones	El usuario debe estar autenticado en el sistema. El usuario debe poseer los permisos para ver la información de los recursos existentes en el servidor FTP.
Flujo de eventos	
Flujo básico Mostrar información de los recursos en el servidor FTP	
1.	El usuario selecciona la conexión deseada
2.	El sistema debe mostrar junto al listado de recursos la información de cada recurso existente en el servidor FTP.
3.	Concluye así el requisito.
Pos-condiciones	
1.	Se mostró la información satisfactoriamente.
Validaciones	
FORTES_SIGIES_Modelo_conceptual.odt/Servidor FTP	
Conceptos	Visibles en la interfaz: Uso interno: <ul style="list-style-type: none"> id
Requisitos especiales	Seguridad: Acceso restringido
Asuntos pendientes	NA

Prototipo elemental de interfaz gráfica de usuario



Fig. 1: PIU_Mostrar información de los recursos en el servidor FTP

Anexo 2: Diagramas de clases del análisis

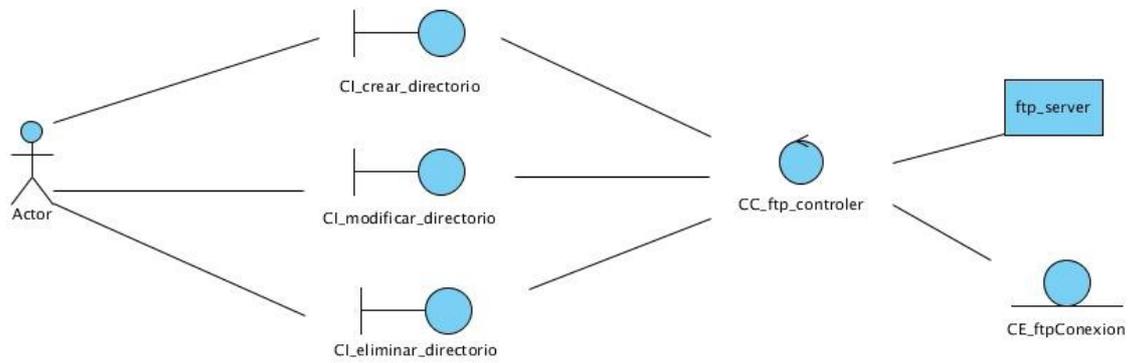


Fig 17: DCA_Gestionar directorios

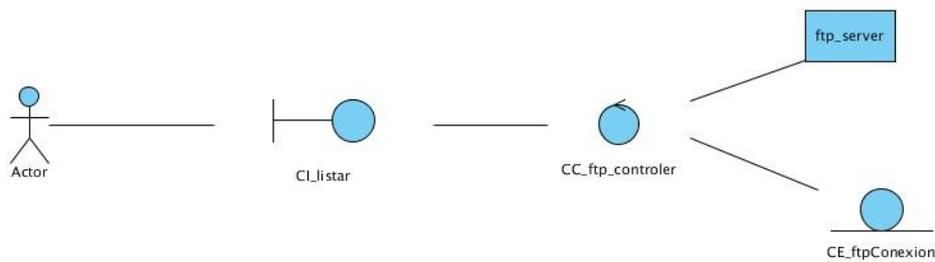


Fig 18: DCA_Listar contenido alojado en el servidor FTP

Anexos



Fig 19: Subir un archivo al servidor FTP



Fig 20: Descargar un archivo del servidor FTP



Fig 21: Eliminar archivos del servidor FTP

Anexos



Fig 22: Modificar un archivo en el servidor FTP

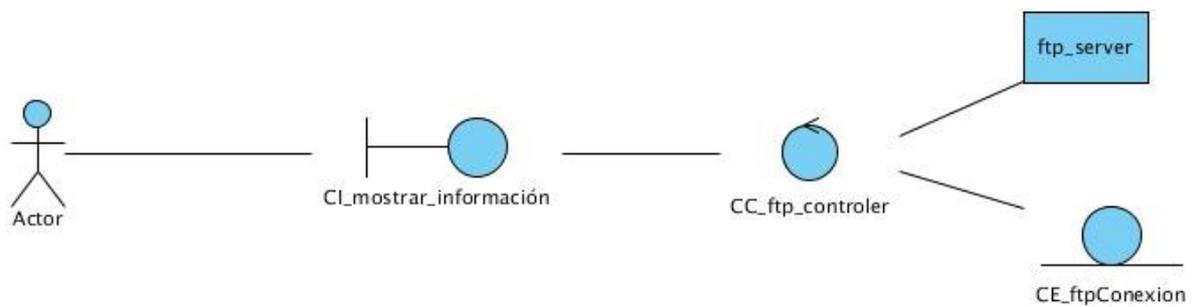


Fig 23: Mostrar información de los recursos en el servidor FTP



Fig 24: Asignar rol FTP



Fig 25: Mostrar permisos para la conexión

Anexo 3: Diagramas de colaboración del análisis

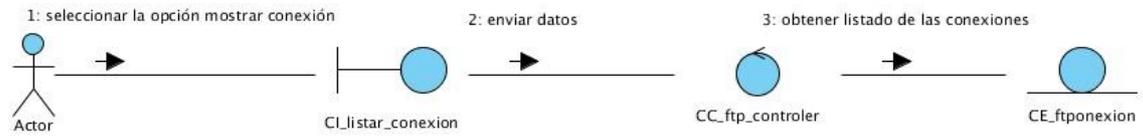


Fig 26: Listar conexiones con el servidor FTP

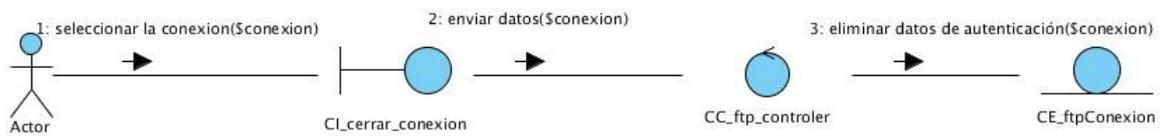


Fig 27: Eliminar conexiones con el servidor FTP

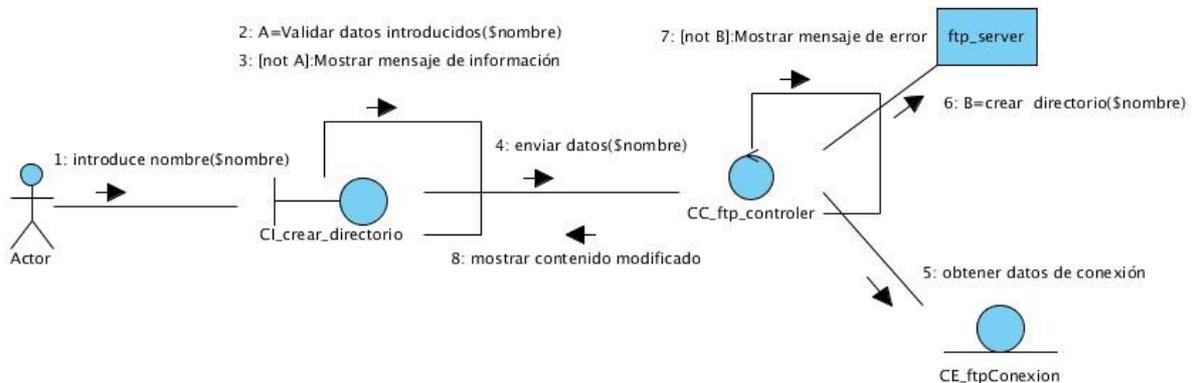


Fig 28: Crear directorios en el servidor FTP

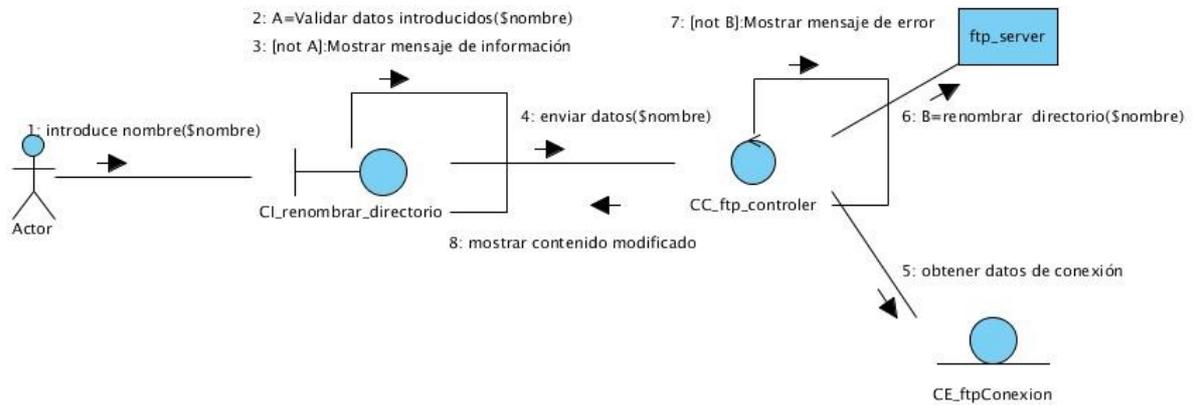


Fig 29: Modificar directorios en el servidor FTP

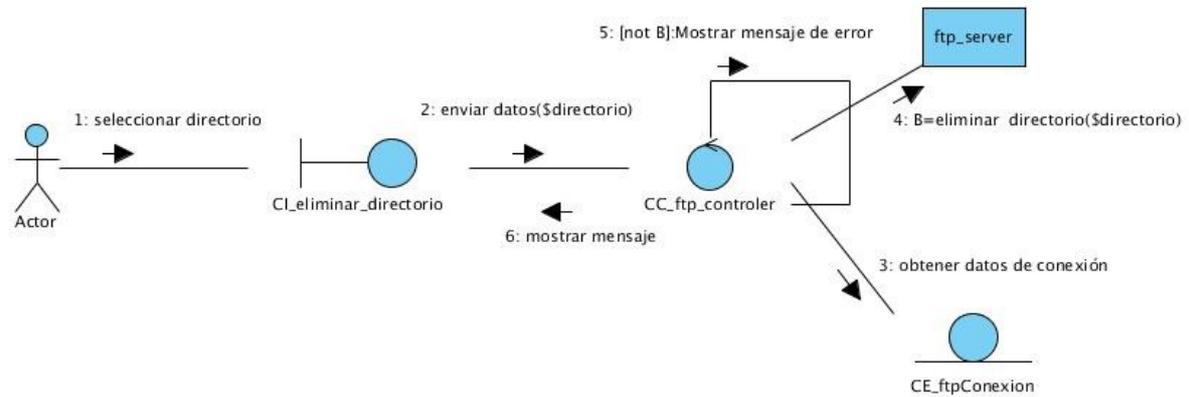


Fig 30: Eliminar directorios del servidor FTP

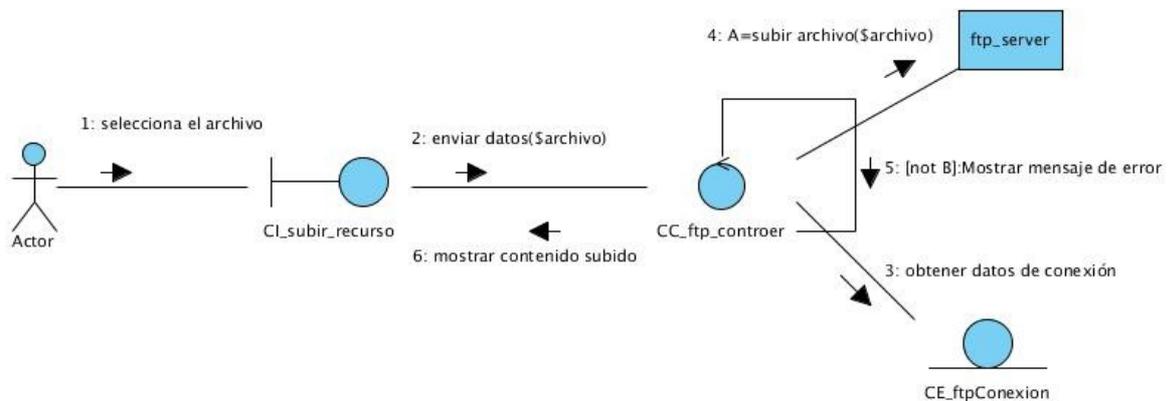


Fig 31: Subir recursos al servidor FTP

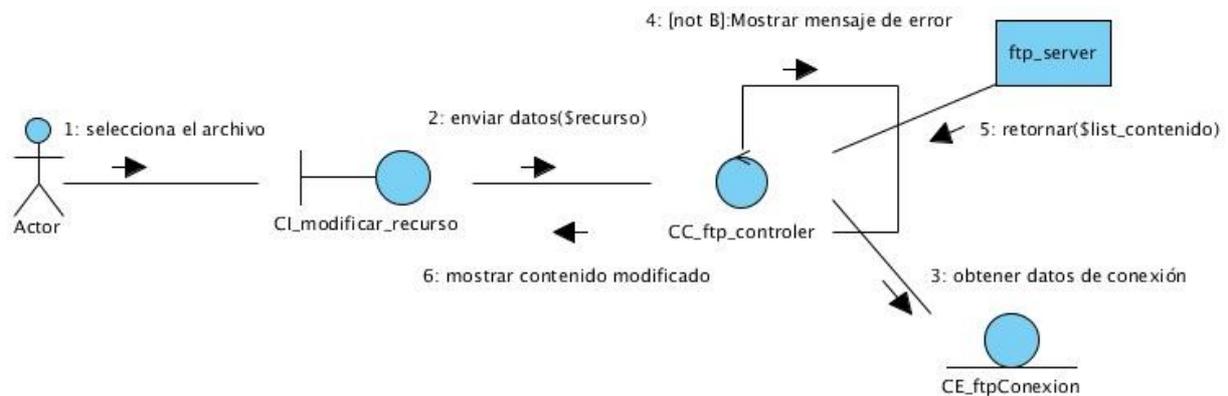


Fig 32: Modificar un archivo en el servidor FTP

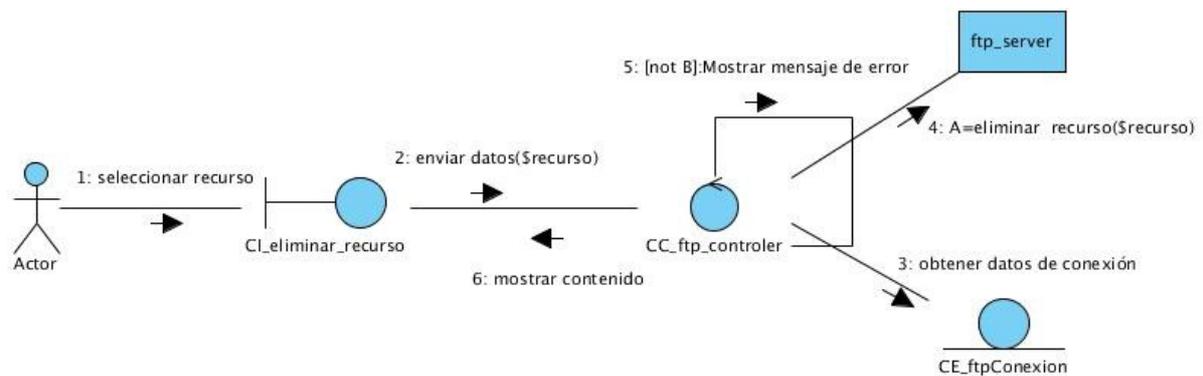


Fig 33: Eliminar un archivo del servidor FTP

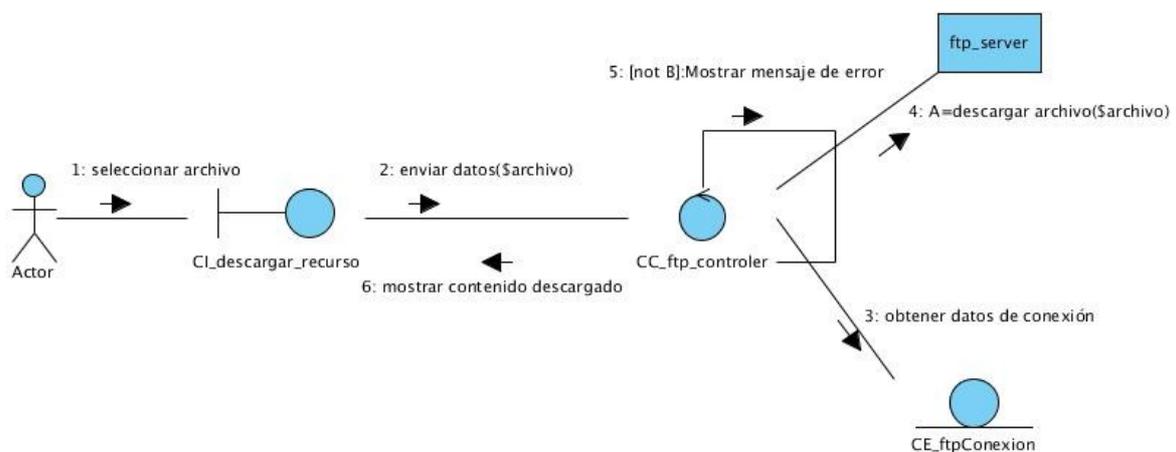


Fig 34: Descargar un archivo del servidor FTP

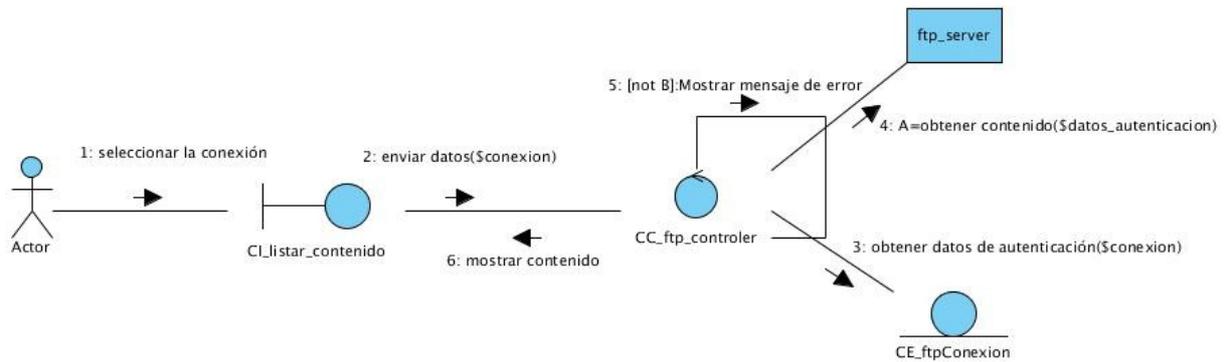


Fig 35: Listar contenido alojado en el servidor FTP

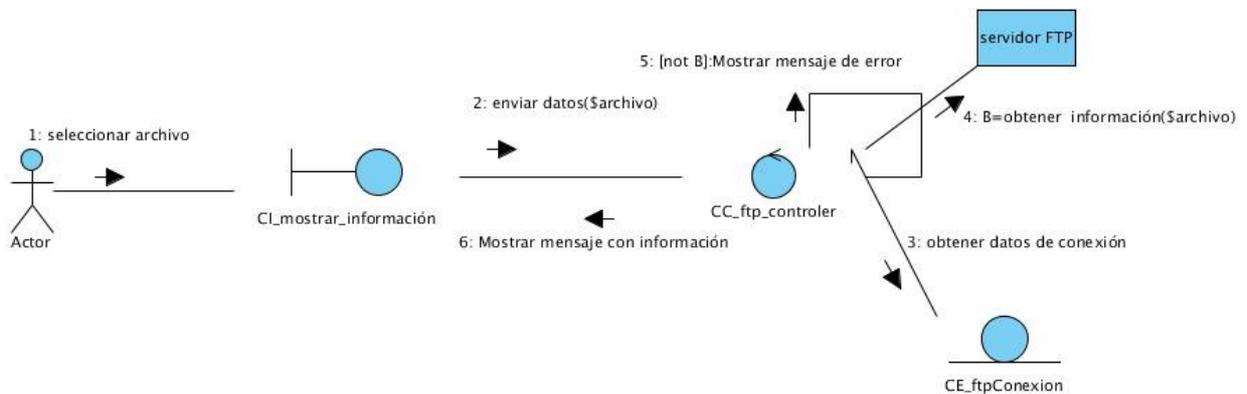


Fig 36: Mostrar información de los recursos en el servidor FTP

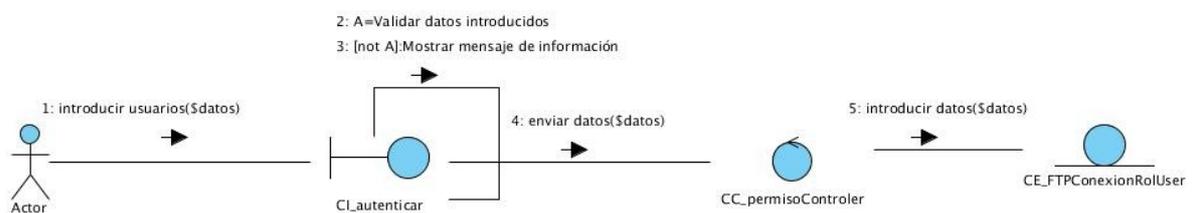


Fig 37: Asignar rol FTP

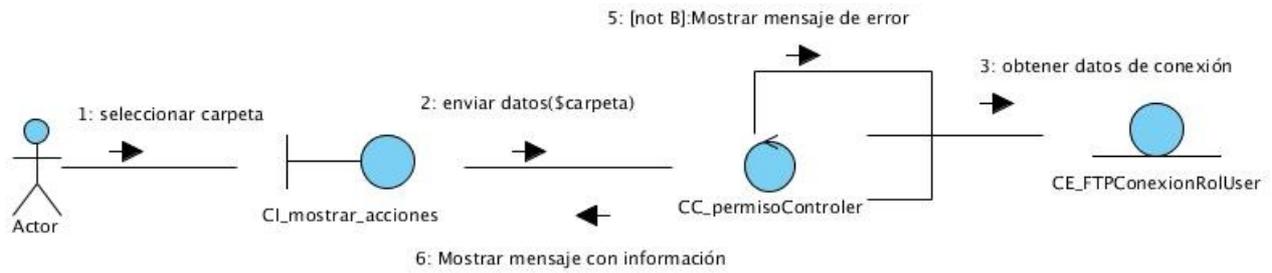


Fig 38: Mostrar permisos para la conexión

Anexo 4: Diagramas de clases del diseño

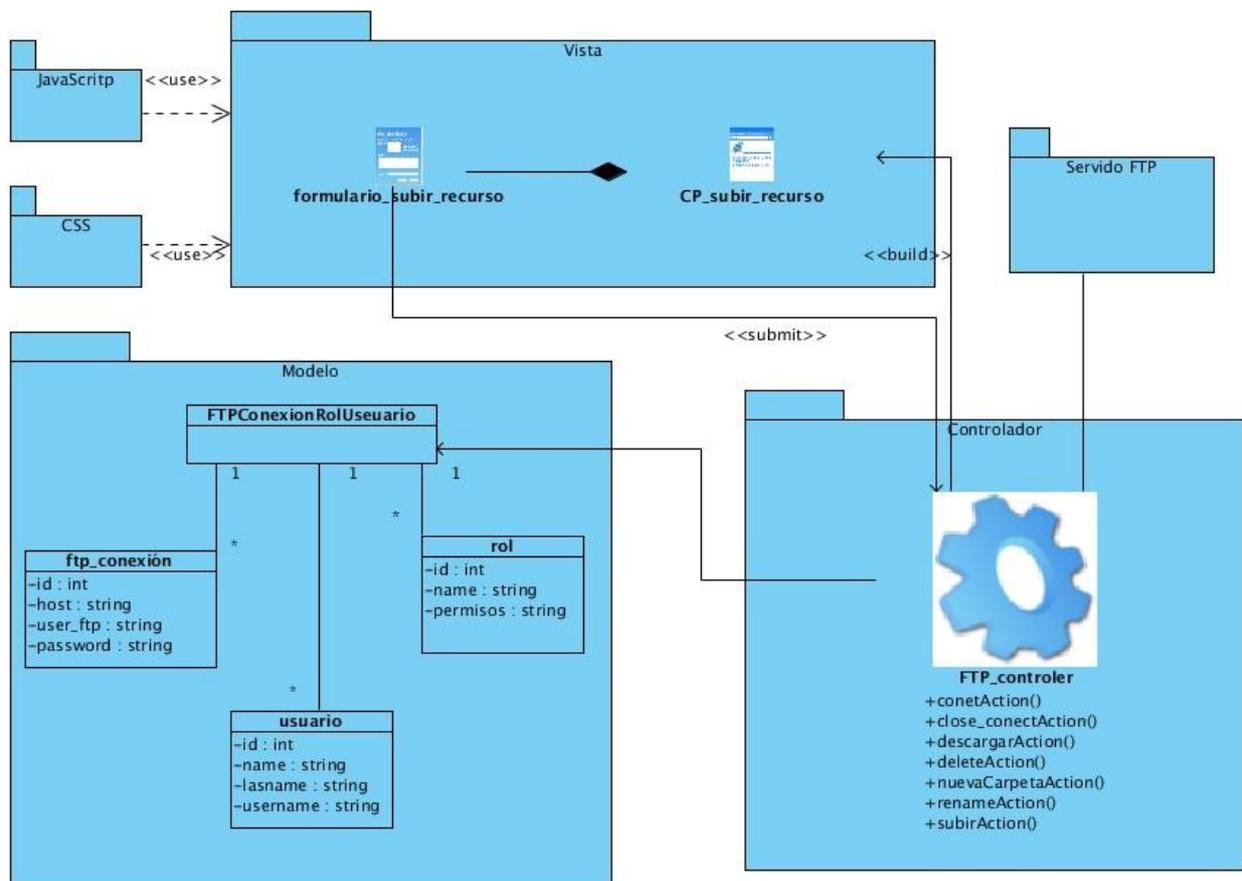


Fig 39: Subir archivos al servidor FTP

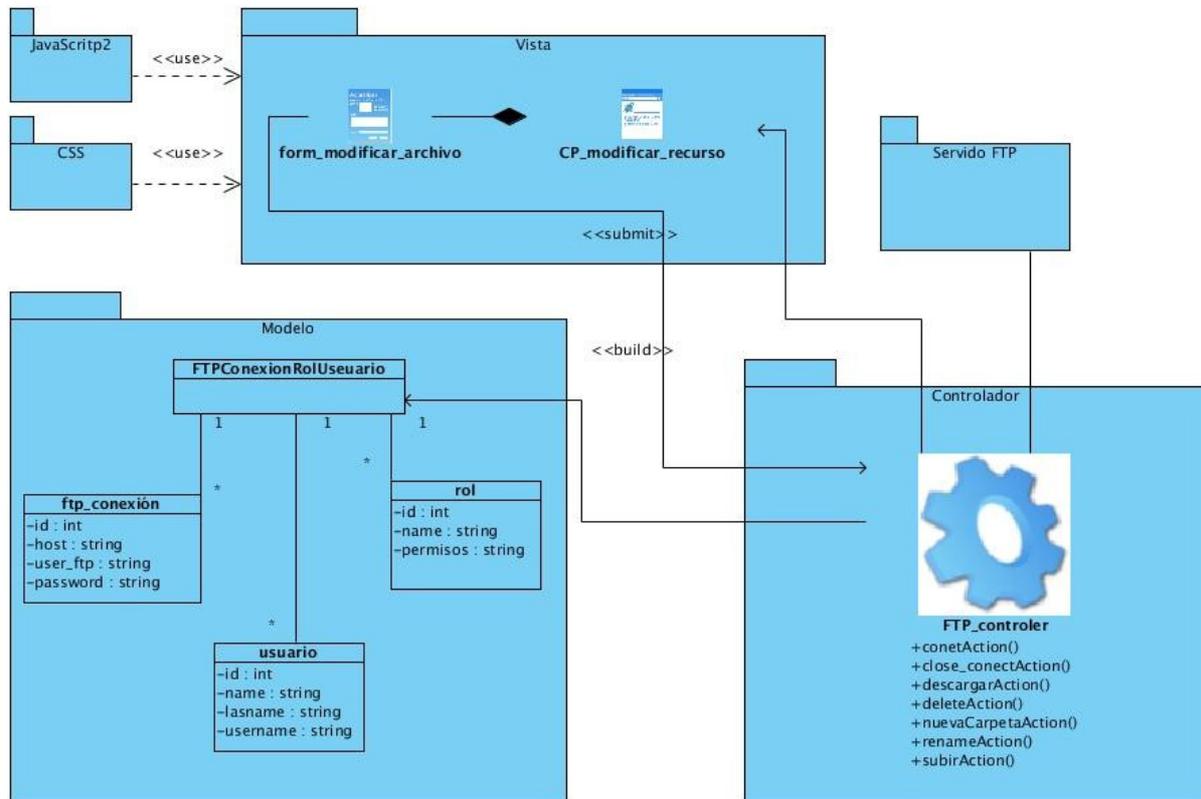


Fig 40: Modificar archivos en el servidor FTP

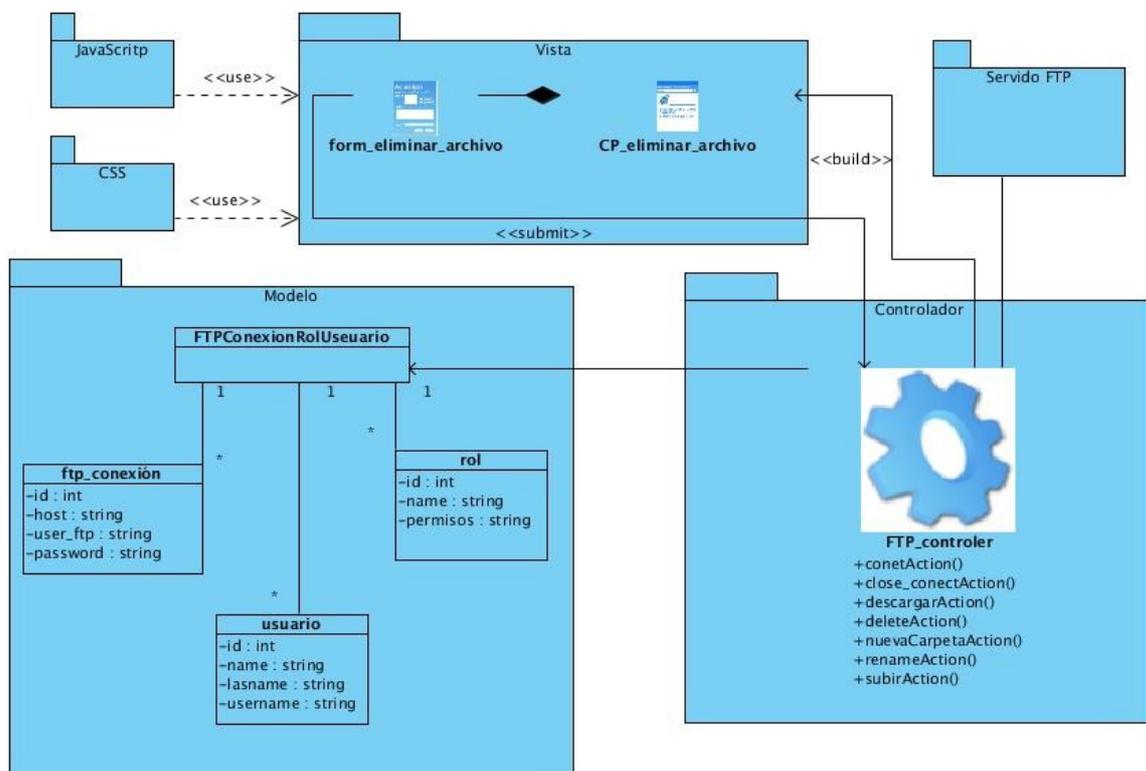


Fig 41: Eliminar archivos del servidor FTP

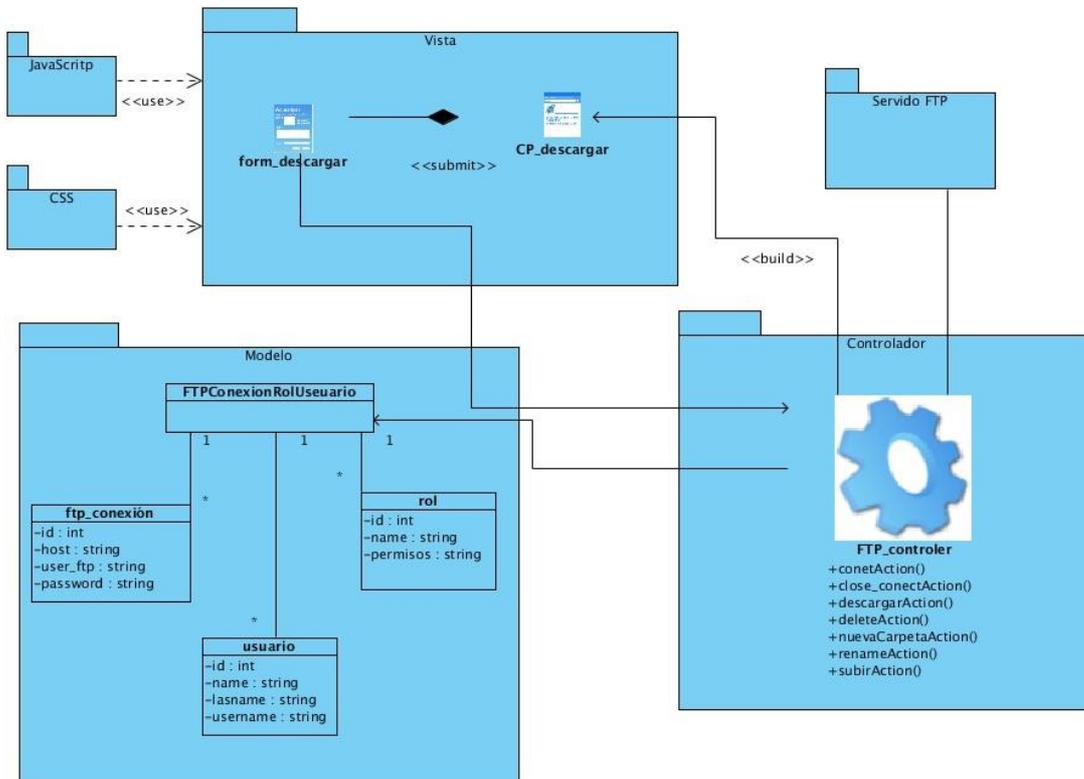


Fig 42: Descargar archivos del servidor FTP

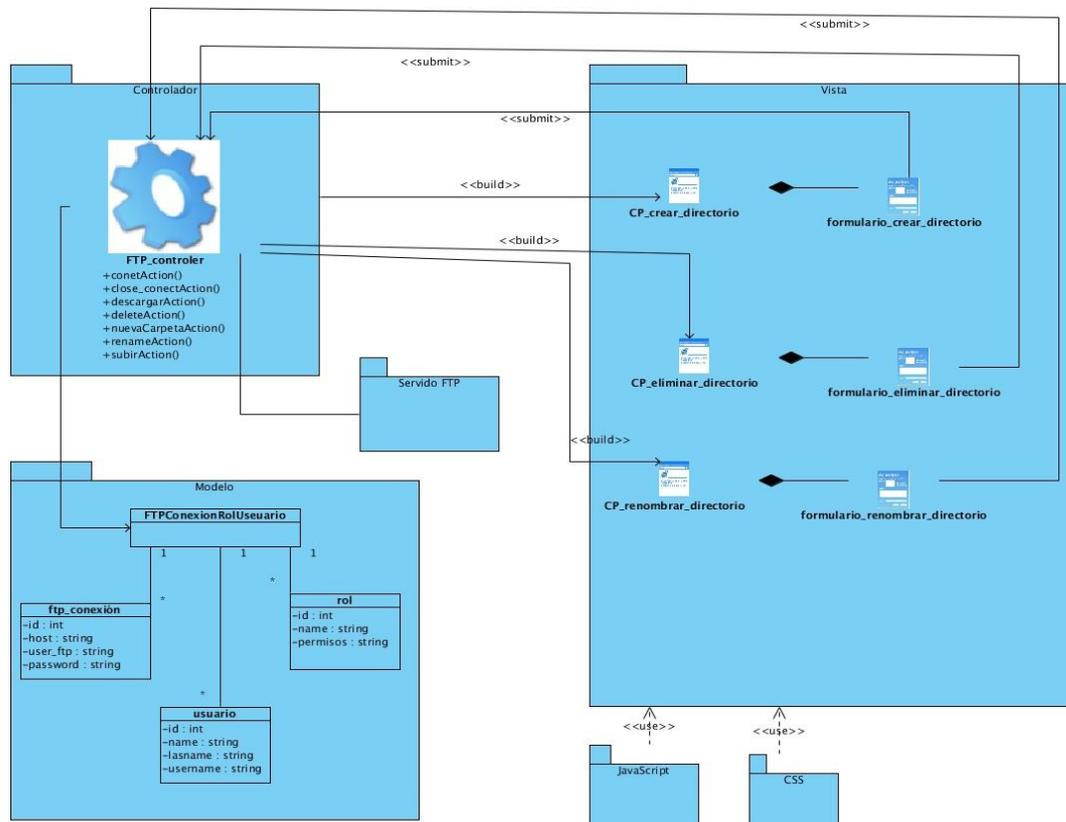


Fig 43: Gestionar directorios en el servidor FTP

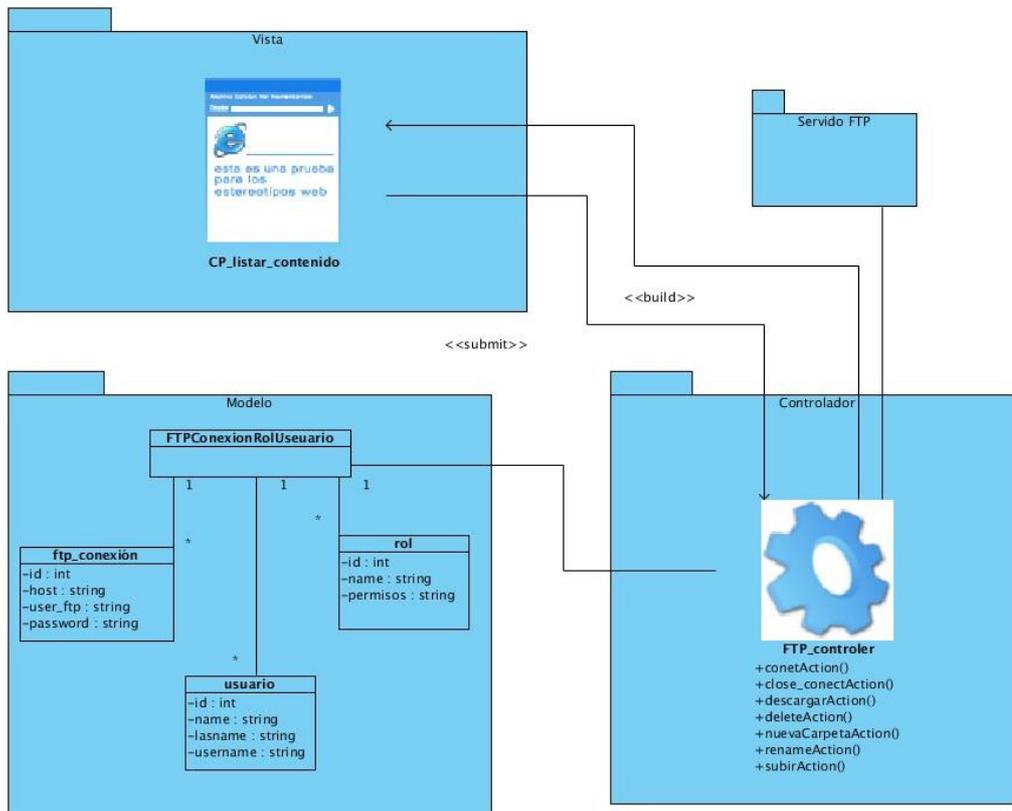


Fig 44: Listar el contenido en el servidor FTP

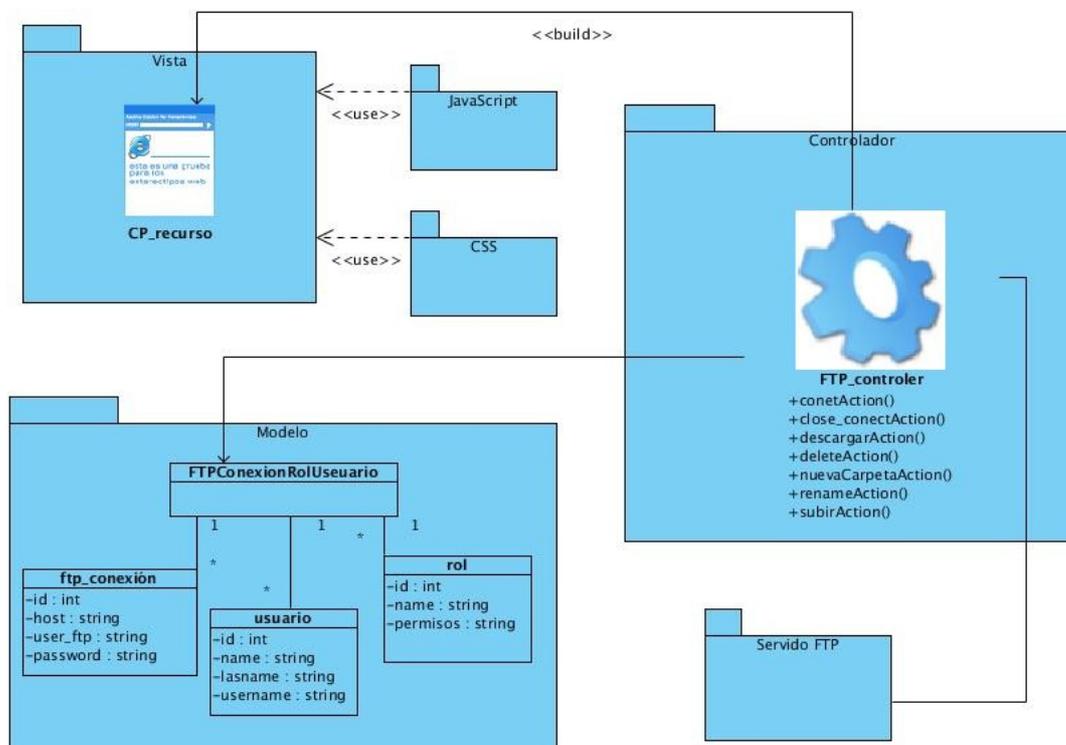


Fig 45: Mostrar información de los recursos en el servidor FTP

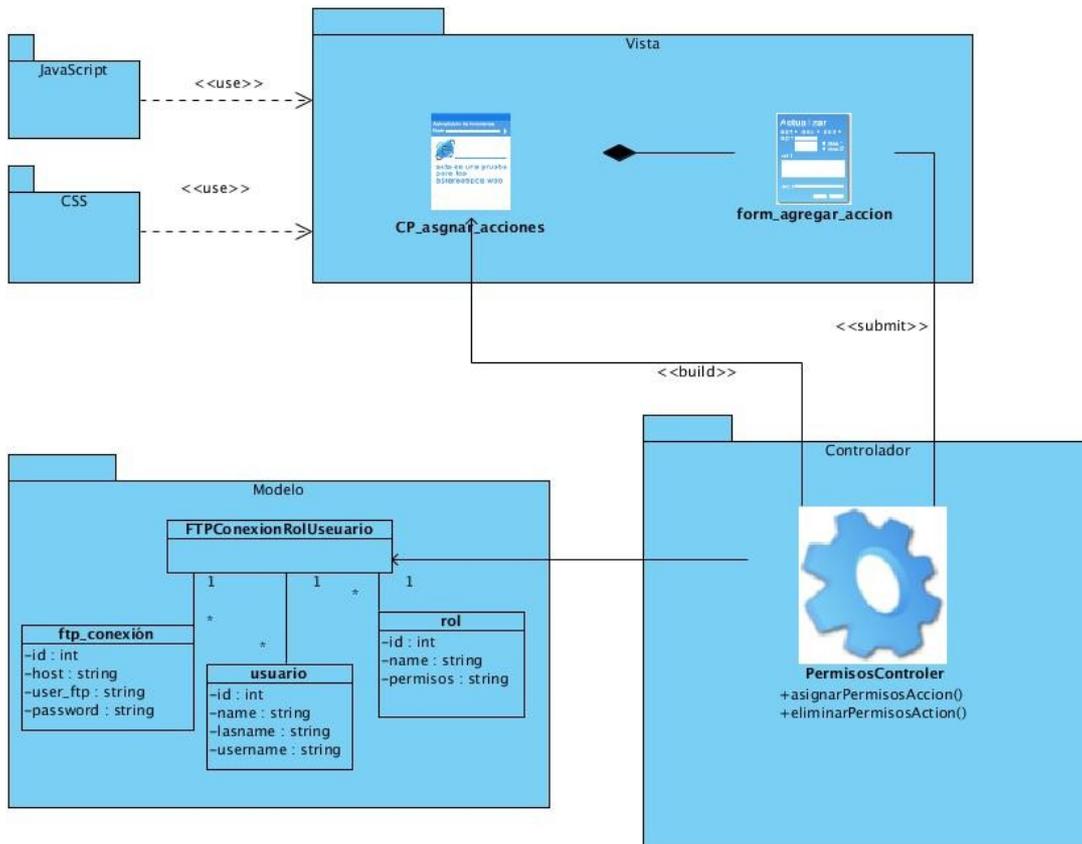


Fig 46: Asignar rol FTP

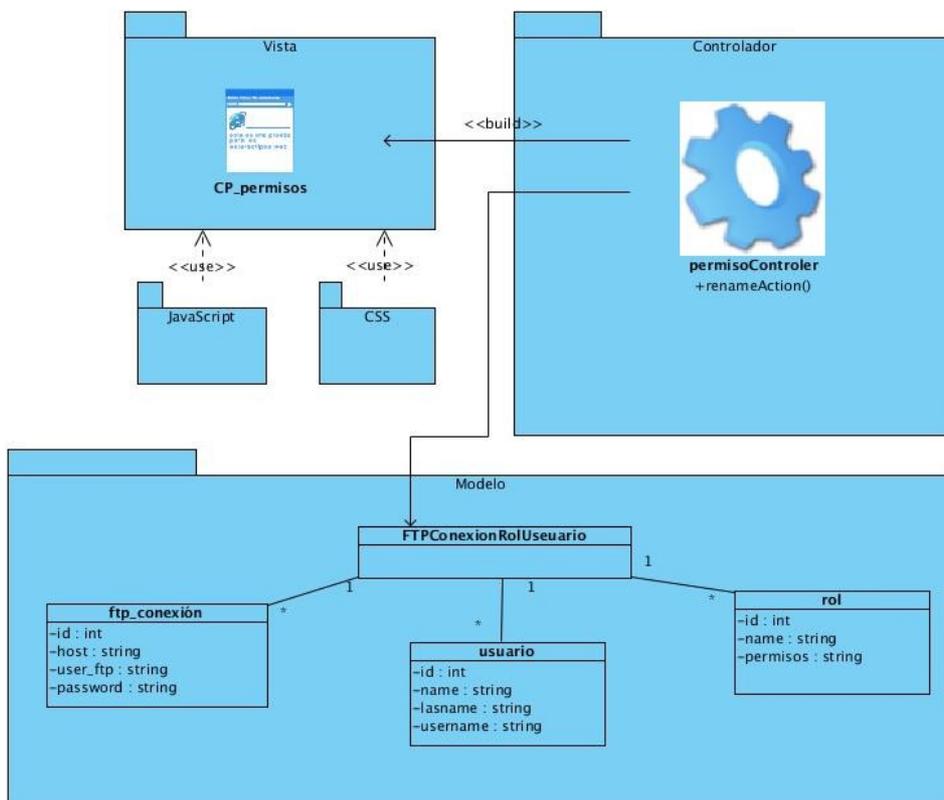


Fig 47: Mostrar permisos para la conexión

Anexo 5: Diagramas de secuencias del diseño

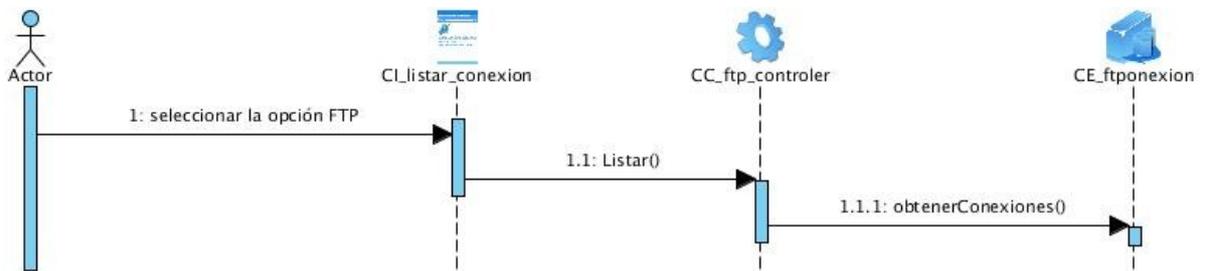


Fig 48: Listar conexiones con el servidor FTP

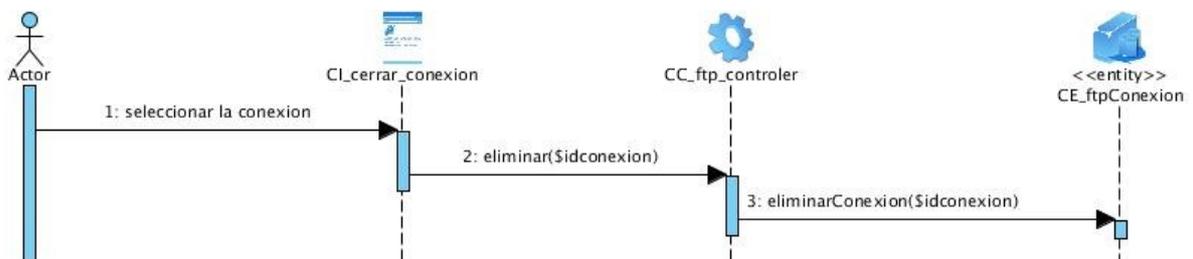


Fig 49: Eliminar conexiones con el servidor FTP

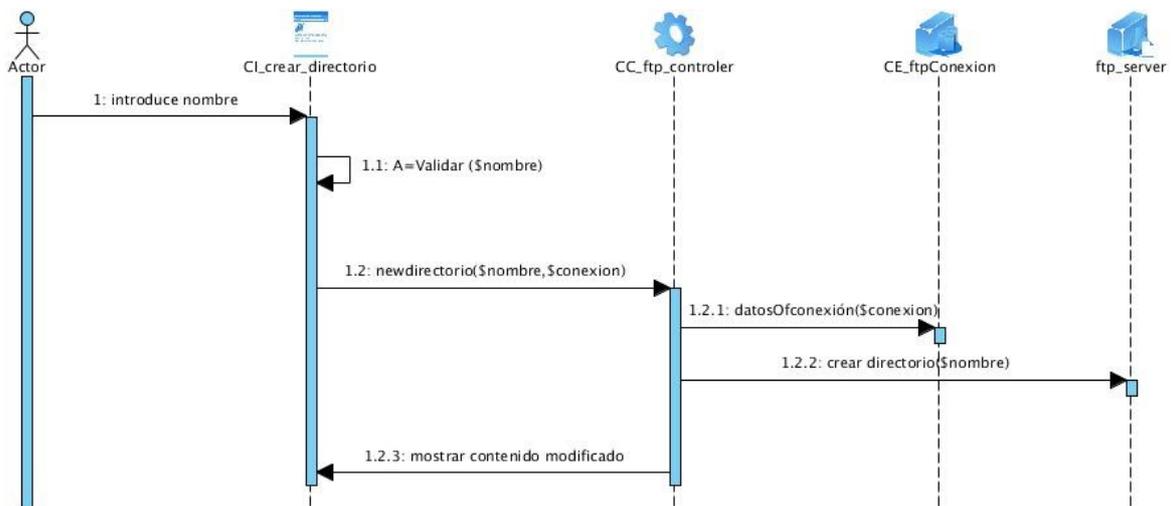


Fig 50: Crear directorios en el servidor FTP

Anexos

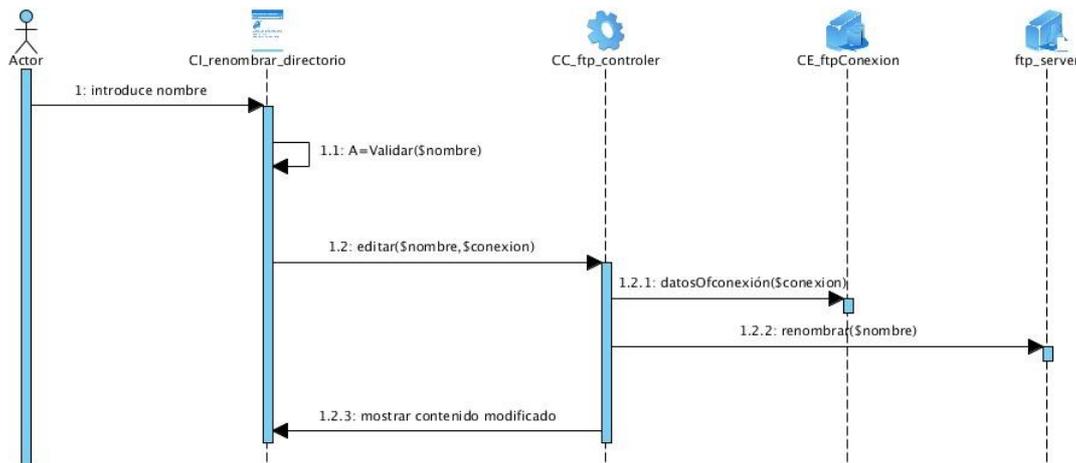


Fig 51: Modificar directorios en el servidor FTP

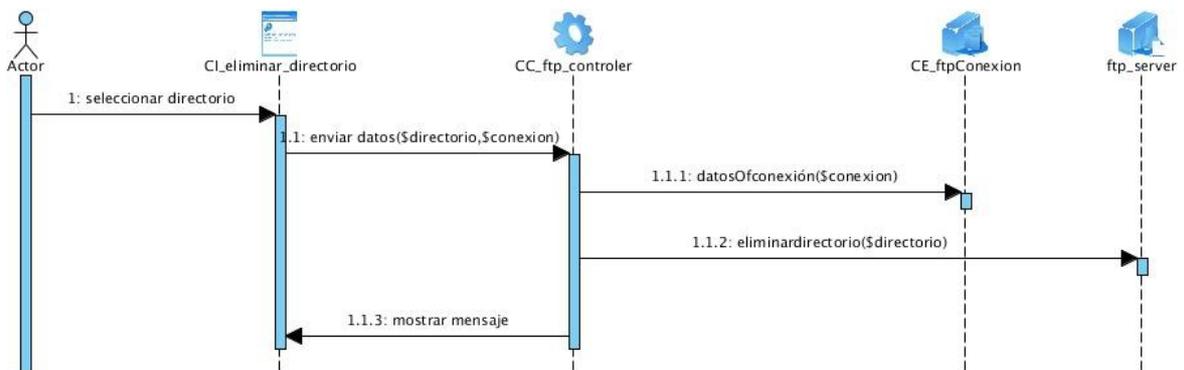


Fig 52: Eliminar directorios del servidor FTP

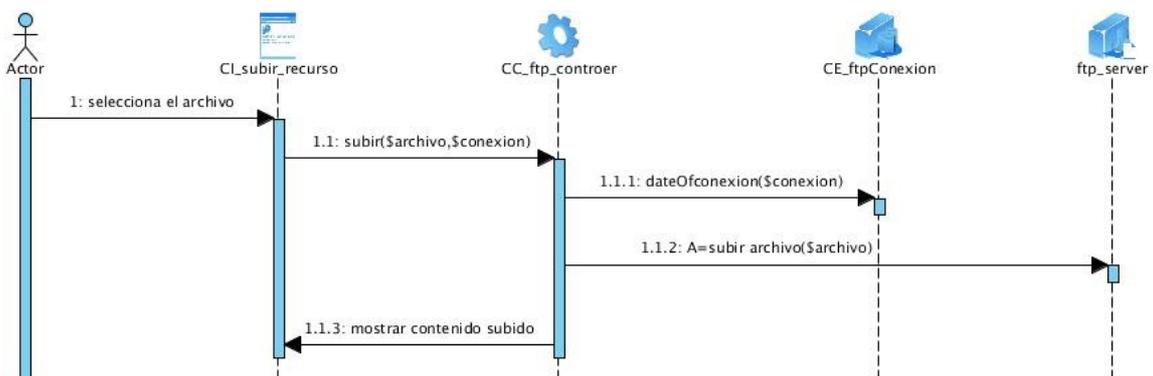


Fig 53: Subir recursos al servidor FTP

Anexos

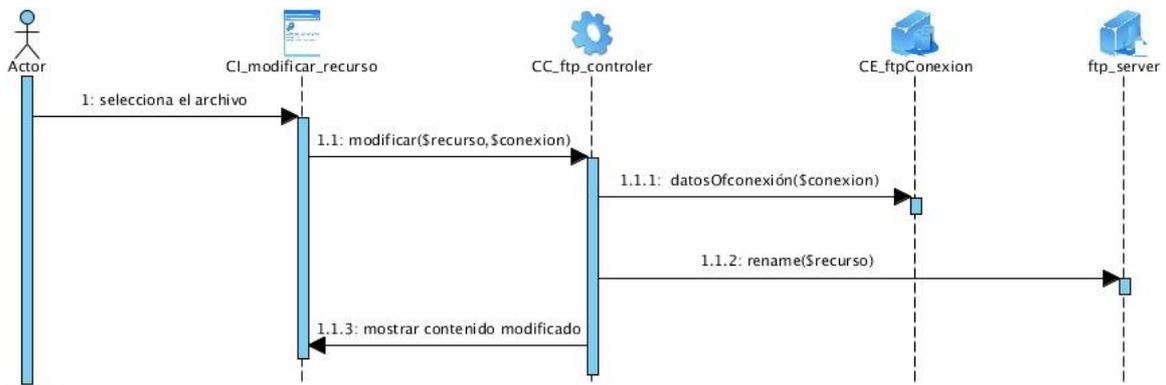


Fig 54: Modificar archivos en el servidor FTP

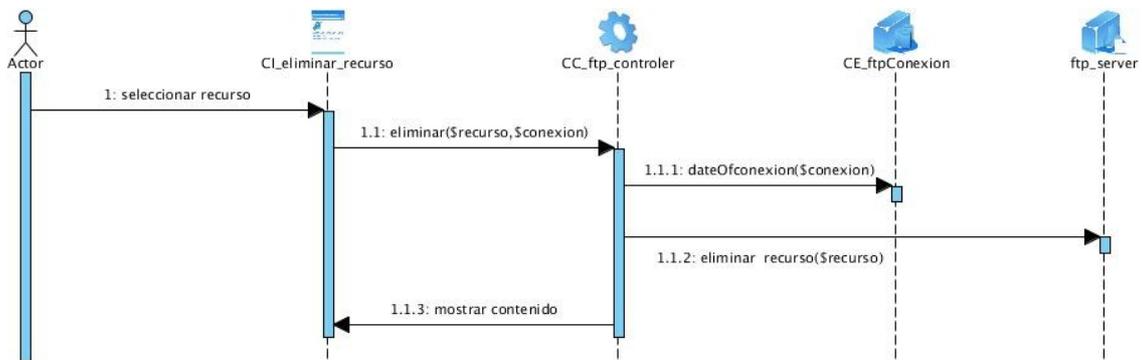


Fig 55: Eliminar archivos del servidor FTP

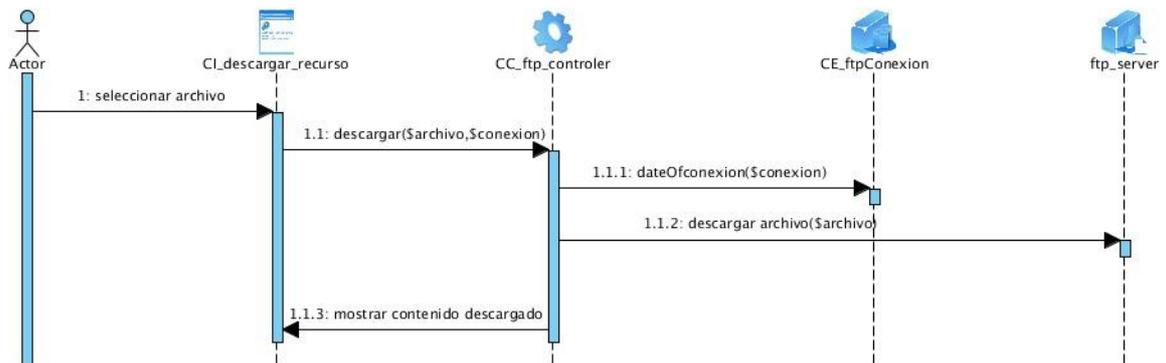


Fig 56: Descargar archivos del servidor FTP

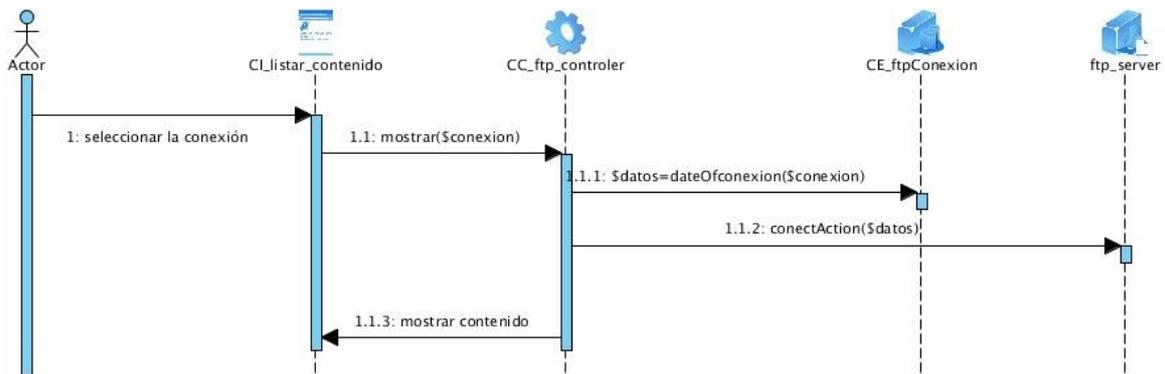


Fig 57: Listar el contenido del servidor FTP

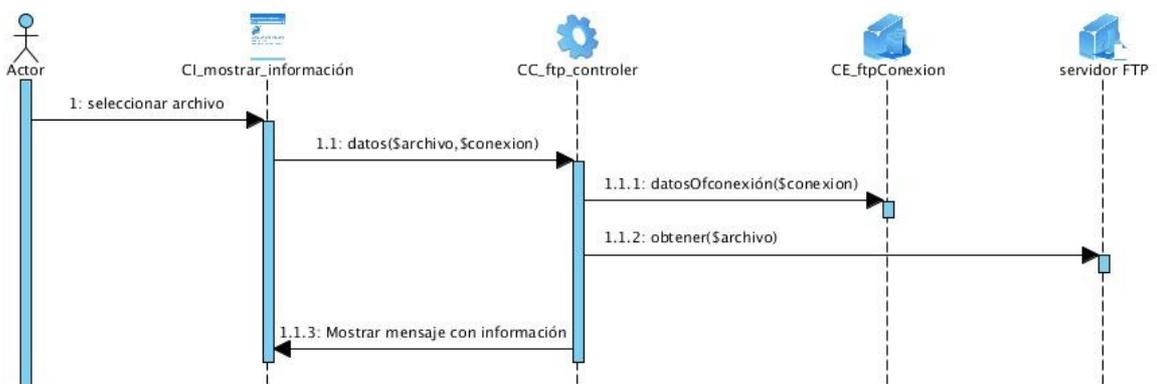


Fig 58: Mostrar información de los recursos en el servidor FTP

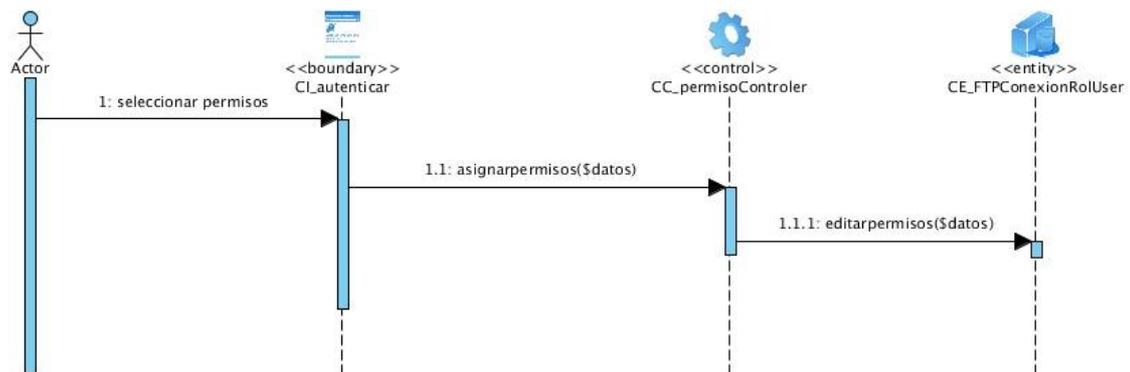


Fig 59: Asignar rol FTP

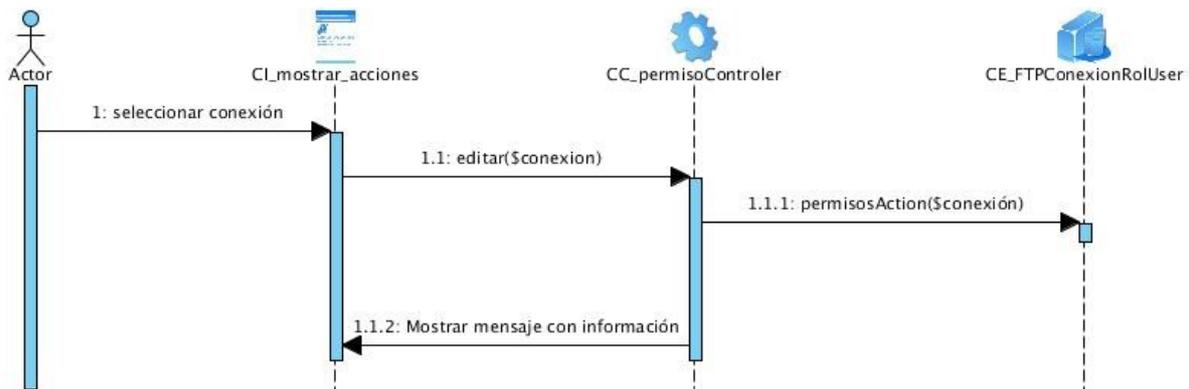


Fig 60: Mostrar permisos para la conexión

Anexo 6: Diseños de casos de pruebas

Listar conexiones abiertas con el servidor FTP.

Descripción general: permitir listar las conexiones establecidas con el servidor FTP.						
Condiciones de ejecución: para listar las conexiones hay que estar autenticado en el sistema.						
Nombre de la sección: SC1 Listar conexiones con el servidor FTP						
Escenarios	Descripción				Respuesta del sistema	Flujo central
EC1.1 Listar las conexiones con el servidor FTP.	El usuario selecciona la opción FTP en el menú				El sistema muestra un listado con las conexiones establecidas. Y además, permite seleccionar las siguientes opciones: Filtrar Editar	Principal/FTP/Conexiones
EC1.2 Opción Filtrar	El usuario selecciona los datos las opciones por las cuales se				El sistema muestra el contenido resultante de la búsqueda	Principal/FTP/Conexiones/Filtrar

Anexos

	realiza el filtrado				aplicando los filtros seleccionados	
EC1.3 Opción Editar	El usuario selecciona la opción Editar				El sistema permite editar los datos de la conexión. Además, permite realizar las opciones: <ul style="list-style-type: none"> • Cancelar • Aceptar y regresar al listado 	Principal/FTP/Conexiones/Editar
EC1.4 Opción Cancelar	El usuario selecciona la opción cancelar				El sistema no realiza la modificación y regresa al listado	Principal/Conexiones/Editar/Cancelar
EC1.5 opción Aceptar y regresar al listado	El usuario selecciona la opción Aceptar y regresar al listado				El sistema realiza las modificaciones y regresa al listado.	Principal/Conexiones/Editar/Aceptar y regresar al listado.

Eliminar conexiones con el servidor FTP.

Descripción general: permitir eliminar las conexiones establecidas con el servidor FTP.						
Condiciones de ejecución: para eliminar una conexión hay que estar autenticado en el sistema con rol ADMIN_DIUL						
Nombre de la sección: SC1 Eliminar conexiones con el servidor FTP						
Escenarios	Descripción				Respuesta del sistema	Flujo central
EC1.1 Eliminar las conexiones con el servidor	El usuario selecciona la opción Eliminar				El sistema muestra un mensaje de confirmación. Además, permite	Principal/FTP/Conexiones/Eliminar

Anexos

FTP.					seleccionar las siguientes opciones: <ul style="list-style-type: none"> • Cancelar • Aceptar 	
EC1.2 Opción Cancelar	El usuario selecciona la opción Cancelar en el mensaje de confirmación				El sistema no elimina la conexión.	Principal/FTP/Conexiones/Eliminar/Cancelar
EC1.3 Opción Aceptar	El usuario selecciona la opción Aceptar en el mensaje de confirmación				El sistema elimina la conexión.	Principal/FTP/Conexiones/Eliminar/Aceptar

Crear directorios en el servidor FTP.

Descripción general: permitir crear directorios en el servidor FTP.				
Condiciones de ejecución: para crear directorios en el servidor FTP debe poseer permisos sobre el servidor medios, avanzados o administrativos.				
Nombre de la sección: SC1 Crear directorios en el servidor FTP				
Escenarios	Descripción	Nombre de la carpeta	Respuesta del sistema	Flujo central
EC1.1 Opción Crear carpeta	El usuario selecciona la opción Crear carpeta		El sistema permite introducir el nombre de la carpeta. Además, permite seleccionar las siguientes opciones: <ul style="list-style-type: none"> • Cancelar • Aceptar 	Principal/FTP/Mostrar/Crear carpeta
EC1.2 El usuario inserta los datos	El usuario especifica el nombre del directorio y ejecuta la opción aceptar.	v	El sistema valida los datos introducidos y crea el directorio.	Principal/FTP/Mostrar/Crear carpeta

Anexos

EC1.3 Opción Cancelar	El usuario selecciona la opción Cancelar		El sistema no crea el directorio.	Principal/FTP/ Mostrar/Crear carpeta/Cancelar
EC1.4 Opción Aceptar	El usuario selecciona la opción Aceptar		El sistema crea el directorio.	Principal/FTP/ Mostrar/Crear carpeta/Aceptar
EC1.5 Datos incorrectos	Existen datos incorrectos.		El sistema muestra un mensaje de información.	Principal/FTP/ Mostrar/Crear carpeta

Descripción de variables.

No	Nombre de campo	Clasificación	Valor nulo	Descripción
1	Nombre de la carpeta	Campo de texto	No	Campo de texto que representa el nombre del nuevo directorio. Ejemplo 1: namedir

Modificar directorios en el servidor FTP.

Descripción general: permitir modificar directorios en el servidor FTP.				
Condiciones de ejecución: para modificar directorios en el servidor FTP debe poseer permisos sobre el servidor medios, avanzados o administrativos.				
Nombre de la sección: SC1 Modificar directorios en el servidor FTP				
Escenarios	Descripción	Nombre de la carpeta	Respuesta del sistema	Flujo central
EC1.1 Opción Modificar carpeta	El usuario selecciona la opción modificar carpeta		El sistema permite introducir el nombre de la carpeta. Además, permite seleccionar las siguientes opciones: <ul style="list-style-type: none"> • Cancelar • Aceptar 	Principal/FTP/ Mostrar/Modificar carpeta
EC1.2 El usuario inserta los	El usuario especifica el nombre del directorio y ejecuta la	v	El sistema valida los datos introducidos y Modifica el directorio.	Principal/FTP/ Mostrar/ Modificar

Anexos

datos	opción aceptar.			carpeta
EC1.3 Opción Cancelar	El usuario selecciona la opción Cancelar		El sistema no Modifica el directorio.	Principal/FTP/ Mostrar/ Modificar carpeta/Canc elar
EC1.4 Opción Aceptar	El usuario selecciona la opción Aceptar		El sistema Modifica el directorio.	Principal/FTP/ Mostrar/ Modificar carpeta/Acept ar
EC1.5 Datos incorrectos	Existen datos incorrectos.		El sistema muestra un mensaje de información.	Principal/FTP/ Mostrar/ Modificar carpeta

Descripción de variables.

No	Nombre de campo	Clasificación	Valor nulo	Descripción
1	Nuevo nombre	Campo de texto	No	Campo de texto que representa el nombre nuevo del directorio. Ejemplo 1: newname

Eliminar directorios del servidor FTP.

Descripción general: permitir eliminar directorios en el servidor FTP.				
Condiciones de ejecución: para modificar directorios en el servidor FTP debe poseer permisos sobre el servidor avanzados o administrativos.				
Nombre de la sección: SC1 Eliminar directorios en el servidor FTP				
Escenarios	Descripción		Respuesta del sistema	Flujo central
EC1.1 Opción Eliminar carpeta	El usuario selecciona la opción Eliminar carpeta		El sistema muestra un mensaje de alerta. Además, permite seleccionar las siguientes opciones:	Principal/FTP/ Mostrar/ Eliminar carpeta

Anexos

			<ul style="list-style-type: none"> • Cancelar • Aceptar 	
EC1.2 Opción Cancelar	El usuario selecciona la opción Cancelar		El sistema no Elimina el directorio.	Principal/FTP/ Mostrar/ Eliminar carpeta/Canc elar
EC1.3 Opción Aceptar	El usuario selecciona la opción Aceptar		El sistema Elimina el directorio.	Principal/FTP/ Mostrar/ Eliminar carpeta/Acept ar

Subir archivos al servidor FTP.

Descripción general: permitir subir archivos al servidor FTP.				
Condiciones de ejecución: para subir archivos al servidor FTP debe poseer permisos sobre el servidor medios, avanzados o administrativos.				
Nombre de la sección: SC1 Subir archivos al servidor FTP				
Escenarios	Descripción	archivo	Respuesta del sistema	Flujo central
EC1.1 Opción Agregar archivos	El usuario selecciona la opción Agregar archivos		El sistema permite seleccionar el archivo. Además, permite seleccionar las siguientes opciones: <ul style="list-style-type: none"> • Cancelar • Aceptar 	Principal/FTP/ Mostrar/Agreg ar archivos
EC1.2 El usuario selecciona el archivo	El usuario especifica el archivo y ejecuta la opción aceptar.	v	El sistema valida los datos introducidos y almacena el archivo en el servidor	Principal/FTP/ Mostrar/Agreg ar archivos
EC1.3 Opción Cancelar	El usuario selecciona la opción Cancelar		El sistema no almacena el archivo en el servidor	Principal/FTP/ Mostrar/Agreg ar archivos/Canc elar
EC1.4	El usuario selecciona		El sistema almacena el	Principal/FTP/

Anexos

Opción Aceptar	la opción Aceptar		archivo en el servidor	Mostrar/Agregar archivos/Aceptar
EC1.5 Datos incorrectos	Existen datos incorrectos.		El sistema muestra un mensaje de información.	Principal/FTP/ Mostrar/Agregar archivos

Descripción de variables.

No	Nombre de campo	Clasificación	Valor nulo	Descripción
1	Archivo	File	No	Campo de texto que representa el archivo a transferir al servidor. Ejemplo 1: archivo.pdf

Modificar archivos en el servidor FTP.

Descripción general: permitir modificar directorios en el servidor FTP.				
Condiciones de ejecución: para modificar archivos en el servidor FTP debe poseer permisos sobre el servidor medios, avanzados o administrativos.				
Nombre de la sección: SC1 Modificar archivos en el servidor FTP				
Escenarios	Descripción	Nombre del archivos	Respuesta del sistema	Flujo central
EC1.1 Opción Modificar archivos	El usuario selecciona la opción modificar archivos		El sistema permite introducir el nombre del archivo. Además, permite seleccionar las siguientes opciones: <ul style="list-style-type: none"> • Cancelar • Aceptar 	Principal/FTP/ Mostrar/Modificar archivos
EC1.2 El usuario inserta los datos	El usuario especifica el nombre del archivo y ejecuta la opción aceptar.	v	El sistema valida los datos introducidos y Modifica el archivo.	Principal/FTP/ Mostrar/ Modificar archivos
EC1.3 Opción Cancelar	El usuario selecciona la opción Cancelar		El sistema no Modifica el archivo.	Principal/FTP/ Mostrar/ Modificar

Anexos

				archivos/Cancelar
EC1.4 Opción Aceptar	El usuario selecciona la opción Aceptar		El sistema Modifica el archivo.	Principal/FTP/ Mostrar/ Modificar archivos/Aceptar
EC1.5 Datos incorrectos	Existen datos incorrectos.		El sistema muestra un mensaje de información.	Principal/FTP/ Mostrar/ Modificar archivo

Descripción de variables.

No	Nombre de campo	Clasificación	Valor nulo	Descripción
1	Nuevo nombre	Campo de texto	No	Campo de texto que representa el nombre nuevo del archivo. Ejemplo 1: newname

Eliminar archivos en el servidor FTP.

Descripción general: permitir eliminar archivos en el servidor FTP.				
Condiciones de ejecución: para eliminar archivos en el servidor FTP debe poseer permisos sobre el servidor avanzados o administrativos.				
Nombre de la sección: SC1 Eliminar directorios en el servidor FTP				
Escenarios	Descripción		Respuesta del sistema	Flujo central
EC1.1 Opción Eliminar archivos	El usuario selecciona la opción Eliminar archivos		El sistema muestra un mensaje de alerta. Además, permite seleccionar las siguientes opciones: <ul style="list-style-type: none"> • Cancelar • Aceptar 	Principal/FTP/ Mostrar/ Eliminar archivos
EC1.2 Opción	El usuario selecciona la opción Cancelar		El sistema no Elimina el archivo.	Principal/FTP/ Mostrar/

Anexos

Cancelar				Eliminar archivos/Cancelar
EC1.3 Opción Aceptar	El usuario selecciona la opción Aceptar		El sistema Elimina el archivo.	Principal/FTP/Mostrar/ Eliminar archivos/Aceptar

Descargar archivos del servidor FTP.

Descripción general: permitir descargar archivos del servidor FTP.				
Condiciones de ejecución: para descargar archivos del servidor FTP debe poseer permisos sobre el servidor basicos, medios, avanzados o administrativos.				
Nombre de la sección: SC1 Subir archivos al servidor FTP				
Escenarios	Descripción		Respuesta del sistema	Flujo central
EC1.1 Opción descargar archivos	El usuario selecciona la opción descargar archivos		El sistema permite descargar el archivo seleccionado.	Principal/FTP/ Mostrar/ Descargar archivos
EC1.2 Opción Cancelar	El usuario selecciona la opción Cancelar		El sistema no almacena el archivo en el servidor	Principal/FTP/ Mostrar/Desc argar archivos/Can celar
EC1.3 Opción Aceptar	El usuario selecciona la opción Aceptar		El sistema almacena el archivo en el servidor	Principal/FTP/ Mostrar/Agre gar archivos/Acep tar

Asignar rol_FTP

Descripción general: permitir Asignar roles_FTP					
Condiciones de ejecución: para Asignar roles_FTP debe poseer permisos administrativos sobre el servidor.					
Nombre de la sección: SC1 Asignar roles_FTP					
Escenarios	Descripción	Usuarios	Rol_FTP	Respuesta del sistema	Flujo

Anexos

					central
EC1.1 Opción Editar	El usuario selecciona la opción Editar			El sistema permite Editar los roles en la conexión seleccionada.	Principal/FTP/Editar
EC1.2 El usuario inserta los datos	El usuario especifica los usuarios que obtendrán un rol especificado	V	V	El sistema permite asignarle a los usuarios introducidos el rol especificado.	Principal/FTP/Editar
EC1.3 Opción Cancelar	El usuario selecciona la opción Cancelar			El sistema no almacena el archivo en el servidor	Principal/FTP/Mostrar/Descargar archivos/Cancelar
EC1.4 Opción Aceptar	El usuario selecciona la opción Aceptar			El sistema almacena el archivo en el servidor	Principal/FTP/Mostrar/Agregar archivos/Aceptar
EC1.5 Campos en blancos	Existen campos en blancos	I V	V I	El sistema muestra un mensaje de información.	Principal/FTP/Editar

Descripción de variables

No	Nombre de campo	Clasificación	Valor nulo	Descripción
1	Usuarios	Selección	No	Campo de texto que representa los usuarios del sistema que se le cambiará el rol_FTP Ejemplo 1: Admin
2	Rol_FTP	Selección	No	Campo de texto que representa el rol_FTP a ser asignado a los usuarios seleccionados

Anexos

				Ejemplo : avanzados
--	--	--	--	----------------------------

Mostrar permisos para una conexión.

Descripción general: permitir mostrar permisos para una conexión.					
Condiciones de ejecución: para mostrar permisos para una conexión debe estar autenticado en el sistema					
Nombre de la sección: SC1 Mostrar permisos para una conexión					
Escenarios	Descripción	Usuarios	Rol_FTP	Respuesta del sistema	Flujo central
EC1.1 Opción Mostrar permisos	El usuario selecciona la opción Mostrar permisos			El sistema muestra los permisos para la conexión actual del usuario autenticado	Principal/ FTP/Mos trar/Mostr ar permisos

Mostrar información de los recursos en el servidor FTP

Descripción general: permitir mostrar información de los recursos en el servidor FTP.					
Condiciones de ejecución: para mostrar información de los recursos en el servidor FTP debe estar autenticado en el sistema.					
Nombre de la sección: SC1 Mostrar información de los recursos en el servidor FTP.					
Escenarios	Descripción	Usuarios	Rol_FTP	Respuesta del sistema	Flujo central
EC1.1 Opción Mostrar	El usuario selecciona la opción Mostrar			El sistema muestra información referente al tipo de recurso, el tamaño y la fecha de la última modificación.	Principal/ FTP/Mos trar