**TABLA DE CONTENIDO**

**Pág.**

**INTRODUCCIÓN 6**

# PRESENTACIÓN GENERAL DEL ANTEPROYECTO 7

1. **TÍTULO 7**
2. **PLANTEAMIENTO DEL PROBLEMA 7**
3. **JUSTIFICACIÓN 7**
4. **OBJETIVOS 8**
5. Objetivó General 8
6. Objetivos Específicos 8
7. ALCANCES Y DELIMINTACIONES 8
8. Alcance 8
9. Limitación y Delimitación 9
10. Delimitación Espacial 9
11. **Delimitación Temporal 9**

**Pág.**

1. **Limitación 9**

# MARCO DE REFERENCIA 10

1. ANTECEDENTES 10
2. MARCO TEÓRICO 11
3. Ingeniería de Software 11
4. Protocolo de transferencia de archivos (FTP) 11
5. CPanel 12
6. Servidor 13
7. MARCO CONCEPTUAL 13
8. SandBox 13
9. PHP (Pre – procesador de Hipertexto). 14
10. **JSP – JavaServer Pages 15**
11. **PostgreSQL 16**
12. **Mysql 17**
13. **Tomcat 17**
14. MARCO LEGAL 18

# DISEÑO METODOLOGICO 19

Pág.

1. **TIPO DE INVESTIGACIÓN 19**
2. **FUENTES DE INFORMACIÓN 19**
3. **TÉCNICAS Y PROCEDIMIENTO PARA LA RECOLECCIÓN DE INFORMACIÓN 20**
4. **Técnicas 20**
5. **Método 20**
6. **Procedimiento 20**
7. **CRONOGRAMA 21**
8. **PRESUPUESTO 22**
9. **RESULTADOS ESPERADOS 24**

**BIBLIOGRAFIA 25**

**ANEXOS 27**

**LISTA DE TABLAS**

**Pág.**

**Tabla 1.** Cronograma de actividades. 21

**Tabla 2.** Presupuesto global del trabajo de grado 22

**Tabla 3.** Descripción de los gastos de personal 22

**Tabla 4.** Descripción de materiales y suministros 22

**Tabla 5.** Descripción de equipos 23

**LISTA DE ANEXOS**

**Pág.**

**ANEXO A. 28**

**ANEXO B. 32**

**INTRODUCCIÓN**

Para la presentación de proyectos para el Departamento de Ingeniería de Sistemas (DISIS) de la UFPS relacionados con el desarrollo de software podría decirse que era común encontrarse con los siguientes escenarios:

1. El estudiante desarrollaba su proyecto en su computador personal. La ventaja de este escenario para el estudiante es la fácil administración y configuración de su entorno de trabajo. Sin embargo, en caso de presentarse alguna falla en el equipo personal se corre el riesgo de perder la información del proyecto; adicionalmente, si el proyecto requiere del despliegue en servidores de la Universidad, generalmente se presentan grandes inconvenientes para la configuración del mismo en el nuevo ambiente.
2. El estudiante utiliza los servicios de préstamo de computadores suministrados por la DISIS. Este escenario es adecuado para estudiantes que con los recursos para adquirir un computador personal adecuado para la elaboración del proyecto. Sin embargo, el préstamo de los recursos para los estudiantes es limitado y generalmente en los proyectos de grado de Ingeniería de Sistemas se requiere dedicar un tiempo considerable para la elaboración del mismo.

Teniendo en cuenta lo anterior el departamento de ingeniería de sistemas promovió un proyecto que tenía como idea facilitar la construcción de software por parte de los estudiantes, mediante con la creación de una herramienta intuitiva denominada Sandbox.

El Sandbox es un sistema que integra un conjunto de aplicaciones y políticas orientadas en el desarrollador de aplicaciones, facilitando la administran de los proyectos, este método se comenzó a implementar en el segundo semestre del 2011 en la carrera de Ingeniería de Sistemas de la UFPS; de este modo los estudiantes obtienen un ambiente confiable para realizar las pruebas de sus estos.

En la actualidad la lista de tecnologías utilizadas para el desarrollo es muy intensa pero dada la necesidad de prestar el mejor servicio se integrar tecnologías para el desarrollo tales como: PHP y JSP, del mismo modo para el almacenamiento de información se utilizaran administradores de base de datos relacionales como Postgresql y mysql.

Al comienzo de la implementación de este sistemas se generaron una serie de inconvenientes que al transcurrir del tiempo se ha tenido la necesidad de optimizar. Como alternativa de esta optimización se lleva a la construcción de un frontend (interfaz) que realiza la visualización de los elementos de los usuarios en el navegador y backend (motor) que es el administrador del sitio con sus respectivos sistemas, este conjunto de conceptos garantiza una mejora del sistema. No obstante a este conjunto de implementaciones el sistema no deja de presentar un número de inconvenientes y necesidades tales como;

* la creación de cuentas de usuario dinámicamente.
* Problemas de usabilidad y escalabilidad en el administrador FTP y War.

A continuación se presenta una solución con justificación, alcances, delimitaciones y cronograma para mejorar el sistema Sandbox, de este modo la integración de nuevas herramientas tales como Python, .Net y MongolDB.

.

# PRESENTACIÓN GENERAL DEL ANTEPROYECTO

1. **TÍTULO**

Construcción de un frontend y backend para gestión de aplicaciones web en la nube para el programa de Ingeniería de Sistemas de la Universidad Francisco de Paula de Santander.

1. **PLANTEAMIENTO DEL PROBLEMA**

Hace unos años en las materias de ingeniería de software se llevaba a cabo la sustentación de proyectos en unos servidores en el centro de cómputo de la Universidad, siendo estos solamente asequibles desde estas instalaciones, creando inconvenientes a la hora de desplegarlos para la sustentación, en algunos momentos se creaban errores de versiones de las tecnologías utilizadas, así mismo como problemas de acceso a las base de datos, ya que se contaba un número limitado de equipos y de monitores en esta área para la colaboración en este proceso.

Esto hizo que a la hora de sustentar los estudiantes llevaran horas adaptando sus proyectos a los servidores asignados para dicha tarea y en ocasiones haciendo que estas actividades fuera una pérdida de tiempo para el estudiante y docentes.

Ampliar los problemas por los cuales se hace necesario el desarrollo de este proyecto

Problema de investigación

¿Por qué se debe Desarrollar un sistema de gestión de aplicaciones web, dispuestos en los servidores Sandbox de la U.F.P.S.? Cambiar la pregunta relacionarla con el objetivo general

Preguntas de investigación (Se relacionan con los objetivos específicos)

1. **JUSTIFICACIÓN**

Los proyectos de semestre son una parte vital para los estudiantes de ingeniería de sistemas, porque es en esta práctica en la que realmente se aprende a conceptualizar lo teórico con lo práctico, haciendo de esta manera más completa y competitiva la Ingeniería de Sistemas de la U.F.P.S.

El concepto de Sandbox nace como una alternativa en la que los estudiantes pudiera desplegar sus aplicaciones y administrarlas sin tener que depender de turnos en salas y administradores de servidores, es de esta manera que surge la idea de crear una plataforma para la administración de estas tareas en un entorno web el cual esté disponible todo el semestre, haciendo posible que el estudiante realice un desarrollo de sus proyectos utilizándola e ir solucionando los inconvenientes que se vayan presentando en el desarrollo.

No obstante aunque se realizó este cambio trascendental en la forma de presentar los proyectos nació un nuevo inconveniente enfatizado en la confianza de esos servicio a la hora de utilizarlo, tales como problemas a la hora de desplegar las aplicaciones, la falta de integración en las cuentas de usuario para el uso de los diferentes servicios y poco soporte ocasionando que la plataforma no cumpliera con el. Esto forzando a que los estudiantes presentaran sus proyectos nuevamente en servidores locales.

El por qué? Para qué? Quienes se benefician? Que impacto? La viabilidad y los resultados que se quieren tener?

1. **OBJETIVOS**
2. Objetivó General

Desarrollar un panel de control genérico de despliegue de proyectos de software en entorno web, para facilitar la sustentación de los trabajos de los estudiantes del plan de estudio de ingeniería de sistemas de la UFPS mediante el uso de una GUI amigable para el usuario.

1. Objetivos Específicos

* Diagnosticar las arquitecturas de software existente en los servidores y procesos de administración de la plataforma Sandbox.
* Integrar las nuevas tecnologías y servicios tales como .Net, Python y MongoDB a la plataforma Sandbox.
* Elaborar un Backend y Frontend utilizando JSP, Html5, CSS3 y Jquery que permita la integración de las nuevas herramientas con las ya existentes utilizando la metodología SCRUM.
* Elaborar una estrategia de prueba ágil de software para garantizar el correcto funcionamiento de las aplicaciones administrativas en la plataforma Sandbox.

1. ALCANCES Y DELIMINTACIONES
2. Alcance

El alcance para la construcción de la herramienta web de administración de los servicios del sistema Sandbox está dado por las siguientes actividades:

* Construcción de una herramienta Web que administre (ampliar) los siguientes servicios: PHP, HTML, python, FTP, Aplicaciones WAR, .NET y github.
* Creación de los protocolos de seguridad para la administración de las bases de datos en Postgresql, Mysql y MongolDB.
* Implantación de sistemas de login y registro de usuarios.
* La realización de manuales para usuarios y administradores del sistema.
* Implementación de procedimientos de instalación y actualización del sistema.

1. Limitación y Delimitación
2. Delimitación Espacial

El proyecto se desarrollará para el Plan de Estudio de Ingeniería de Sistemas de la Universidad Francisco de Paula Santander, ubicado en el 4 piso del edificio de Aula Sur en la Avenida Gran Colombia No. 12E – 96B Barrio Colsag.

1. **Delimitación Temporal**

El desarrollo del proyecto se dará en un tiempo de 8 meses que serán distribuidos en las diferentes etapas del proyecto.

1. **Limitación**

* La seguridad y robustez de los entornos Linux donde se implementa el sistema, impiden en ocasiones la correcta utilización de los servicios por parte de los usuarios.
* La falta de documentación existente para el desarrollo de sistemas Sandbox, esto impide conseguir información segura para implementar fragmentos de código en el servidor que ejecute procesos del sistema para la solución de actividades del proyectos.
* Restricciones propias de los servicios de Microsoft para el despliegue de aplicaciones .Net en sus servidores, podría ocacionar alta dependencia del sistema a configuraciones del servicio de host.

# MARCO DE REFERENCIA

1. ANTECEDENTES

* Sandboxie. es un software de tipo SandBox para Microsoft Windows, parecido a un sistema de virtualización por aislamiento. Este freeware crea un contenedor dentro del cual los programas de aplicación se ejecutan de forma segura. Sandboxie Holdings LLC – 2004. Referenciar todo
* Google Sandbox. Es un efecto observado, que postula un sistema de filtrado especial de Google en su algoritmo, que hace que las páginas web de nueva creación no aparezcan, o aparezcan en posiciones muy inferiores, en los resultados de búsqueda, aún estando muy especializadas para ciertas palabras clave, o teniendo muchos enlaces entrantes. La explicación teórica supone que Google predice que no es normal que un sitio de reciente creación, tenga muchos enlaces entrantes en sus primeras semanas de vida, por lo que debe ser automáticamente retirado de los primeros puestos de las búsquedas. De esta manera, Google parece establecer un periodo de pruebas para comprobar si el sitio web puede o no, conseguir más enlaces y relevancia, de manera normal. Google – 2004.

### Universidad de los Andes, Laboratorio de Ingeniería de Sistemas y Computación, la cual cuenta con un sistema de servidores para la administración de proyectos informáticos orientado a estudiantes con tesis o proyectos informáticos.

* Caracterización de servidores Web de ámbito académico, realizado por: Vicente José Castaño Díaz, cuyo objetivo tuvo “Comprobar si en la actualidad se cumplen los invariantes establecidos con anterioridad por otros autores en la caracterización de servidores Web de ámbito académico.” y “Comprobar si es posible realizar un estudio de estas características haciendo uso de las herramientas software disponibles en la actualidad.” Cuya conclusión fue: “Los datos de otros invariantes reflejan diferencias significativas con respecto a estudios anteriores, algunos de ellos recientes. Habría que hacer una revisión de los invariantes para poder caracterizar de manera efectiva las nuevas tendencias en Internet, como por ejemplo portales de tipo YouTube o aplicaciones Second Life”.

1. MARCO TEÓRICO
2. Ingeniería de Software

Software es el conjunto de instrucciones que permite al hardware de la computadora desempeñar trabajo útil. En las últimas décadas del siglo XX, las reducciones de costo en hardware llevaron a que el software fuera un componente ubicuo de los dispositivos usados por las sociedades industrializadas.

La ingeniería de software se encarga de crear y mantener las aplicaciones de software aplicando tecnologías y prácticas de las ciencias computacionales y la ingeniería.

Ingeniería de software, se puede definir como el conjunto de métodos, técnicas y herramientas que controlan el proceso integral del desarrollo de software y suministra las bases para construir software de calidad de forma eficiente en los plazos adecuados. Referenciar

Metodologías ágiles

1. SCRUM

Es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto. Los roles principales en Scrum son el ScrumMaster, que mantiene los procesos y trabaja de forma similar al director de proyecto, el ProductOwner, que representa a los stakeholders (interesados externos o internos), y el Team que incluye a los desarrolladores.

Durante cada sprint, un periodo entre una y cuatro semanas (la magnitud es definida por el equipo), el equipo crea un incremento de software potencialmente entregable (utilizable). El conjunto de características que forma parte de cada sprint viene del Product Backlog, que es un conjunto de requisitos de alto nivel priorizados que definen el trabajo a realizar. Los elementos del Product Backlog que forman parte del sprint se determinan durante la reunión de Sprint Planning. Durante esta reunión, el Product Owner identifica los elementos del Product Backlog que quiere ver completados y los hace del conocimiento del equipo. Entonces, el equipo determina la cantidad de ese trabajo que puede comprometerse a completar durante el siguiente sprint.1 Durante el sprint, nadie puede cambiar el Sprint Backlog, lo que significa que los requisitos están congelados durante el sprint.

Scrum permite la creación de equipos autoorganizados impulsando la co-localización de todos los miembros del equipo, y la comunicación verbal entre todos los miembros y disciplinas involucrados en el proyecto.

Un principio clave de Scrum es el reconocimiento de que durante un proyecto los clientes pueden cambiar de idea sobre lo que quieren y necesitan (a menudo llamado requirements churn), y que los desafíos impredecibles no pueden ser fácilmente enfrentados de una forma predictiva y planificada. Por lo tanto, Scrum adopta una aproximación pragmática, aceptando que el problema no puede ser completamente entendido o definido, y centrándose en maximizar la capacidad del equipo de entregar rápidamente y responder a requisitos emergentes.

1. Control de Versiones

Se llama control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo. Una versión, revisión o edición de un producto, es el estado en el que se encuentra dicho producto en un momento dado de su desarrollo o modificación. Aunque un sistema de control de versiones puede realizarse de forma manual, es muy aconsejable disponer de herramientas que faciliten esta gestión dando lugar a los llamados sistemas de control de versiones o SVC (del inglés System Version Control).

1. Servidor

En informática, un servidor es un nodo que forma parte de una red, provee servicios a otros nodos denominados clientes. También se suele denominar con la palabra servidor a:

Una aplicación informática o programa que realiza algunas tareas en beneficio de otras aplicaciones llamadas clientes. Algunos servicios habituales son los servicios de archivos, que permiten a los usuarios almacenar y acceder a los archivos de una computadora y los servicios de aplicaciones, que realizan tareas en beneficio directo del usuario final. Este es el significado original del término. Es posible que un ordenador cumpla simultáneamente las funciones de cliente y de servidor.

Una computadora en la que se ejecuta un programa que realiza alguna tarea en beneficio de otras aplicaciones llamadas clientes, tanto si se trata de un ordenador central (mainframe), un miniordenador, una computadora personal, una PDA o un sistema embebido; sin embargo, hay computadoras destinadas únicamente a proveer los servicios de estos programas: estos son los servidores por antonomasia.

Un servidor no es necesariamente una máquina de última generación de grandes proporciones, no es necesariamente un superordenador; un servidor puede ser desde una computadora vieja, hasta una máquina sumamente potente (ej.: servidores web, bases de datos grandes, etc. Procesadores especiales y hasta varios terabytes de memoria). Todo esto depende del uso que se le dé al servidor. Si usted lo desea, puede convertir al equipo desde el cual usted está leyendo esto en un servidor instalando un programa que trabaje por la red y a la que los usuarios de su red ingresen a través de un programa de servidor web como Apache.

1. Protocolo de transferencia de archivos (FTP)

El servicio FTP es ofrecido por la capa de aplicación del modelo de capas de red TCP/IP al usuario, utilizando normalmente el puerto de red 20 y el 21. Un problema básico de FTP es que está pensado para ofrecer la máxima velocidad en la conexión, pero no la máxima seguridad, ya que todo el intercambio de información, desde el login y password del usuario en el servidor hasta la transferencia de cualquier archivo, se realiza en texto plano sin ningún tipo de cifrado, con lo que un posible atacante puede capturar este tráfico, acceder al servidor y/o apropiarse de los archivos transferidos.

Para solucionar este problema son de gran utilidad aplicaciones como scp y sftp, incluidas en el paquete SSH, que permiten transferir archivos pero cifrando todo el tráfico.

Las aplicaciones más comunes de los servidores FTP suelen ser el alojamiento web, en el que sus clientes utilizan el servicio para subir sus páginas web y sus archivos correspondientes; o como servidor de backup (copia de seguridad) de los archivos importantes que pueda tener una empresa. Para ello, existen protocolos de comunicación FTP para que los datos se transmitan cifrados, como el SFTP.

Algunos clientes de FTP básicos en modo consola vienen integrados en los sistemas operativos, incluyendo Microsoft Windows, DOS, GNU/Linux y Unix. Sin embargo, hay disponibles clientes con opciones añadidas e interfaz gráfica. Aunque muchos navegadores tienen ya integrado FTP, es más confiable a la hora de conectarse con servidores FTP no anónimos utilizar un programa cliente.

Un «cliente FTP basado en Web» no es más que un cliente FTP al cual podemos acceder a través de nuestro navegador web sin necesidad de tener otra aplicación para ello. El usuario accede a un servidor web (HTTP) que lista los contenidos de un servidor FTP. El usuario se conecta mediante HTTP a un servidor web, y el servidor web se conecta mediante 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.

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 y los listados, en el puerto especificado.

Lo anterior tiene un grave problema de seguridad, y es que la máquina cliente debe estar dispuesta a aceptar cualquier conexión de entrada en un puerto superior al 1024, con los problemas que ello implica si tenemos el equipo conectado a una red insegura como Internet. 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.

Cuando el cliente envía un comando PASV sobre el canal de control, el servidor FTP le indica por el canal de control, el puerto (mayor a 1023 del servidor. Ejemplo: 2040) al que debe conectarse el cliente. El cliente inicia una conexión desde el puerto siguiente al puerto de control (Ejemplo: 1036) hacia el puerto del servidor especificado anteriormente (Ejemplo: 2040).

1. CPanel

Es una herramienta de administración basada en tecnologías web para administrar sitios de manera fácil, con una interfaz limpia. Se trata de un software no libre disponible para un gran número de distribuciones de Linux que soporten RPM, como SuSE, Fedora, Mandriva, etc. El soporte preliminar es para Debian, se debió solicitar que saliera del estado beta por varios años, y para el cual no posee soporte. A cPanel se accede por los puertos 2082 y 2083 (para versiones de SSL). La autenticación es por HTTP o una página de ingreso.

Se diseñó para el uso comercial de servicios de alojamiento web, es por esto que la compañía no lo ofrece con licencia de uso personal. Sin embargo, los dueños de organizaciones sin fines de lucro, como instituciones educacionales y caridad pueden solicitar una licencia sin costo.

1. Base de Datos Relacional

Una Base de Datos Relacional, es una base de datos que cumple con el modelo relacional, el cual es el modelo más utilizado en la actualidad para implementar bases de datos ya planificadas. Permiten establecer interconexiones (relaciones) entre los datos (que están guardados en tablas), y a través de dichas conexiones relacionar los datos de ambas tablas, de ahí proviene su nombre: "Modelo Relacional".

Elementos y características:

* **Relaciones Base y Derivadas:** en una base de datos relacional, todos los datos se almacenan y se accede a ellos por medio de relaciones. Las relaciones que almacenan datos son llamadas "relaciones base" y su implementación es llamada "tabla". Otras relaciones no almacenan datos, pero son calculadas al aplicar operaciones relacionales. Estas relaciones son llamadas "relaciones derivadas" y su implementación es llamada "vista" o "consulta". Las relaciones derivadas son convenientes ya que expresan información de varias relaciones actuando como si fuera una sola.
* **Restricciones**: una restricción es una limitación que obliga el cumplimiento de ciertas condiciones en la base de datos. Algunas no son determinadas por los usuarios, sino que son inherentemente definidas por el simple hecho de que la base de datos sea relacional. Algunas otras restricciones las puede definir el usuario, por ejemplo, usar un campo con valores enteros entre 1 y 10.

Las restricciones proveen un método de implementar reglas en la base de datos. Las restricciones limitan los datos que pueden ser almacenados en las tablas. Usualmente se definen usando expresiones que dan como resultado un valor booleano, indicando si los datos satisfacen la restricción o no.

Las restricciones no son parte formal del modelo relacional, pero son incluidas porque juegan el rol de organizar mejor los datos. Las restricciones son muy discutidas junto con los conceptos relacionales.

* **Dominios:** un dominio describe un conjunto de posibles valores para cierto atributo. Como un dominio restringe los valores del atributo, puede ser considerado como una restricción. Matemáticamente, atribuir un dominio a un atributo significa "todos los valores de este atributo deben ser elementos del conjunto especificado".

Distintos tipos de dominios son: enteros, cadenas de texto, fecha, no procedurales, etc.

Cada tabla puede tener uno o más campos cuyos valores identifican de forma única cada registro de dicha tabla, es decir, no pueden existir dos o más registros diferentes cuyos valores en dichos campos sean idénticos. Este conjunto de campos se llama clave única. Pueden existir varias claves únicas en una determinada tabla, y a cada una de éstas suele llamársele candidata a clave primaria.

* **Clave primaria:** una clave primaria es una clave única elegida entre todas las candidatas que define unívocamente a todos los demás atributos de la tabla, para especificar los datos que serán relacionados con las demás tablas. La forma de hacer esto es por medio de claves foráneas.
* **Clave foránea:** una clave foránea es una referencia a una clave en otra tabla, determina la relación existente en dos tablas. Las claves foráneas no necesitan ser claves únicas en la tabla donde están y sí a donde están referenciadas.
* **Clave índice**: Las claves índices surgen con la necesidad de tener un acceso más rápido a los datos. Los índices pueden ser creados con cualquier combinación de campos de una tabla. Las consultas que filtran registros por medio de estos campos, pueden encontrar los registros de forma no secuencial usando la clave índice.

Las bases de datos relacionales incluyen múltiples técnicas de ordenamiento, cada una de ellas es óptima para cierta distribución de datos y tamaño de la relación.

Los índices generalmente no se consideran parte de la base de datos, pues son un detalle agregado. Sin embargo, las claves índices son desarrolladas por el mismo grupo de programadores que las otras partes de la base de datos.

* **Procedimientos almacenados:** un procedimiento almacenado es código ejecutable que se asocia y se almacena con la base de datos. Los procedimientos almacenados usualmente recogen y personalizan operaciones comunes, como insertar un registro dentro de una tabla, recopilar información estadística, o encapsular cálculos complejos. Son frecuentemente usados por un API por seguridad o simplicidad.

Los procedimientos almacenados no son parte del modelo relacional, pero todas las implementaciones comerciales los incluyen.

1. Base de Datos No relacionales (NoSQL)

Típicamente las bases de datos relacionales modernas han mostrado poca eficiencia en determinadas aplicaciones que usan los datos de forma intensiva, incluyendo el indexado de un gran número de documentos, la presentación de páginas en sitios que tienen gran tráfico, y en sitios de streaming audiovisual. Las implementaciones típicas de RDBMS se han afinado o bien para una cantidad pequeña pero frecuente de lecturas y escrituras o para un gran conjunto de transacciones que tiene pocos accesos de escritura. Por otro lado NoSQL puede servir gran cantidad de carga de lecturas y escrituras.

Implementaciones de NoSQL usadas en el mundo real incluyen los 3TB de los marcadores verdes de Digg (indicados para señalar las historias votadas por otros en la red social; aunque duró menos de 3 meses y fue abandonado). Los 6 TB de la base de datos del “ENSEMBLE” de la Comisión Europea usado en los modelos de comparación y calidad del aire, y en los 50 TB de la búsqueda de la bandeja de entrada de Facebook.

Las arquitecturas NoSQL frecuentemente aportan escasas garantías de consistencia, tales como consistencia de eventos o transaccional restringida a ítems únicos de datos. Algunos sistemas, sin embargo, aportan todas las garantías de los sistemas ACID en algunas instancias añadiendo una capa intermedia (como por ejemplo, AppScale o CloudTPS). Hay dos sistemas que han sido desplegados y que aportan aislamiento snapshot para almacenamientos de columna: El sistema Percolator de Google (basado en el sistema BigTable) y el sistema transaccional de Hbase desarrollado por la universidad de Waterloo. Estos sistemas, desarrollados de forma independiente, usan conceptos similares para conseguir transacciones ACID distribuidas de múltiples filas con garantías de aislamiento snapshot para el sistema subyacente de almacenamiento en esa columna, sin sobrecarga extra en la gestión de los datos, despliegue en el sistema de middleware, ni mantenimiento introducido por la capa de middleware.

Bastantes sistemas NoSQL emplean una arquitectura distribuida, manteniendo los datos de forma redundante en varios servidores, usando frecuentemente una tabla hash distribuida. De esta forma, el sistema puede realmente escalar añadiendo más servidores, y el fallo en un servidor puede ser tolerado.

Algunos defensores de NoSQL promueven interfaces simples tales como los arrays asociativos o los pares clave-valor. Otros sistemas, tales como las bases de datos nativas en XML, promueven el soporte del estándar Xquery. Los sistemas más novedosos tales como CloudTPS también soportan unión de queries.

1. MARCO CONCEPTUAL
2. SandBox

El sandbox protege "en tiempo real" los servidores de datos, y hace de control preventivo de la ejecución de código fuente, datos y/o contenido, evitando unos cambios que podrían ser perjudiciales (independientemente de la intención del autor de los mismos) para un sistema, o que simplemente, podrían ser cambios de difícil reversión.

1. PHP (Pre – procesador de Hipertexto).

Es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante.

1. **Python**

Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma.

1. **.NET**

La plataforma .NET de Microsoft es un componente de software que puede ser añadido al sistema operativo Windows. Provee un extenso conjunto de soluciones predefinidas para necesidades generales de la programación de aplicaciones, y administra la ejecución de los programas escritos específicamente con la plataforma.

1. **JSP – JavaServer Pages**

Las JSPs son en realidad una forma alternativa de crear servlets ya que el código JSP se traduce a código de servlet Java la primera vez que se le invoca y en adelante es el código del nuevo servlet el que se ejecuta produciendo como salida el código HTML que compone la página web de respuesta.

1. **PostgreSQL**

Es un sistema de gestión de base de datos relacional orientado a objetos y libre.

1. **Mysql**

Es un sistema de gestión de bases de datos relacionales, multihilo y multiusuario

1. **MongoDB**

Es un sistema de base de datos NoSQL orientado a documentos, desarrollado bajo el concepto de código abierto. MongoDB forma parte de la nueva familia de sistemas de base de datos NoSQL. En vez de guardar los datos en tablas como se hace en las base de datos relacionales, MongoDB guarda estructuras de datos en documentos tipo JSON con un esquema dinámico (MongoDB llama ese formato BSON), haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida.

1. **Git**

Es un software de control de versiones, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente.

1. **Tomcat**

Tomcat es un servidor web con soporte de servlets y JSPs. Tomcat no es un servidor de aplicaciones, como JBoss o JOnAS. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache.

1. MARCO LEGAL

* El software utilizado para este proyecto se encuentra enmarcado bajo los términos de la “GNU General Public License” publicada por la “Free Software Foundation”.

Esta licencia se denomina GNU Copyleft (liberación de derechos de GNU, un juego de palabras con el termino ingles Copyright).

Cubre todos los programas producidos por GNU un acrónimo compuesto también de las palabras: GNU no es Unix y por la Free Software Foundation (Fundación de Software Gratuito). Esta licencia permite a los desarrolladores crear programas para el público en general.

* Ya que el sistema es un prestador de servicios todos sus usuario estarán regidos por la ley 1273 de 2009 estando está relacionada con los delitos informáticos y la protección de la información y de los datos, la cual castiga con penas de prisión de hasta 120 meses y multas de hasta 1500 salarios mínimos legales mensuales vigentes.
* Tomando en cuenta que el sistema está orientado principalmente a estudiantes de la UFPS, en el caso de conductas que desmeriten el nombre de la institución estas será castigadas por la institución bajo las normas y políticas amparadas por el estatuto estudiantil de dicha institución.

# DISEÑO METODOLOGICO

1. **TIPO DE INVESTIGACIÓN**

Nuestra investigación se basa en encontrar herramientas o módulos que nos sirvan para desarrollar aplicaciones que estén orientadas a la administración del servidor Sandboxya que este como se ha explicado anteriormente había presentado una serie de dificultades a la hora de ejecutar su plataforma.

Este motivo es el que hace que nuestra investigación se desarrolle en un tipo de investigación aplicada ya que nuestro resultado dará una posible solución a esta problemática; así mismo se puede observar que la parte de análisis de esta misma se llevara a cabo bajo una filosofía investigativa tipo documental, esto hace que nuestra investigación sea de tipo documental y aplicada.

1. **FUENTES DE INFORMACIÓN**

* **Fuentes de información primario:** esta consiste en la recopilación de la experiencia de los administradores del sistema y así mismo utilizarla para evitar posibles inconvenientes pasados.

También tenemos que recopilar información de los usuarios que han ejecutado este servicio, como los estudiantes del Plan de Estudio de Ingeniería de Sistemas de la UFPS.

* **Fuentes de información secundaría:** para este ítem nos centramos en la utilización de la documentación ya existente para las aplicaciones a utilizar y en otras que se vayan explorando durante el proceso de investigación.

1. **TÉCNICAS Y PROCEDIMIENTO PARA LA RECOLECCIÓN DE INFORMACIÓN**
2. **Técnicas**

El tipo de técnica para realizar la recolección de los datos es de Observación ordinaria y/o participante, ya que nuestro medio de investigación es mucho más cualitativa participativa que cuantitativa.

1. **Método**

En nuestra investigación nos centramos en realizar una observación de los procesos que se ejecutan en el servidor para poder entender mejor como enfocarlos en una aplicación web.

1. **Procedimiento**

A continuación se realizara un listado de tareas para ejecutar esta investigación:

1. Conocer las actividades diseñadas para la ejecución del servidor.
2. Familiarizarse con las aplicaciones instaladas en el servidor para su administración.
3. Identificar los posibles procesos problemáticos en la administración del servidor.
4. Integrar las nuevas tecnologías necesarias para el correcto funcionamiento del servidor.
5. Mejorar las aplicaciones ya existentes en el servidor, tanto funcionalmente como visualmente.
6. Investigar sobre los componentes de FTP y administración de War ya existentes orientados a la administración de servidores.
7. Analizar las nuevas tecnologías integradas a los servidores.
8. Basándonos en el análisis y diseño de software, crear una serie de modelos de datos y objetos que nos permitan desarrollar una herramienta web acorde a nuestras necesidades.
9. Reunir los Product Owner, colocarlos en Product Backlog para diseñar los Sprint, todo esto según el ciclo de vida SCRUM.
10. Ejecutar los Sprints necesarios.
11. Diseñar un plan de pruebas para las aplicaciones en el servidor, orientado a medir su tasa de efectividad y la identificación de posibles errores con SCRUM.
12. Documentar los procesos de instalación y corrección de errores del servidor.
13. Crear un manual de usuarios para la utilización de las aplicaciones del servidor.
14. **CRONOGRAMA**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Objetivos  específicos | actividades | Meses y semanas | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Mes 1 | | | | Mes 2 | | | | Mes 3 | | | | Mes 4 | | | | Mes 5 | | | | Mes 6 | | | | Mes 7 | | | | Mes 8 | | | |
| S1 | S2 | S3 | S4 | S1 | S2 | S3 | S4 | S1 | S2 | S3 | S4 | S1 | S2 | S3 | S4 | S1 | S2 | S3 | S4 | S1 | S2 | S3 | S4 | S1 | S2 | S3 | S4 | S1 | S2 | S3 | S4 |
| 1 | a1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| a2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| a3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | a4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| a 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| a 6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| a7 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | a8 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| a9 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| a  10 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | a 11 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| a  12 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| a  13 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Tabla 1.** Cronograma de actividades.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PRESUPUESTO GLOBAL DEL TRABAJO DE GRADO** | | | | |
| **RUBROS** | **FUNETE DE FINANCIACION** | | | **TOTAL** |
|  | **Estudiante** | **UFPS** | **Externa** |  |
| Personal |  | $ 2’080.000 |  | $ 2’080.000 |
| Materiales | $ 127.500 |  |  | $ 127.500 |
| Equipos | $ 800.000 | $3’400.000 |  | $ 4’200.000 |
| **Total** | $ 927.500 | $ 5’480.000 |  | $ 6’307.500 |

1. **PRESUPUESTO**

**Tabla 2.** Presupuesto global del trabajo de grado

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **DESCRIPCIÓN DE LOS GASTOS DE PERSONAL** | | | | | | |
| **Nombre de estudiantes** | **Función en el trabajo** | **Dedicación Hora / Semana** | **FUENTE DE FINANCIACION** | | | **TOTAL** |
| **Estudiante** | **UFPS** | **Externa** |
| Fernando José Torres Bermúdez | Administrador del servidor Sandbox | 40 Horas |  | $ 2’080.000 |  | $ 2’080.000 |
| **Total** |  |  |  | $ 2’080.000 |  | $ 2’080.000 |

**Tabla 3.** Descripción de los gastos de personal

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DESCRIPCION DE MATERIALES Y SUMINISTROS** | | | | |
| **MATERIAL** | **FUENTE DE FINANCIACION** | | | **TOTAL** |
| **Estudiante** | **UFPS** | **Externa** |
| Resma tam. carta | $ 40.0000 |  |  | $ 40.000 |
| Tinta impresora | $ 80.000 |  |  | $ 80.000 |
| Carpetas de presentación | $ 7.500 |  |  | $ 7.500 |
| **Total** | $ 127.500 |  |  | $ 127.500 |

**Tabla 4.** Descripción de materiales y suministros

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DESCRIPCION DE EQUIPOS** | | | | |
| **DESCRIPCION DE EQUIPOS** | **FUENTE DE FINANCIACION** | | | **TOTAL** |
| **Estudiante** | **UFPS** | **Externa** |
| Mac Pro |  | $ 3’400.000 |  | $ 3’400.000 |
| Inspirion N4010 | $800.000 |  |  | $ 800.000 |
| **Total** | $800.000 | $ 3’400.000 |  | $ 4’200.000 |

**Tabla 5.** Descripción de equipos

1. **RESULTADOS ESPERADOS**

Al terminar el proyecto se espera obtener un software para la administración y gestión de servidores de aplicación en Php, JSP, Python y .NET, que cumpla con las buenas prácticas en el desarrollo de software y así mismo brindar la mejor experiencia a sus usuarios tomando en cuenta los siguientes puntos:

1. Que su GUI sea funcional y amigable.
2. Disponibilidad de herramientas para la ejecución de proyectos de software.
3. Fácil administración y mantenimiento.

Estos puntos resumen el objetivo de este proyecto de grado el cual entregara una aplicación tipo CPANEL para facilitar la entrega de proyectos de software del plan de estudios de ingeniería de software.

**BIBLIOGRAFIA**

Txon - Wikipedia; Sandboxie [online]. 26 de junio de 2007. Disponible en: <URL:http://es.wikipedia.org/wiki/Sandboxie>.

Muro Bot - Wikipedia; Google Sandbox [online]. 21 de febrero de 2009. Disponible en: <URL:http://es.wikipedia.org/wiki/Google\_Sandbox>.

Rizome - Wikipedia; Entorno de pruebas (informática) [online]. 14 de abril de 2009. Disponible en: <URL:http://es.wikipedia.org/wiki/Entorno\_de\_pruebas\_(inform%C3%A1tica)>.

Tei - Wikipedia; Servidor [online]. 17 de febrero de 2002. Disponible en: <URL:http://es.wikipedia.org/wiki/Servidor>.

Superzerocool - Wikipedia; cPanel [online]. 9 de marzo de 2006. Disponible en: <URL:http://es.wikipedia.org/wiki/CPanel>.

cm05840.telecable.es - Wikipedia; File Transfer Protocol [online]. 3 de marzo de 2002. Disponible en: <URL:http://es.wikipedia.org/wiki/File\_Transfer\_Protocol>

Maveric149 - Wikipedia; PHP [online]. 3 de noviembre del 2002. Disponible en: <URL:http://es.wikipedia.org/wiki/PHP>.

JorgeGG - Wikipedia; JavaServer Pages [online]. 27 de marzo de 2004. Disponible en: <URL:http://es.wikipedia.org/wiki/JavaServer\_Pages>.

Pilaf - Wikipedia; PostgreSQL [online]. 30 de mayo 2004. Disponible en: <URL:http://es.wikipedia.org/wiki/PostgreSQL>.

SpeedyGonzalez - deltaasesores; MySQL [online]. 14 de octubre de 2003. Disponible en: <URL:http://es.wikipedia.org/wiki/MySQL>.

Isabella Gandini, Andrés Isaza, Alejandro Delgado - Wikipedia; Ley de Delitos Informáticos en Colombia [online]. 14 de octubre de 2011. Disponible en: <URL:http://www.deltaasesores.com/articulos/autores-invitados/otros/3576-ley-de-delitos-informaticos-en-colombia>.

Sergio Vares - Wikipedia; Tomcat [online]. 18 de febrero de 2005. Disponible en: <URL:http://es.wikipedia.org/wiki/Tomcat>.

Universidad de los Andes; Laboratorios de Ingeniería de Sistemas y Computación [online]. <URL:http://admonsis.uniandes.edu.co/isclabs/index.php?option=com\_content&view=article&id=60&Itemid=18.>

Castaño Díaz, Vicente José; Caracterización de servidores Web de ámbito académico; pp 6, 57.

Mantilla Duque, Leidy Johanna. Desarrollo de una herramienta web para la incorporación de recursos de educación 2.0 en la Universidad Francisco de Paula Santander. Trabajo de grado Ingeniería de Sistemas. San José de Cúcuta. Universidad Francisco de Paula Santander. Facultad de Ingenierías. 2011.134p.

Armonth – Wikipedia; Front-end y back-end [online]. 20 de noviembre de 2004. Disponible en: <URL: http://es.wikipedia.org/wiki/Front-end\_y\_back-end>.

Juan.palacio – Wikipedia; Scrum [online]. 18 de abril de 2006. Disponible en: <URL: http://es.wikipedia.org/wiki/Scrum#Caracter.C3.ADsticas\_de\_Scrum>.

Berenice Ruiz – sg.com revista. Marcando la Pauta para las Pruebas Ágiles. Febrero de 2013. Disponible en: <URL: http://sg.com.mx/revista/38/marcando-la-pauta-para-las-pruebas-%C3%A1giles#.U88WkvTuKC->.

Lluvia – Wikipedia. Python. 5 de febrero de 2008. Disponible en: <URL: http://es.wikipedia.org/wiki/Python>.

200.92.213.173 – Wikipedia. Microsoft .NET. 7 de abril de 2005. Disponible en: <URL: http://es.wikipedia.org/wiki/Microsoft\_.NET>.

Str0b0s – Wikipedia. Base de datos relacional. 14 de abril de 2005. Disponible en: <URL: http://es.wikipedia.org/wiki/Base\_de\_datos\_relacional>.

El Pantera – Wikipedia. NoSQL. 2 de marzo de 2010. Disponible en: <URL: http://es.wikipedia.org/wiki/NoSQL>.

Victormlgh – Wikipedia. MongoDB. 9 de mayo 2012. Disponible en <URL: http://es.wikipedia.org/wiki/MongoDB>.

Ascánder – Wikipedia. Control de versiones. 21 de noviembre de 2004. Disponible en: <URL: http://es.wikipedia.org/wiki/Control\_de\_versiones>.

Paradoja – Wikipedia. Git. 26 de octubre de 2007. Disponible en: <URL: http://es.wikipedia.org/wiki/Git>.

**ANEXOS**

**ANEXO A. POLÍTICAS DE USO DE SANDBOX**



**POLÍTICAS DE USO DE SANDBOX**

1. Los servicios ofrecidos por los servidores sandbox1.ufps.edu.co y sandbox1.ufps.edu.co, serán para uso exclusivamente académicos. La UFPS y el Programa de Ingeniería de Sistemas no se hace responsable de los contenidos expuestos por sus usuarios. En caso de existir información ilegal, será auditada y se procederá a informar a las autoridades competentes.
2. El usuario de SANDBOX, debe ser un estudiante activo  del Programa de Ingeniería de Sistemas de la UFPS. Los usuarios y sus aplicaciones  son dados de baja semestralmente. Es responsabilidad de sus usuarios realizar las copias de seguridad respectivas. Las aplicaciones web (php o java)  y los almacenes de datos (base de datos, archivos de texto plano, etc), solo serán resguardados durante el período académico lectivo.
3. El estudiante es responsable de cuidar su nombre de usuario y contraseña. En caso de alguna pérdida debe dirigirse PERSONALMENTE, con los administradores del sistema. Estas solicitudes NO son atendidas por medios virtuales.
4. El estudiante es responsable de las aplicaciones y datos que se encuentren bajo el dominio de su cuenta de usuario.
5. Todas las aplicaciones y sus datos, son de carácter académico y deben en su gran mayoría, estar bajo la tutela de un Profesor del Programa de Ingeniería de Sistemas. En caso de requerir para pruebas de software en proyectos de grado, este debe ser consultado con los administradores del sistema, con previa autorización del director del Proyecto.
6. Cada cuenta de usuario posee su propio directorio public\_html/ (para aplicaciones en php) y acceso al directorio de despliegue del servidor de aplicación Tomcat,  el cual puede ser usado libremente por el estudiante para ejecutar cualquier aplicación web, reservándose SANDBOX el derecho de deshabilitar cualquier aplicación que llegará a interferir o afectar la operación normal o alto rendimiento de los servidores, sin previo aviso. Para protección de todos los usuarios, SANDBOX se reserva el derecho de negar el servicio o suspender definitivamente el mismo a estudiantes que hayan abusado de los recursos de los servidores creando el riesgo de afectar la operación normal o alto rendimiento sin previo aviso.
7. Todos los servicios proveídos por SANDBOX deben ser usados para propósitos totalmente lícitos y académicos. La transmisión, almacenamiento o presentación de cualquier información, datos o materiales en violación de cualquier ley o reglamento aplicable local, estatal, federal o internacional, está prohibido. Esto incluye, pero no está limitado a: material con derechos de autor, material obsceno o agresivo, o material protegido por derechos de marca o similar. Ejemplos de contenido o enlaces inaceptables son: (i) Material pornográfico o explícitamente sexual que esté fuera de las leyes locales, estatales o federales aplicables a la materia. (ii) Software ilegal (“pirata”) (iii) Programas o archivos “hacker”.
8. Cualquier violación a material protegido por derechos de autor, cuya marca esté registrada o patentada, sin el expreso conocimiento del poseedor del derecho de autor, la marca registrada o los derechos patentados. SANDBOX, puede ser requerido por la Ley a borrar o bloquear el acceso a contenido bajo sospecha de violación de derechos de autor.
9. Configuración General del Servidor. SANDBOX,  se reserva el derecho de modificar archivos fuentes de configuración de las aplicaciones en php y java, para cuidar la integridad e infraestructura del mismo con la finalidad de proporcionarle un buen servicio, procurando no afectar en ninguno de los aspectos a los miembros de la comunidad académica del Programa.
10. Para aplicaciones que hagan uso de servicios de email, no se permite réplicas de email de más de 10 suscriptores por mensaje enviado. En caso de generar algún tipo de spam, la cuenta del estudiante será deshabilitada. Sea adopta una política de cero tolerancia al correo basura. El término “Correo basura” o "SPAM" incluye, pero no se limita a, el envío masivo de correo no solicitado y/o mensajes comerciales en Internet, así como “cartas de cadena”. Los administradores de SANDBOX, determinarán a su sola discreción violaciones a esta política por parte de los estudiantes. La venta a través de los Servicios de programas para generar correo basura está absolutamente prohibida.
11. Está prohibido el uso de los Servicios para crear o enviar virus, “gusanos” o “caballos de Troya” en Internet, así como el bombardeo de correo electrónico y ataques de negación de Servicio. Todos los estudiantes deberán abstenerse de involucrarse en actividades cuyo fin sea interrumpir o interferir con la habilidad de los demás usuarios de usar los Servicios o llevar a cabo sus actividades en Internet, o que puedan resultar en ello.
12. El "Hacking" y actividades relacionadas o similares están prohibidas. El "Hacking" incluye, pero no está limitado a las siguientes actividades: el acceso ilegal o no autorizado a computadoras, redes, cuentas y otros sitios restringidos, o la penetración o intento de penetrar medidas de seguridad y zonas no públicas.
13. Esta absolutamente prohibido el uso de los servicios para almacenar, enviar, desplegar, trasmitir, anunciar o poner a disposición del público en general pornografía infantil. La presencia y/o transmisión de pornografía infantil será notificada inmediatamente por a la UFPS y  a las autoridades competentes.
14. El uso de los servicios para llevar a cabo cualquier actividad ilegal está prohibido. Actividades ilegales incluyen, pero no están limitadas a: (i) almacenar, enviar, desplegar, trasmitir, anunciar o poner a disposición del público en general pirámides de estafa, información confidencial sobre tarjetas de crédito sin el permiso de sus propietarios (ii) introducir, publicar, enviar correos electrónicos o de otra manera transmitir cualquier contenido al que usted no tenga derecho de transmitir bajo alguna ley o bajo alguna relación contractual o fiduciaria (tal como información privilegiada, información propia o confidencial conocida o revelada como parte de una relación laboral o bajo acuerdos de confidencialidad); (iii) el no cumplir con leyes aplicables de privacidad en línea, así “acechar” o de otra manera acosar al público en general. (iv) introducir, publicar, enviar correos electrónicos o de otra manera transmitir cualquier contenido que infrinja cualquier patente, marca registrada, secretos comerciales, derechos de reproducción o cualquier otro derecho de propiedad de cualquiera de las partes o terceros. (v) La violación, de manera intencional o sin intención, de cualquier ley aplicable local, estatal, federal o internacional, incluyendo, pero no limitado a, regulaciones promulgadas por la Comisión de Vigilancia del Mercado de Valores, cualquier regla de cualquier nacional o de otra comisión. Infracciones a este apartado serán notificadas por la UFPS a las autoridades competentes, y se cooperará siempre que sea necesario con dichas autoridades.
15. Está prohibido el uso de los servicios para introducir, publicar, enviar por correo electrónico o de otra manera transmitir cualquier contenido que vaya en contra de la ley, perjudique, amenace, abuse, ataque, torture, difame, sea vulgar, obsceno, calumnie, invada la privacidad de otro, genere odio racial y/o étnico o de otra manera sea motivo de objeción.
16. Está prohibido falsificar, omitir, suprimir o enmascarar cabeceras de mensaje, direcciones de respuesta de mail y/o las direcciones IP para ocultar, encubrir o distorsionar el origen de un mensaje.

**LOS SERVIDORES ESTÁN CONSTANTEMENTE AUDITADOS**

**Y SE RIGEN BAJO LAS NORMAS LEGALES VIGENTES DE LA LEY COLOMBIANA DE DELITOS INFORMÁTICOS**

**ANEXO B. MANUAL DE USUARIOS DEL SERVIDOR SANDBOX**

# MANUAL DE USUARIO ADMINISTRADOR WAR



El administrador War es una herramienta WEB creada en JSP integrada al servidor Sandbox por estudiantes para estudiantes de Ingeniería de Sistemas, para que estos puedan administrar sus proyectos en .JSP en este servidor. Este proceso se describe a continuación paso a paso.

1. Se inicia sesión con el usuario y la clave asignada en el registro a Sandbox, el campo tipo de usuario debe mantenerse en “Usuario” y se realiza click en iniciar sesión.



1. Al entrar en la aplicación nos encontramos con una interfaz de usuario que costa de 6 puntos de importantes, de un label con la leyenda “Mis Archivo War que entre paréntesis contiene le número de war que posee el usuario.



1. Botón Inicio, el cual retorna a la página principal de la aplicación de usuario.
2. Botón Listar Mis WAR, el cual retorna a la página principal de la aplicación de usuario.
3. Botón Subir War, el cual permite subir un archivo WAR a la aplicación y guardarlo.



1. Botón Seleccionar archivo; se selecciona el archivo . war que se desea subir al servidor.



1. Botón Guardar; permite subir la aplicación al servidor, una vez subida nos muestra un link para acceder a la aplicación.



1. Botón Link, el cual lleva al link donde se encuentra la aplicación.
2. Botón Eliminar, el cual elimina un archivo WAR del servidor, una vez eliminado este se aprecia un mensaje que dice: “se ha Eliminado Archivo WAR con éxito”.



1. Botón Cerrar Sesión; el cual cierra la sesión del usuario en la aplicación, volviendo a la pantalla inicial.

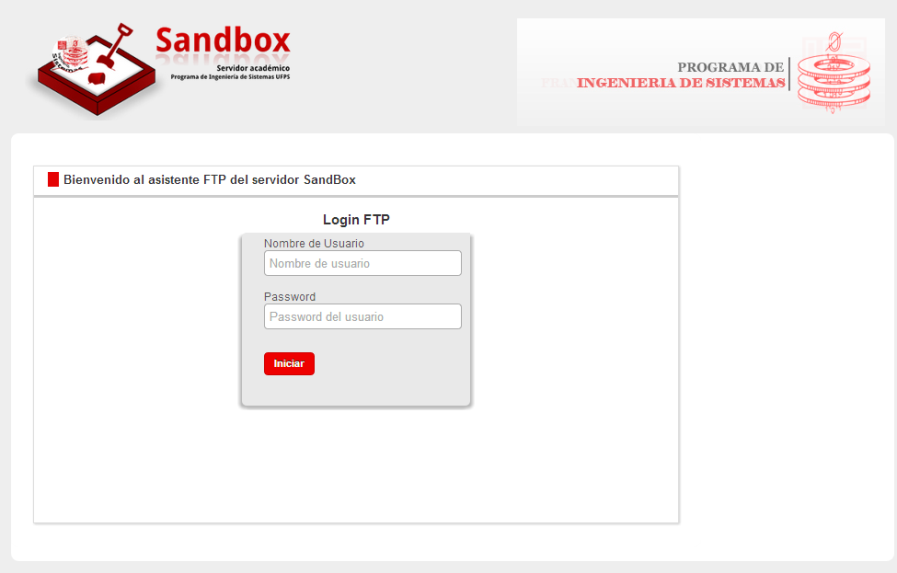
# MANUAL DE USUARIO DEL ADMINISTRADOR FTP



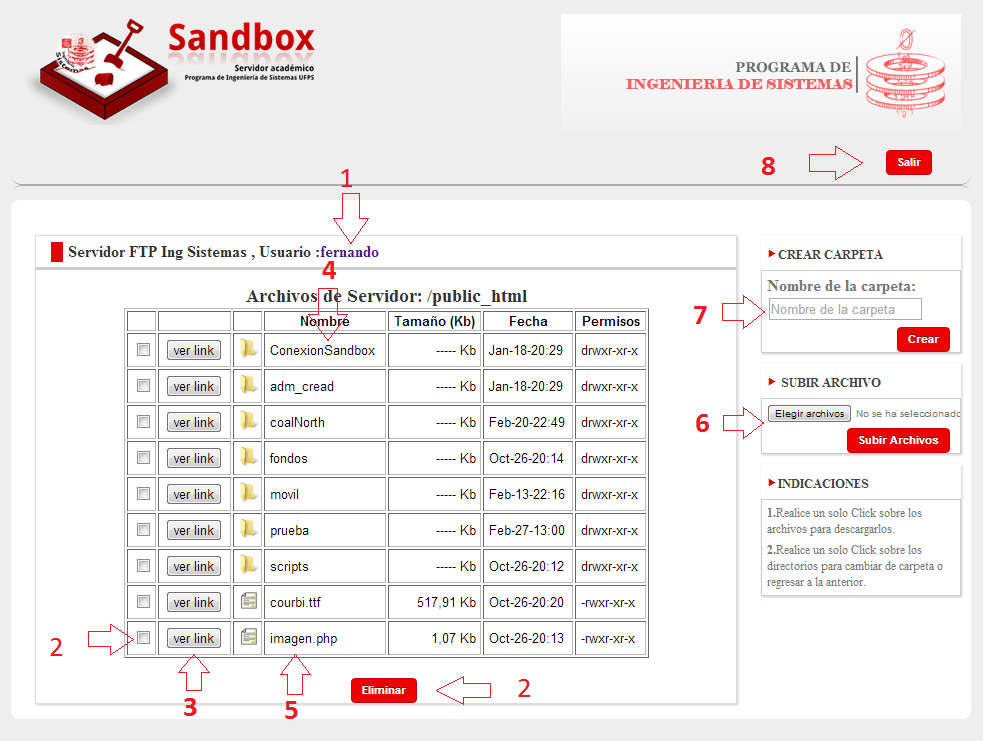
El administrador FTP es una herramienta WEB creada en JSP la cual utiliza un componente en Java creado por la compañía Apache, este componente permite realizar una sesión Ftp con un servidor.

El administrador FTP es una herramienta integrada al servidor Sandbox para que los estudiantes puedan acceder a su carpeta public del servidor. Este proceso se elabora en los siguientes pasos:

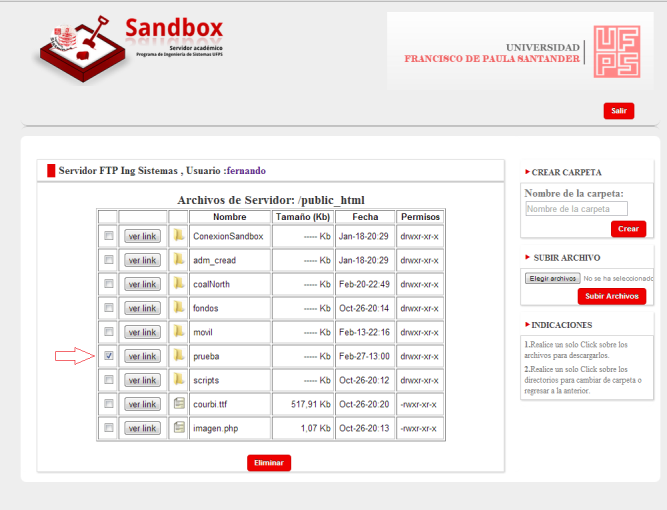
1. Se realiza el inicio de sesión con el usuario y la contraseña dada por Sandbox en el registro a este es sus respectivos campos, después se da click en el boto de Iniciar.



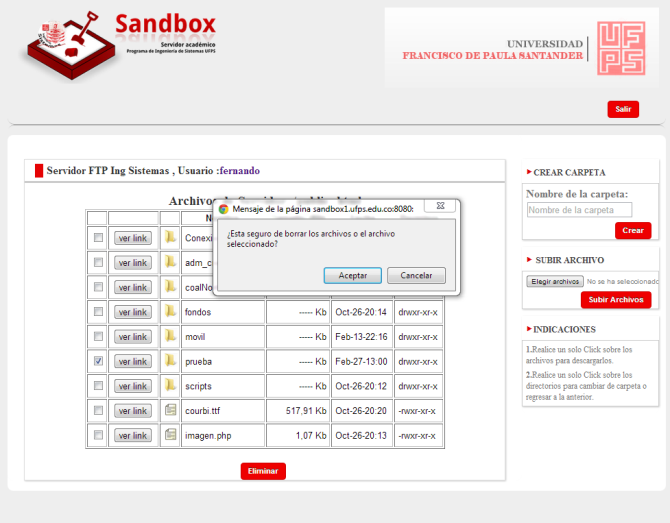
1. Una vez el inicio de sesión es correcto obtenemos una interfaz de usuario como la siguientes



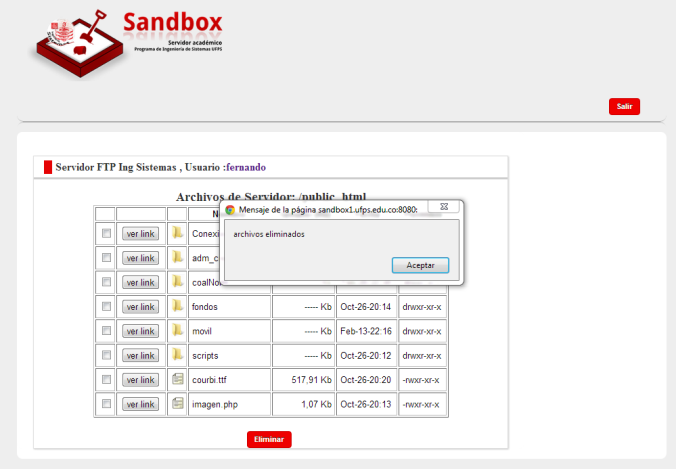
1. Botón Link; acceso para ver la url de la carpeta pública.
2. Proceso de eliminar, se cumple en 2 pasos:
3. Se selecciona el archivo o carpeta que se desea eliminar.



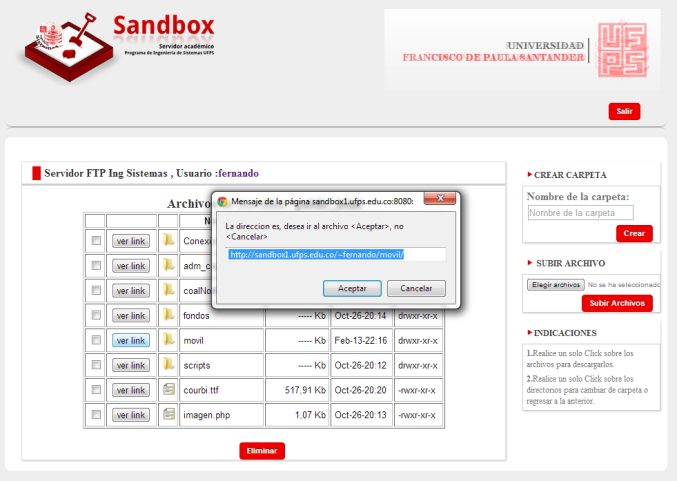
1. Se realiza click sobre el botón Eliminar, se abre una ventana de confirmación sobre si se desea eliminar o no.



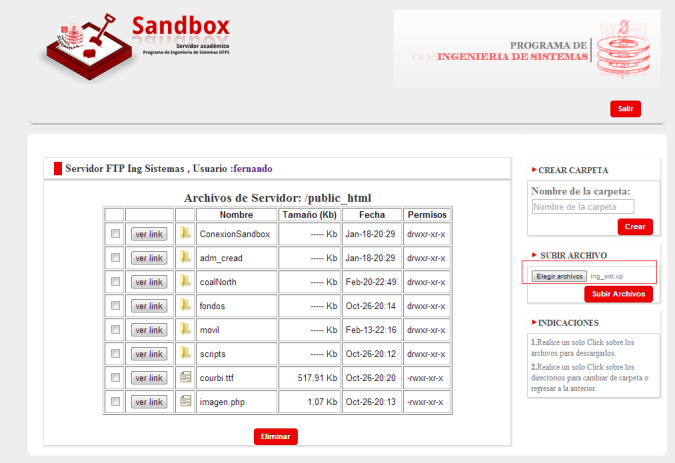
1. Al realizar el procedimiento exitosamente sale un mensaje emergente que dice que el archivo fue eliminado.



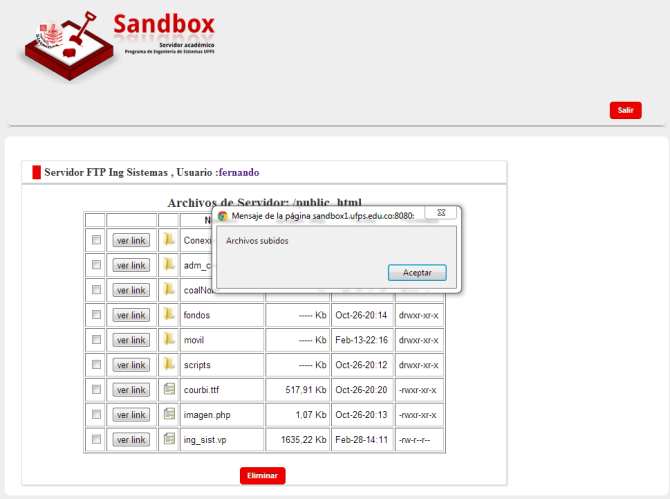
1. Botón Ver Link; este botón sirve para abrir el link directo a esa carpeta o archivo, aparte de esto permite copiar el link sin tener que abrirlo.



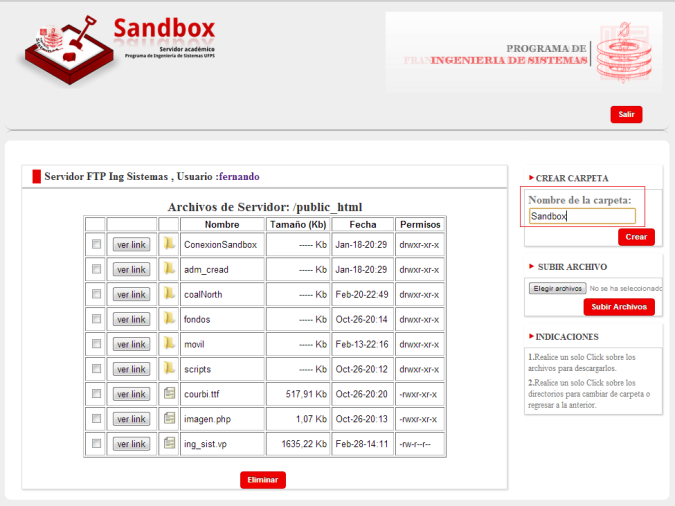
1. Botón Abrir Carpeta; al realizar click sobre el nombre de la carpeta para abrir esta, en el caso que no posea nombre sino “…” esto sirve para regresa a la carpeta contenedora de donde nos encontremos.
2. Botón Descargar Archivo; al realizar click sobre el nombre del archivo para descargar está a nuestro equipo.
3. Proceso subir archivo; este proceso se realiza en dos simples pasos.
4. Se realiza click sobre Elegir archivos y se selecciona este o varios.



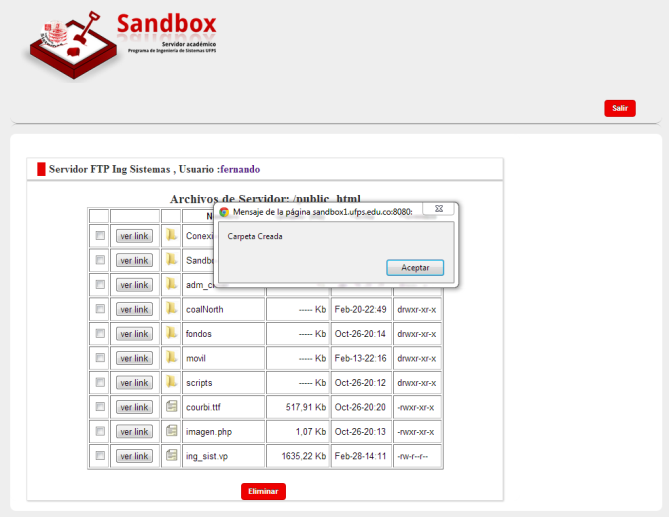
1. Realizar click en el botón Subir Archivo para completar la acción, al finalizar esta aparece una ventana con la leyenda Archivos subidos.



1. Proceso de Crear Carpeta; se realiza en dos simples pasos.
2. En el campo Nombre de carpeta, se escribe el nombre de la nueva carpeta.



1. Se realiza click en el botón crear, al realizar el proceso aparece una ventana emergente con la leyenda Carpeta Creada.



1. Botón Salir; al realizar click en este botón se cierra la sesión FTP con el servidor.