

Universidad de las Ciencias Informáticas

Facultad 2



**Personalización de Ubuntu Server 12.04 para el módulo
Gserver.**

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

Autores:

Diego Miguel Ferreiro García.


Julio César Quesada Arias.

Tutores:

Ing. Diannet Sospedra López.

Ing. Yor Alex Remond Recio.

“Habana, Junio, 2015”



PENSAMIENTO

“El aspecto fundamental en el cual la juventud debe señalar el camino es precisamente en el aspecto de ser vanguardia en cada uno de los trabajos que les compete.”

Che

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos al Centro Telemática de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Julio César Quesada Arias

Diego Miguel Ferreiro Garcia

Firma Autor

Firma Autor

Ing. Diannet Sospedra López

Ing. Yor Alex Remond Recio

Firma Tutor

Firma Tutor

DATOS DE CONTACTO

dsospedra@uci.cu Ing. Diannet Sospedra López, graduada de Ingeniero en Ciencias Informáticas en el año 2013. Pertenece al centro Telemática. Se desempeña como analista del proyecto Gestión de Recursos de hardware y Software.

reymond@uci.cu Ing. Yor Alex Remond Recio, graduado de Ingeniero en Ciencias Informáticas en el año 2014. Pertenece al centro de Telemática. Se desempeña como Asesor de Tecnología del centro.

AGRADECIMIENTOS

Agradecemos a todas las personas que de una forma u otra han colaborado a que este sueño se haga realidad.

DEDICATORIA

Dedicamos el presente trabajo a todas las personas que de una forma u otra han colaborado a que este sueño se haga realidad.

RESUMEN

El Centro de desarrollo Telemática, específicamente el proyecto Gestión de Recursos de Hardware y Software, cuyo objeto es mantener el control sobre los medios tecnológicos ya sean de hardware o software, actualmente está dividido en tres módulos: Gadmin, Gclient y Gserver. La instalación y configuración del último módulo antes mencionado se realiza de forma engorrosa, lo que provoca demora a la hora de su uso así como diversas inconformidades por parte de los clientes.

Con el objetivo de lograr elevar la calidad de los servicios brindados y agilizar los procesos de instalación y configuración de Gserver, se requiere el desarrollo de una solución que brinde la posibilidad de trabajar con este módulo de una forma sencilla y rápida. El presente trabajo permite la creación de una solución basada en la personalización de Ubuntu Server 12.04 para el módulo Gserver.

PALABRAS CLAVE

Desarrollo, Gserver, módulo, personalización, Ubuntu Server.

TABLA DE CONTENIDO

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	5
1.1 Conceptos Fundamentales.....	5
1.1.1 Software	5
1.1.2 Sistema Operativo	5
1.1.3 Hardware.....	5
1.2 Análisis de soluciones existentes.....	5
1.2.1 Sistema Operativo Canaima GNU/Linux 3.1.....	6
1.2.2 Ubuntu Desktop	6
1.2.3 Ubuntu Phone	6
1.2.4 Ubuntu Tablet.....	7
1.2.5 Ubuntu TV	7
1.2.6 Ubuntu for Android.....	7
1.2.7 Ubuntu Server	7
1.2.8 Ubuntu Business Desktop Remix	8
1.2.9 Ubuntu-china	8
1.3 Metodología, Lenguajes y Herramientas de desarrollo	9
1.3.1 Metodología de Desarrollo de Software	9
1.3.2 Metodología de desarrollo seleccionada. Programación Extrema.....	10
1.3.3 Lenguaje de programación Python.	10
1.3.4 ImgBurn.....	11
1.3.5 Visual Paradigm	11
1.3.6 Servidor web Nginx.....	12
1.3.7 Interfaz de entrada de servidor Web.....	12
1.3.8 Intérprete de comandos Linux Bourne again shell (Bash)	13
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	14
2.1 Propuesta de la solución	14
2.2 Modelación de Procesos del Negocio.....	15
2.3 Funcionalidades de la solución propuesta	16
2.4 Lista de Reserva de la Solución	17
2.5 Historias de Usuario	18

2.6	Planificación	20
2.6.1	Estimación de esfuerzo por HU	20
2.6.2	Plan de Iteraciones	21
2.6.3	Plan de Entregas	22
CAPITULO 3: DISEÑO E IMPLEMENTACION DE LA SOLUCION		24
3.1	Arquitectura del Sistema.....	24
3.1.1	Arquitectura en Capas.....	24
3.2	Patrones de Diseño.....	26
3.2.1	Patrones GRASP	26
3.2.2	Patrones GOF	27
3.3	Tarjetas Clase – Responsabilidad – Colaborador.....	27
3.4	Tareas de la Ingeniería	28
3.5	Pruebas.....	30
3.5.1	Pruebas unitarias	30
3.5.2	Pruebas de aceptación	30
CONCLUSIONES.....		33
RECOMENDACIONES.....		34
REFERENCIA BIBLIOGRÁFICA.....		35
BIBLIOGRAFÍA.....		37
ANEXOS.....		41

INTRODUCCIÓN

El mundo en la actualidad experimenta una serie de transformaciones económicas, sociales y culturales, en las cuales juegan un papel importante las Tecnologías de la Información y la Comunicación (TIC). El crecimiento de Internet, la rápida propagación de la información a través de la red de redes y la necesidad de facilitar el trabajo en todas las esferas de la sociedad hacen que las TIC sean cada vez más usadas.

Cuba ha mostrado interés en el uso de las TIC, es por ello que ha estado inmersa en las últimas décadas en un profundo y novedoso proceso de transformaciones educacionales y sociales. Como parte de los programas de la Batalla de Ideas, a partir de la cual se emprendieron nuevas acciones destinadas a elevar el nivel cultural de la población y su calidad de vida. En estas circunstancias y con la visión de futuro próspero y sostenible del Comandante en Jefe Fidel Castro Ruz, surge la Universidad de Ciencias Informáticas (UCI).

La UCI se divide en varios centros productivos, entre ellos el Centro de Telemática (TLM), perteneciente a la facultad número dos, el cual a su vez incluye varios proyectos como es el de Gestión de Recursos de Hardware y Software (GRHS). El mismo surge con la necesidad de llevar un control sobre los medios tecnológicos ya sean de hardware o software, así como cada una de las partes con que cuentan y evitar cualquier tipo de afectación a la integridad de estos.

Dicho proyecto consta de un sistema que permite realizar un inventario de hardware y software en una red de computadoras, para ello detecta incidencias, envía alertas, realiza reportes y configuraciones del sistema. Consiste en un agente instalado en las computadoras de la entidad, que enviará la información a un gestor donde se procesa y almacenan los datos obtenidos, además el sistema permite visualizar los datos obtenidos en una aplicación web.

La solución se divide en tres aplicaciones: Módulo Gclient, Módulo de Administración de Recursos de Hardware, Software, Incidencias y Alarmas (Gadmin) y el Módulo Gserver. La aplicación Gclient es la encargada de la recolección de inventarios de las computadoras, de la ejecución de acciones y detección de incidencias sobre las mismas. A esta información se puede acceder a través del módulo Gadmin que permite la visualización de los datos, realización de reportes e informes, así como seleccionar diferentes acciones de control a realizar sobre las máquinas al detectarse una incidencia.

Mediante el módulo Gserver se guarda en la Base de Datos los inventarios, trazas e incidencias y se le envía al cliente las configuraciones, envía notificaciones de actualización de la configuración y el identificador asignado al agente. Además Gserver permite enviar órdenes de control hacia el agente como son: reiniciar, suspender, inventariar, apagar e hibernar.

Actualmente la instalación del módulo Gserver se realiza sobre el Sistema Operativo (SO) Ubuntu Server 12.04. El SO Ubuntu Server 12.04 está orientado a servidores, permite instalar Ubuntu en un computador usado como servidor y no instala una interfaz gráfica de usuario por defecto (1).

El proceso de configuración e instalación del módulo Gserver en el SO Ubuntu Server 12.04 se realiza de forma engorrosa ya que requiere de la instalación de diferentes dependencias, plugins¹, librerías y configuraciones para su correcto funcionamiento. No existe la posibilidad de usar dicho módulo fuera de la universidad, a menos que se realice todo el proceso manual paso a paso. Además el proceso de instalación y configuración de Gserver se realiza cargando de los repositorios de la universidad que aunque están disponibles en algunas partes fuera de la misma no se puede garantizar su correcto funcionamiento.

Existe la necesidad de simplificar y facilitar este proceso para llevar a cabo la instalación del módulo Gserver de una forma profesional, por ello se propone realizar una personalización de Ubuntu Server 12.04 para el módulo Gserver.

Teniendo en cuenta la **situación problemática** anterior, se plantea como **problema a resolver**: ¿cómo integrar el módulo Gserver en la distribución Ubuntu Server 12.04?

Se define como **objeto de estudio**: la personalización de distribuciones GNU/Linux.

Se enmarca como **campo de acción**: la personalización de Ubuntu Server 12.04 para el módulo Gserver.

Se traza como **objetivo general**: desarrollar una distribución personalizada de Ubuntu Server 12.04 para el módulo Gserver.

Para darle cumplimiento al objetivo general de la investigación se trazaron las siguientes **tareas de investigación**:

¹ Plugins: Un plugin es aquella aplicación que, en un programa informático, añade una funcionalidad adicional o una nueva característica al software. Por lo tanto, puede nombrarse al plugin como un complemento.

- Establecimiento de los principales referentes teóricos - metodológicos que deben sustentar la gestión de la información en la realización de una distribución personalizada de Ubuntu Server 12.04 para el módulo Gserver.
- Valoración de la contribución del desarrollo de una distribución personalizada de Ubuntu Server 12.04 para el módulo Gserver.
- Selección de las diferentes herramientas y metodologías de desarrollo de software que faciliten el desarrollo de la solución propuesta.
- Desarrollo del análisis y diseño de la solución propuesta.
- Desarrollo de la solución propuesta como una distribución personalizada de Ubuntu Server 12.04 para el módulo Gserver.
- Realización de pruebas a cada una de las funcionalidades desarrolladas para comprobar su correcto funcionamiento.

En el desarrollo de la investigación se utilizan un conjunto de **métodos científicos**. El método científico de investigación es la forma de abordar la realidad, de estudiar la naturaleza, la sociedad y el pensamiento, con el propósito de descubrir su esencia y sus relaciones. Se clasifican en teóricos, empíricos y matemáticos-estadísticos los cuales están dialécticamente relacionados.

Métodos Teóricos

Histórico – Lógico: Permite determinar el comportamiento a través de los años, las necesidades históricas y tendencias actuales de la realización de instaladores para sistemas operativos.

Modelación: A través de este método se puede visualizar y elaborar los diferentes diagramas y modelos mediante abstracciones para explicar e interpretar la realidad y garantizar una correcta implementación de la solución.

Enfoque de Sistema: Es el encargado de que la información sea coherente y no exista polisemia, permitirá poder hacer uso de un solo término durante toda la investigación.

Analítico – Sintético: Método que permite separar los elementos que intervienen en la realización de una solución que permita la personalización de Ubuntu Server 12.04 para el módulo Gserver. Además permite reunir los elementos que tienen relación y lógica entre sí en la investigación en cuestión (como en un rompecabezas) hasta completar y demostrar la verdad del conocimiento.

Métodos Empíricos

Observación: El método de observación es sumamente útil en todo tipo de investigación: descriptiva, analítica y experimental. Es un procedimiento fácil de llevar a cabo y que exige técnicas de trabajo muy sencillas. La misma puede utilizarse en diferentes momentos dentro de la investigación. Permite además observar el estado en que se encuentra el proyecto GRHS y así saber hacia dónde debe ir encaminada la solución propuesta.

Entrevista: Se realiza a miembros de la dirección del proyecto GRHS con el objetivo de aclarar las características de la realización de una solución que permita la integración del módulo Gserver a una personalización de Ubuntu Server 12.04.

El trabajo de diploma se divide en tres capítulos, los cuales estarán estructurados de la siguiente forma:

Capítulo1. Fundamentación Teórica: en este capítulo se describen los conceptos fundamentales a tratar y se realiza un estudio sobre las aplicaciones similares existentes en el mundo y en el país. Se definen las herramientas y tecnologías a utilizar para el desarrollo de la solución propuesta.

Capítulo 2. Características del sistema: en este capítulo se exponen las características funcionales del sistema mediante las Historias de Usuario (HU) así como las no funcionales. Se realiza una propuesta del prototipo no funcional de la aplicación, y se estiman y planifican las HU para su posterior implementación.

Capítulo 3. Diseño e implementación de la solución: en este capítulo se define la arquitectura del sistema y los patrones de diseño a emplear. Además se organizan las clases fundamentales del sistema utilizando las tarjetas Clase – Responsabilidad – Colaborador (CRC). Se procede a desarrollar las tareas de la ingeniería que corresponden a las HU desarrolladas anteriormente, luego mediante las pruebas se verifica que el producto resultante cumpla con los requisitos definidos.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Introducción

A lo largo del desarrollo de este capítulo se profundizan algunos conceptos fundamentales para la realización de una solución que permita la personalización de Ubuntu Server 12.04 para el módulo Gserver. De igual forma se profundiza en las principales tecnologías, herramientas y técnicas de desarrollo propuestas para la solución del problema, así como la justificación de la selección de las mismas.

1.1 Conceptos Fundamentales

1.1.1 Software

Software es el equipamiento lógico e intangible de un ordenador. En otras palabras, el concepto de software abarca todas las aplicaciones informáticas, como los procesadores de textos, las planillas de cálculo y los editores de imágenes. El software es desarrollado mediante distintos lenguajes de programación, que permiten controlar el comportamiento de una máquina (2).

1.1.2 Sistema Operativo

El conjunto de programas informáticos que permite la administración eficaz de los recursos de una computadora es conocido como sistema operativo o software de sistema. Estos programas comienzan a trabajar apenas se enciende el equipo, ya que gestionan el hardware desde los niveles más básicos y permiten además la interacción con el usuario (3).

1.1.3 Hardware

La Real Academia Española define al hardware como el conjunto de los componentes que conforman la parte material (física) de una computadora.

Sin embargo, el concepto suele ser entendido de manera más amplia y se utiliza para denominar a todos los componentes físicos de una tecnología (4).

1.2 Análisis de soluciones existentes.

A continuación se muestran varios ejemplos de personalizaciones realizadas a diferentes SO en el mundo, que servirán para profundizar en la investigación y así conocer personalizaciones que logran integrar un sistema a una distribución Linux. Actualmente no existe una aplicación que logre integrar el módulo Gserver a un SO, pues Gserver se instala de forma manual sobre un sistema operativo ya instalado.

1.2.1 Sistema Operativo Canaima GNU/Linux 3.1

El instalador es una de las nuevas aplicaciones que se incluyen en el Sistema Operativo Canaima GNU/Linux 3.1 y es quizás una de las más importantes. La incorporación de nuevas aplicaciones dentro del CD de Canaima, hizo necesario que se explorara la eliminación de otras aplicaciones menos relevantes. Se tomó la decisión de hacer un instalador desde cero, en un lenguaje popular y ampliamente documentado (python), más completo, robusto y amigable, pero sobre todo: liviano. El nuevo instalador integra varias tecnologías, entre las que se puede mencionar: python-gtk2, python-cairo, python-parted, python-webkit, jquery, multihilos, entre otros (5).

A partir del núcleo de Ubuntu y su interfaz Unity, Canonical decidió desarrollar versiones enfocadas a diferentes dispositivos, tales como televisores, teléfonos inteligentes y futuras versiones móviles para tabletas. Estas son las actuales y en desarrollo de versiones de Ubuntu mantenidas por Canonical, las cuales van dirigidas a diferentes ambientes y dispositivos (6).

La personalización antes descrita no podrá ser utilizada para la obtención de la solución final de la investigación debido a que la misma va orientada a dispositivos como televisores, tabletas y móviles. A continuación se muestran varios ejemplos.

1.2.2 Ubuntu Desktop

Orientado para usuarios con computador de escritorio. Cuenta con miles de aplicaciones para entretenimiento y desarrollo (7).

Ubuntu desktop no podrá ser utilizada en la obtención de la solución final de la investigación ya que dicha personalización está orientada específicamente para computadoras de escritorio y no para servidores.

1.2.3 Ubuntu Phone

Orientado para usuarios con teléfonos inteligentes. Cuenta con una interfaz por gestos para teléfonos y con la posibilidad de iniciar el escritorio de Ubuntu desde un dock² con monitor externo (7).

La personalización antes descrita no podrá ser utilizada para la obtención de la solución final de la investigación pues la misma está orientada solo para teléfonos inteligentes.

² Dock o Muelle es un elemento de interfaz gráfica de usuario que permite a los usuarios iniciar, cambiar y monitorear aplicaciones.

1.2.4 Ubuntu Tablet

Orientado para usuarios con tabletas portátiles. Al igual que la versión para teléfonos, cuenta con una interfaz por gestos, aunque también puede utilizar una interfaz multitarea para utilizar dos aplicaciones al mismo tiempo, la posibilidad de usar cuentas multiusuario para proteger el contenido de cada usuario y por último cargar el escritorio de Ubuntu al conectar el dispositivo a un dock con teclado y mouse externo. Está en el mismo estado de desarrollo que Ubuntu Phone (6).

La personalización antes descrita no podrá ser utilizada para la obtención de la solución final de la investigación pues la misma está orientada solo a tabletas portátiles.

1.2.5 Ubuntu TV

Orientado para usuarios con televisor inteligente. Provee de una interfaz simple e intuitiva para organizar contenidos y servicios para TV (6).

Ubuntu TV no podrá ser utilizada para la obtención de la solución final de la investigación pues la misma no está orientada a servidores solo a televisores inteligentes.

1.2.6 Ubuntu for Android

Orientada para usuarios con teléfonos inteligentes con SO Android. Permite ejecutar el escritorio de Ubuntu al conectar el teléfono a un dock con monitor externo. También permite la sincronización de contactos y redes sociales con Android, y ejecución de aplicaciones Android desde el escritorio (6).

La personalización antes descrita no podrá ser utilizada para la obtención de la solución final de la investigación pues la misma está orientada solo a teléfonos inteligentes con SO Android.

1.2.7 Ubuntu Server

Es una distribución de Linux creada por Canonical y basada en Debian. Este sistema operativo ha ido creciendo exponencialmente en número de usuarios debido a su creciente acercamiento en cada una de sus nuevas versiones hacia un sistema con núcleo Linux cada vez más intuitivo y amigable para los usuarios medios, en el que tareas normales como por ejemplo la de descargar una aplicación e instalarla estén lo más automatizadas posibles. Ubuntu Server Edition es una versión de Ubuntu diseñada en específico para instalar un servidor y prestar servicios a clientes a través de la red.

No está hecho pensado en absoluto para ser usado en un equipo de escritorio, está probado que por defecto no se instala ningún contorno gráfico o navegador. La idea es que se instale exclusivamente el software necesario para prestar los servicios de redes, y omitir todo lo demás. De esta manera se aumentará la seguridad del servidor y se optimizará el uso de los recursos del mismo (1).

La personalización Ubuntu Server no brinda solución a la problemática antes planteada pues ninguna de sus versiones contiene el módulo Gserver. Por las características de esta personalización, principalmente la de estar orientada a servidores, se decide por parte del equipo de desarrollo y como una especificación del cliente utilizar la versión 12.04 y así darle solución a la investigación que es obtener una personalización de Ubuntu Server 12.04 para el módulo Gserver.

1.2.8 Ubuntu Business Desktop Remix

Orientada al sector empresarial con varias herramientas instaladas por defecto, tales como: Adobe Flash Plugin, VMware View, OpenJDK 6 Java, Canonical Landscape, se han eliminado aplicaciones sociales y juegos. Ofrece soporte para Windows RDP 7.1 y el diagrama Microsoft Visio en LibreOffice Draw (8).

La personalización antes descrita no podrá ser utilizada para la obtención de la solución final de la investigación pues la misma está orientada solo a computadoras de escritorio para realizar diversos trabajos pero no facilita el trabajo para servidores.

1.2.9 Ubuntu-china

Software libre, variante de Linux, contará con una serie de características adaptadas para los usuarios chinos. La primera versión está dirigida a usuarios de ordenadores de mesa y portátiles. De esta manera, en un principio, Ubuntu Kylin incorporará el calendario chino, métodos para introducir caracteres y rápidos accesos a los servicios de música más importantes. Además de estas propiedades, este sistema operativo basado en Raring Ringtail añadirá después algunos de los servicios web más conocidos del país, entre los que se incluye el portal de compras de Taobao, mapas de Baidu, así como editores de imágenes y otras herramientas para el sistema. El desarrollo del código se lleva a cabo en un laboratorio de Beijing, en el que participarán tanto ingenieros de Canonical como varias agencias chinas (9).

Este acuerdo de colaboración para desarrollar un SO libre es una de las medidas que está tomando China para conseguir el objetivo de que, cada vez, más personas utilicen

SO abiertos, evitando el uso de los SO cerrados, muchos controlados por empresas norteamericanas.

La personalización antes descrita no podrá ser utilizada para la obtención de la solución final de la investigación pues la misma está orientada a especificaciones para usuarios chinos y no brinda solución a la situación problemática antes planteada.

Conclusiones de las soluciones analizadas

A través de la investigación realizada a personalizaciones existentes en el mundo a distintos SO, se pudo arribar a la conclusión de que ninguna de las mismas brinda una solución a la situación problemática anteriormente planteada. El estudio realizado ofrece soluciones reales de como poder integrar archivos a SO, además facilitaron vías para la creación de script y soluciones para la modificación de la interfaz gráfica mostrada inicialmente por el SO.

1.3 Metodología, Lenguajes y Herramientas de desarrollo

1.3.1 Metodología de Desarrollo de Software

Una metodología de desarrollo de software es un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores guiando la realización del nuevo software (10).

Las metodologías de desarrollo de software ágil buscan la satisfacción del cliente y la entrega temprana del software; equipos de proyecto pequeños y con alta motivación; un mínimo de productos de trabajo de la ingeniería de software; y simplicidad general del desarrollo. Las metodologías ágiles están orientadas principalmente donde existe gran incertidumbre con requisitos desconocidos o variables. El cliente es parte del proceso de desarrollo, lo que posibilita la retroalimentación constante y las respuestas rápidas a los cambios en el negocio debido a su gran capacidad de respuesta a los cambios (10).

Se decide centrar la investigación en las metodologías ágiles, debido a que son capaces de entregar un producto funcional en un breve intervalo de tiempo y por la interacción que mantienen en todo el ciclo de desarrollo con el cliente, características que coinciden con la situación existente para el desarrollo del presente trabajo de diploma.

Una de las metodologías ágiles de desarrollo de software más exitosa en la actualidad y ampliamente utilizada a nivel mundial es Programación Extrema (XP), la cual se basa en la simplicidad, la comunicación y reutilización del código desarrollado. Evita elevados

e innecesarios costos ya que en otras metodologías se emplea demasiado tiempo y recursos en cambiar la documentación de la planificación para que se asemeje al código; para esto, intenta implementar una forma de trabajo que se adapte fácilmente a las circunstancias aumentando considerablemente la productividad del equipo.

Para el proceso de desarrollo XP brinda las siguientes ventajas:

- Promueve el trabajo en equipo, se preocupa en todo momento del aprendizaje de los desarrolladores y establece un buen clima de trabajo.
- No presenta resistencia a cambios durante el proceso de desarrollo.
- Presenta iteraciones cortas que permiten obtener versiones funcionales del producto.
- Planificación más transparente para los clientes, conocen las fechas de entrega de funcionalidades vitales para su negocio.
- La presión está a lo largo de todo el proyecto y no en una entrega final.

1.3.2 Metodología de desarrollo seleccionada. Programación Extrema

Como características del trabajo de diploma a desarrollar se encuentran que el equipo de desarrollo está compuesto por dos desarrolladores, el cliente se encuentra en la universidad, específicamente dentro del proyecto GRHS perteneciente a la facultad número dos.

Basándose en lo anteriormente expuesto y teniendo en cuenta que se trata de un proyecto pequeño que no necesita la generación de tantos artefactos, donde el intercambio de opiniones con el cliente juega un papel fundamental a lo largo de todo el proceso de desarrollo de software. Con el propósito principal de alcanzar un producto que satisfaga las necesidades del cliente en el menor tiempo posible y con la calidad requerida, se decide que la metodología XP guíe todo el proceso de desarrollo del software.

1.3.3 Lenguaje de programación Python.

Python: es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible.

Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa³ y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma (11).

Python es usado durante la investigación para la realización del instalador del módulo Gserver, facilitando la realización del mismo y brindando un entorno de trabajo sencillo a los programadores.

1.3.4 ImgBurn.

ImgBurn es un programa para Microsoft Windows y GNU/Linux (mediante Wine) que permite hacer imágenes de discos compactos, DVD, HD DVD, Blu-ray y grabarlas. Permitirá realizar una imagen ISO⁴ de la solución obtenida. Aunque no se puedan extraer directamente archivos concretos de un vídeo DVD, ImgBurn hace sus imágenes en formato ISO, el cual puede ser abierto con la mayoría de los compresores actuales que lo soporten como por ejemplo WinRAR, y extraer manualmente cualquier archivo contenido en la imagen como si se tratara de un fichero comprimido (12).

ImgBurn permitirá realizar la integración del módulo Gserver al SO Ubuntu Server 12.04 brindando la posibilidad de obtener una imagen ISO con la solución.

1.3.5 Visual Paradigm

Visual Paradigm es una herramienta CASE: Ingeniería de Software Asistida por Computación. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Visual Paradigm ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Constituye una herramienta privada disponible en varias ediciones, cada una destinada a satisfacer diferentes necesidades: Enterprise, Professional, Community, Standard, Modeler y Personal. Existe una alternativa libre y gratuita de este software, la versión Visual Paradigm UML 6.4 Community Edition (Community Edition, ya que existe la Enterprise, Professional, etc). Fue diseñado para una amplia gama de usuarios interesados en la

³ Programación imperativa: Es un paradigma de programación que describe la programación en términos del estado del programa y sentencias que cambian dicho estado. Los programas imperativos son un conjunto de instrucciones que le indican al computador cómo realizar una tarea.

⁴ Una imagen ISO es un archivo donde se almacena una copia o imagen exacta de un sistema de ficheros, normalmente un disco compacto, un disco óptico, como un CD, un DVD, pero también soportes USB.

construcción de sistemas de software de forma fiable a través de la utilización de un enfoque Orientado a Objetos.

Visual Paradigm se utiliza en la investigación durante el proceso de modelado del negocio así como en la realización de las tarjetas clase –responsabilidad-colaborador (CRC).

1.3.6 Servidor web Nginx

Está orientado a ser una alternativa estable y de alto rendimiento a otros muy implantados como Apache o Microsoft IIS, y desde luego en los últimos tiempos está teniendo un éxito notable que ha hecho que su creador acabe creando una empresa para dar soporte empresarial y profesional a los interesados en usar Nginx de una forma más avanzada (13).

En nginx se incluyen mejoras en el soporte a HTTP Proxy, con la reutilización de conexiones keepalive⁵ a servidores, consolidación de peticiones múltiples y simultáneas, una configuración más flexible para el DNS resolver o el soporte de redirecciones proxy (14).

Teniendo en cuenta las características antes mencionadas y que el servidor web nginx es el utilizado actualmente en el proceso de instalación y configuración manual del módulo Gserver, se decide su utilización para mostrar la interfaz del módulo Gserver luego de creada la solución.

1.3.7 Interfaz de entrada de servidor Web

Web Server Gateway Interface (WSGI), es decir, Interfaz de entrada de servidor Web. Esta es una especificación simple y universal de interfaz entre los servidores web y aplicaciones web o frameworks para el lenguaje de programación Python (15).

Actualmente esta plataforma es la utilizada en el proceso de instalación y configuración de forma manual del módulo Gserver, por lo que se decide continuar con su utilización en la automatización del proceso antes mencionado.

⁵ Una conexión HTTP/1.1 con keepalive permite realizar más de una petición por la misma conexión establecida, evitando establecer una conexión nueva para cada parte de la web (hoja de estilos, imágenes...).

1.3.8 Intérprete de comandos Linux Bourne again shell (Bash)

Bash es un programa informático cuya función consiste en órdenes. Está basado en la shell de Unix y es compatible con POSIX. Fue escrito para el proyecto GNU y es el intérprete de comandos por defecto en la mayoría de las distribuciones de GNU con Linux (16).

Es un intérprete que ayuda al administrador a realizar la mayor parte de las tareas necesarias, tanto en la automatización como en el arranque del sistema. Se utiliza especialmente (no exclusivamente) en sistemas Unix (17).

Es utilizado en la modificación del archivo txt.cfg para lograr una transformación de la interfaz gráfica principal de Ubuntu Server 12.04. Además se utilizó en la creación del script extractor para lograr que el repositorio local antes creado sea utilizado por el SO luego de la descompresión.

Conclusiones parciales:

A través del desarrollo de la investigación se pudo llegar a las siguientes conclusiones:

- Se identificaron las necesidades existentes en la realización de una solución que permita la integración del módulo Gserver a una personalización de Ubuntu Server 12.04 en la UCI.
- Se definió el diseño teórico de la investigación, así como las tareas que facilitan las vías para cumplir los objetivos propuestos.
- Se utilizó un conjunto de métodos científicos, ya sean teóricos y empíricos, todos con el objetivo de guiar la investigación realizada.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Introducción

En el presente capítulo se abordan los temas relacionados con la fase de planificación de la metodología de desarrollo XP, se elabora una propuesta del sistema a desarrollar y se exponen las características del mismo para un mejor entendimiento en su desarrollo. Se confeccionan las Historias de Usuarios (HU) para cada iteración definida, con vista a documentar los procedimientos y técnicas empleados, proporcionando una mejor visión sobre lo que el cliente desea y además se realiza un análisis de la estimación del esfuerzo por cada HU.

2.1 Propuesta de la solución

Se propone el análisis y diseño de una solución informática que permita la instalación y configuración del módulo Gserver, mediante una personalización del sistema operativo Ubuntu Server 12.04; de esta forma se facilita que la solución sea capaz de instalar de forma sencilla el módulo Gserver antes mencionado.

Para obtener la solución final se propone la realización de un repositorio local que contendrá las dependencias generales para la instalación del módulo Gserver. Además se requiere la creación de una carpeta que contendrá varias dependencias en sus versiones específicas que necesita el módulo Gserver para su correcto funcionamiento.

Otro elemento importante para obtener la solución final es la realización de un instalador que debido a la complejidad que presenta en cuanto a cantidad de instalaciones, configuraciones y asignaciones de permisos permita automatizar este proceso. Para la realización del instalador se propone utilizar el lenguaje de programación orientado a objeto python, lo que permitirá una mejor organización del código.

Se propone crear varios script que faciliten el proceso de instalación y configuración del sistema, para su confección se propone utilizar el intérprete de comandos bash. Con la creación del script config se configuran los certificados SSL por defecto y se garantiza la gestión de usuario y contraseña por defecto del sistema. En el script templates se definen los valores que presentará la solución por defecto de certificados SSL así como usuario y contraseña. Luego de la instalación del SO Ubuntu Server 12.04 se lleva a cabo el proceso de instalación y configuración del módulo Gserver mediante el script postinst. Los script postm y prem se deben crear para lograr la correcta eliminación de

archivos no deseados durante y después del proceso de instalación de la solución propuesta.

Se pretende realizar un cambio en la interfaz principal a la hora de la instalación del SO Ubuntu Server 12.04, para lograrlo se debe modificar el archivo `txt.cfg`. También se propone realizar un cambio de la imagen principal del Ubuntu Server 12.04 para ello se propone modificar el archivo `splash.pcx`. Se debe modificar el archivo `ubuntu-server.seed` que se encuentra en la carpeta `preseed`, lo que posibilitará la ejecución de un script para configurar los repositorios que contiene el SO con el repositorio local antes creado y permitirá además poder ejecutar el instalador del módulo Gserver.

Se propone igualmente recalcular el MD5 ⁶ para la integridad y el correcto funcionamiento del SO con las modificaciones realizadas, además de garantizar la seguridad de la solución.

Finalmente se propone la integración del instalador del módulo Gserver, el repositorio local y la personalización del sistema operativo Ubuntu Server 12.04 para así obtener una imagen ISO con la personalización final de Ubuntu Server 12.04 para el módulo Gserver.

2.2 Modelación de Procesos del Negocio

Para una mejor comprensión del funcionamiento actual en el proceso de instalación y configuración del módulo Gserver en el SO Ubuntu Server 12.04 se realiza la modelación del proceso de negocio.

El especialista es el encargado de llevar a cabo todo el proceso relacionado con la instalación y configuración de Gserver. Es el encargado de realizar acciones como instalar dependencias y paquetes, instalar y configurar el Sistema Gestor de Bases de Datos (SGBD), copiar en la ruta específica la carpeta de instalación de Gserver, configurar permisos, crear los directorios para el almacenamiento de archivos estáticos. Además se encarga de crear el directorio de trazas, mover los archivos estáticos a las carpetas anteriormente creadas y configurar los parámetros de conexión a la Base de Datos (BD). También tiene la función de sincronizar con la BD desde la propia raíz del proyecto, crear una nueva aplicación de uWSGI, configurar el archivo uWSGI y habilitar el sitio haciendo un enlace simbólico desde las aplicaciones habilitadas. Luego pasa a

⁶ Es un algoritmo que proporciona un código asociado a un archivo o un texto concretos. De esta forma, a la hora de descargar un determinado archivo, como puede ser un instalador, el código generado por el algoritmo, también llamado *hash*, viene "unido" al archivo.

eliminar la aplicación predeterminada habilitada de la instalación de nginx, configurar la conexión SSL al servidor, crear y configurar una nueva aplicación de nginx unido a las configuraciones finales. Finalmente se encarga de cargar el backup con los nomencladores de la BD, configurar en la BD los sistemas de ficheros e interfaces definidos, configurar en la BD las plataformas definidas así como configurar en la BD el período de inactividad.

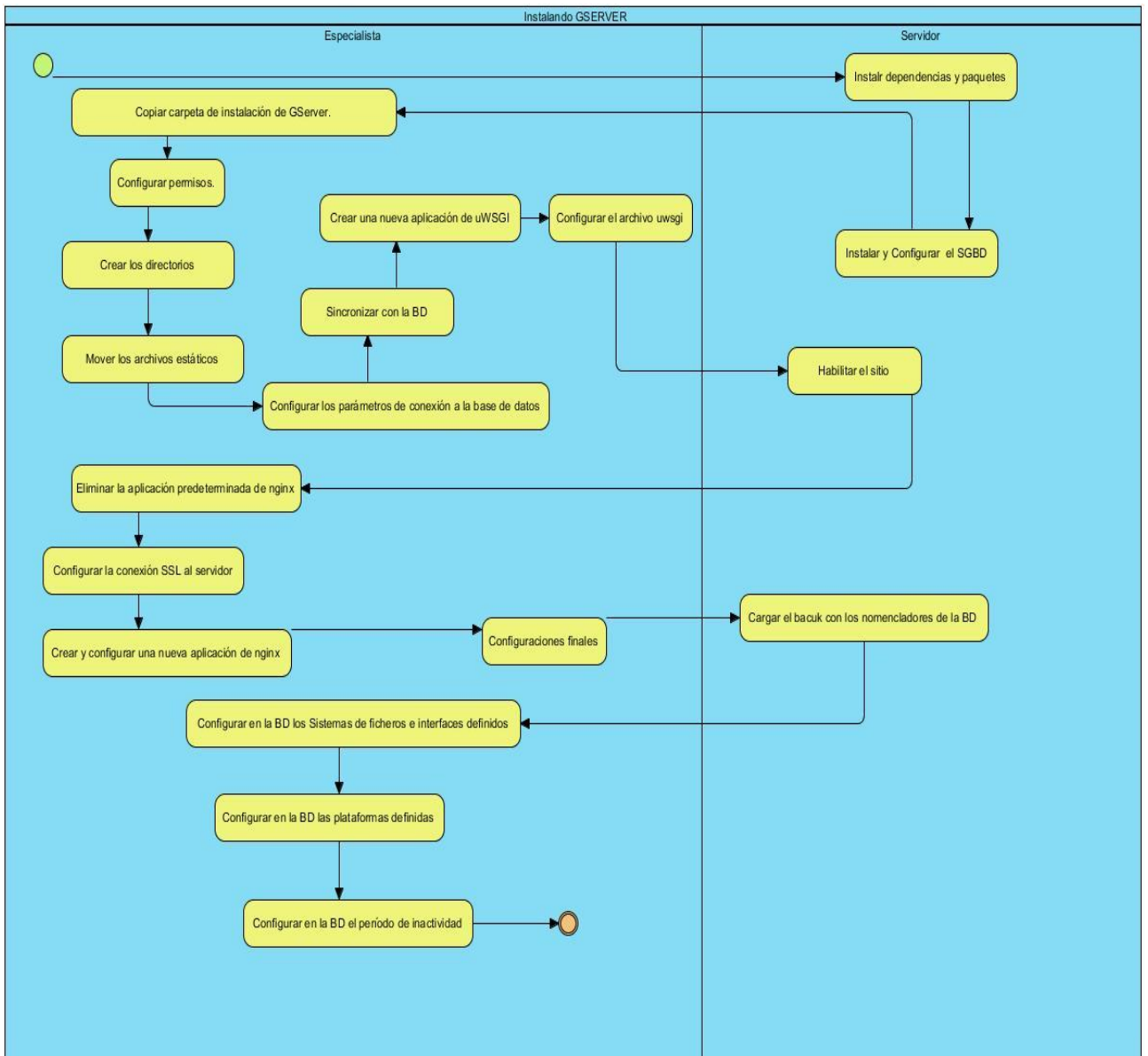


Figura1: Diagrama de Procesos del Negocio.

2.3 Funcionalidades de la solución propuesta

- Automatizar la instalación de dependencias y paquetes.
- Automatizar la instalación y Configuración del SGBD.

- Automatizar la copia en la ruta específica la carpeta de instalación de GServer.
- Automatizar la Configuración de permisos.
- Crear los directorios para el almacenamiento de archivos estáticos.
- Crear el directorio de trazas.
- Mover los archivos estáticos a las carpetas anteriormente creadas.
- Configurar los parámetros de conexión a la base de datos.
- Sincronizar con la BD desde la propia raíz del proyecto.
- Crear una nueva aplicación de uWSGI.
- Configurar el archivo uwsgi.
- Habilitar el sitio haciendo un enlace simbólico desde las aplicaciones habilitadas.
- Eliminar la aplicación predeterminada habilitada de la instalación de nginx.
- Configurar la conexión SSL al servidor.
- Crear y configurar una nueva aplicación de nginx.
- Configuraciones finales.
- Cargar el backup con los nomencladores de la BD.
- Configurar en la BD los Sistemas de ficheros e interfaces definidos.
- Configurar en la BD las plataformas definidas.
- Configurar en la BD el período de inactividad.

2.4 Lista de Reserva de la Solución

A continuación se muestran las propiedades o cualidades que la solución en desarrollo debe poseer:

Usabilidad

Se necesitará una preparación previa para operar con la solución. Se requiere un nivel básico de conocimientos de computación para interactuar satisfactoriamente con la solución.

Seguridad

La seguridad será tratada por el proyecto donde será usada la solución en este caso el proyecto GRHS. Además se utiliza md5sum como algoritmo de seguridad que sirve para verificar la integridad de los datos.

Hardware

Para la instalación de la solución se debe disponer de una computadora de 1GB⁷ de memoria RAM, con una tarjeta de red y 10 GB de capacidad en disco.

Al finalizar el desarrollo de la solución se entregará al cliente un Manual de Ayuda para los usuarios, que les servirá para aprender a interactuar con la solución.

2.5 Historias de Usuario

En XP la gestión de requisitos es extremadamente simple, el cliente escribe y prioriza las HU que expresan las necesidades del sistema. Los programadores estiman el esfuerzo asociado y las dependencias entre ellas.

Las mismas deben tener una mínima descripción como para que los programadores puedan realizar una estimación poco riesgosa del tiempo que llevará su desarrollo. Cuando llega el momento de la implementación, los desarrolladores dialogan directamente con el cliente para obtener todos los detalles necesarios.

La prioridad en el negocio:

Alta: Se le otorga a las HU que resultan funcionalidades fundamentales en el desarrollo del sistema, a las que el cliente define como principales para el control integral del sistema.

Media: Se le otorga a las HU que resultan para el cliente como funcionalidades a tener en cuenta, sin que estas tengan una afectación sobre el sistema que se esté desarrollando.

Baja: Se le otorga a las HU que constituyen funcionalidades que sirven de ayuda al control de elementos asociados al equipo de desarrollo, a la estructura y no tienen nada que ver con el sistema en desarrollo.

El riesgo en su desarrollo:

Alto: Cuando en la implementación de las HU se considera la posible existencia de errores que lleven a la inoperatividad del código.

Medio: Cuando pueden aparecer errores en la implementación de la HU que puedan retrasar la entrega de la versión.

⁷ GB: unidad de medida de almacenamiento informático normalmente abreviada como GB.

Bajo: Cuando pueden aparecer errores que serán tratados con relativa facilidad sin que traigan problemas mayores para el desarrollo del proyecto.

Las HU son representadas mediante tablas divididas por diferentes secciones, a continuación se muestra un ejemplo de estas.

Tabla 1: Muestra de una Historia de Usuario.

Historia de Usuario	
Número: (Número de la HU incremental en el tiempo)	Nombre de Historia de Usuario: (El nombre de la HU sería para identificarlas mejor entre los desarrolladores y el cliente)
Modificación de Historia de Usuario Número: (Si sufrió alguna modificación anterior)	
Usuario: (Se citan los desarrolladores responsables de la implementación de la HU)	Iteración Asignada: (Número de la iteración)
Prioridad en negocio: (Alto / Medio / Bajo)	Puntos estimados: (El tiempo estimado en semanas que se demorará el desarrollo de la HU)
Riesgo en Desarrollo: (Alta / Media / Baja)	Puntos Reales: (El tiempo que se demoró en realidad el desarrollo de la HU)
Descripción: (Breve descripción de la HU)	
Observaciones: (Señalamiento o advertencia del sistema)	
Prototipo de interfaz: (Prototipo de interfaz si aplica)	

A continuación se expone una muestra de una de las HU que presenta alta prioridad en el negocio, el resto de las HU definidas por el equipo de desarrollo en conjunto con el cliente se encuentran en la sección Anexos I.

Tabla 2: HU # 1 Instalar dependencias y paquetes.

Historia de Usuario	
Número: 1	Nombre de Historia de Usuario: Instalar dependencias y paquetes.
Modificación de Historia de Usuario Número: Ninguna.	
Usuario: Diego Miguel Ferreiro Garcia, Julio Cesar Quesada Arias.	Iteración Asignada: 1
Prioridad en negocio: Alta.	Puntos estimados: 1
Riesgo en Desarrollo: Alto.	Puntos Reales: 1
Descripción: Permitir instalar dependencias y paquetes necesarios para el funcionamiento de Gserver.	

2.6 Planificación

Durante la fase de planificación se realiza una estimación del esfuerzo que costará implementar cada HU. Este se expresa utilizando como medida el punto. Un punto se considera como una semana ideal de trabajo donde los miembros de los equipos de desarrollo trabajan el tiempo planeado sin ningún tipo de interrupción. Esta estimación incluye todo el esfuerzo asociado a la implementación de la HU.

2.6.1 Estimación de esfuerzo por HU

Para el desarrollo de la aplicación propuesta se realizó una estimación de esfuerzo por cada una de las HU identificadas:

Tabla 3: Estimación del esfuerzo por HU.

Historia de Usuarios	Puntos de Estimación
Instalar dependencias y paquetes.	1
Instalación y Configuración del SGBD.	2
Enrutamiento de la carpeta de instalación de Gserver y configuración de permisos.	2
Creación de directorios, así como la asignación de permisos.	2
Configuración y sincronización de la BD.	1
Creación de la aplicación de uWSGI.	2
Realizar enlace simbólico y eliminar la aplicación de nginx predeterminada.	2
Configuración de la conexión SSI.	1
Creación y configuración de una nueva aplicación de nginx.	1

2.6.2 Plan de Iteraciones

Una vez identificadas las HU y estimado el esfuerzo dedicado a la realización de cada una de estas se realiza la planificación de la etapa de implementación de la aplicación. Para una mejor organización en el desarrollo del trabajo el equipo de desarrollo dividió la implementación en 2 iteraciones:

Iteración 1:

En esta primera iteración se implementarán las HU desde la 1 hasta la 4 que dependen unas de otras, o sea el desarrollo de cada una de ellas depende de la anterior.

Iteración 2:

En esta iteración se implementarán las HU desde la 5 hasta la 10 pues dependen de las HU que les preceden, es necesario para su implementación que estén vencidas la HU que en la iteración 1 se hacen referencia, para entonces comenzar con las creaciones y configuraciones la BD, conexiones SSI y servidor web.

Plan de duración de las iteraciones

El plan de duración de las iteraciones es el encargado de relacionar las HU que van a ser implementadas con cada una de las iteraciones, además muestra la duración de cada iteración y el orden de realización de las HU.

Tabla 4: Plan de duración de iteraciones.

Iteración	Orden de las HU a implementar	Duración (semanas)	Total
1	1,2,3,4	7	
2	5,6,7,8,9	7	

2.6.3 Plan de Entregas

El plan de entrega detalla la fecha de fin de cada una de las iteraciones y los productos obtenidos divididos en subsistemas.

Tabla 5: Plan de entregas.

Sistema	Final Iteración 1 (25 de marzo del 2015)	Final Iteración 2 (20 de mayo del 2015)
---------	---	--

Conclusiones parciales

En este capítulo se logró obtener una mejor visión sobre lo que se desea realizar por parte del cliente mediante la elaboración de las HU y la propuesta del sistema. Además se identificaron las funcionalidades y características que la solución debe cumplir. Con la obtención del plan de entregas y la planificación se ha logrado delimitar el ciclo de desarrollo de la solución.

CAPITULO 3: DISEÑO E IMPLEMENTACIÓN DE LA SOLUCIÓN

Introducción

En el presente capítulo se detalla la arquitectura de la solución, así como los patrones de diseño utilizados en el desarrollo de la misma. Se definen las tarjetas Clase – Responsabilidad – Colaborador (CRC) para identificar y organizar las clases orientadas a objetos. Además se exponen las tareas de ingeniería generadas por cada HU, las cuales han sido utilizadas como base para la implementación de la solución. Se evalúa la calidad de la aplicación a través de las pruebas de software realizadas al sistema.

3.1 Arquitectura del Sistema

La arquitectura de un sistema constituye un modelo relativamente pequeño e intelectualmente comprensible de cómo está estructurado el sistema, cómo trabajan juntos sus componentes y la relación entre ellos. La arquitectura destaca las decisiones iniciales relacionadas con el diseño que tendrán un impacto profundo en todo el trabajo de la ingeniería del software que le sigue y lo que también resulta importante, en el éxito final del sistema.

3.1.1 Arquitectura en Capas

La arquitectura basada en capas simplifica la comprensión y la organización del desarrollo de sistemas complejos, reduciendo las dependencias de forma que las capas más bajas no son conscientes de ningún detalle o interfaz de las superiores, además ayuda a identificar qué puede reutilizarse (18). En el caso del sistema propuesto se define la arquitectura en capas para guiar la construcción de la solución teniendo en cuenta estas características y las ventajas que brinda.

Entre las ventajas de la arquitectura en capas se encuentra que:

- Soporta un diseño basado en niveles de abstracción crecientes, lo cual a su vez permite a los desarrolladores la partición de un problema complejo en una secuencia de pasos incrementales.
- El desarrollo se lleva a cabo en varios niveles, y en el caso de que exista algún error o la necesidad de algún cambio obligatorio, solo es necesario cambiar el nivel en cuestión, sin afectar el correcto funcionamiento del resto del sistema.
- Posibilita la reutilización de código debido a las características propias de su implementación en capas.

El sistema en desarrollo se divide en tres capas lógicas distintas:

✚ Capa de Interfaz

Es la que se encarga de que el sistema interactúe con el usuario y viceversa, muestra el sistema al usuario y le presenta la información. Esta capa se comunica únicamente con la capa de solución (10).

✚ Capa de Solución

Es donde residen las funciones que se ejecutan y se procesa la información. Es en esta capa donde se establecen todas las reglas que deben cumplirse en la solución; se comunica con la capa de interfaz, para presentar los resultados y con la capa de Acceso a Datos, para solicitar la gestión de los datos (10).

✚ Capa de Acceso a Datos

Esta capa es la encargada de gestionar los datos del sistema, su función es almacenar y devolver datos a la capa de negocio (10).

Luego de realizado el análisis de cada una de las capas antes mencionadas se define la arquitectura de la solución en cuestión.

La solución consta de tres capas, una capa interfaz que es la que se encarga de interactuar con el usuario, la misma realiza una petición a la capa Solución. La capa Solución está compuesta por todas las funcionalidades con que consta la solución, la misma se encarga de realizar una petición de datos a la capa de Acceso a Datos y luego que estos le son devueltos da respuesta a la petición realizada por parte de la capa Interfaz. La capa de Acceso a Datos contiene un repositorio local donde se encuentran todas las dependencias que necesita el módulo Gserver para su correcto funcionamiento, además es la responsable de dar respuesta a la petición de datos realizada por la capa Solución.

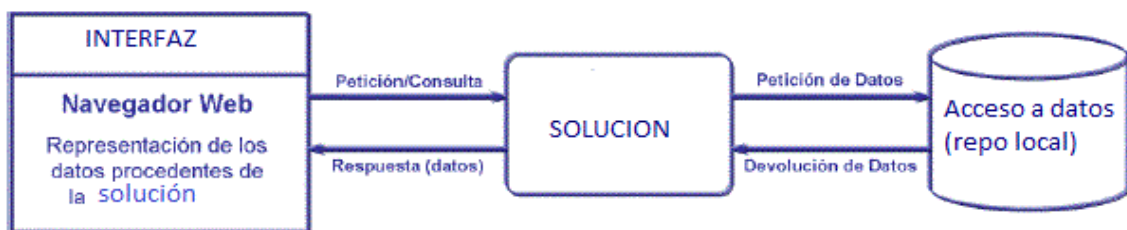


Figura 2: Arquitectura en Capas de la solución

3.2 Patrones de Diseño

Un patrón de diseño provee un esquema para refinar los subsistemas o componentes de un sistema de software, o las relaciones entre ellos, son el esqueleto de las soluciones a problemas comunes en el desarrollo de software. Brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares (19).

3.2.1 Patrones GRASP

GRASP es un acrónimo que significa General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades), representa parejas de problema solución con un nombre, que codifican buenos principios y sugerencias relacionados frecuentemente con la asignación de responsabilidades (15).

Dentro de los patrones GRASP utilizados en el desarrollo de la solución se encuentran los siguientes:

Experto: Indica que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo (15).

Las clases Base de Datos, uWSGI, Instalaciones, Configuraciones y permisos contarán con la información necesaria para cumplir cada una las responsabilidades que le corresponden. De este modo se obtiene un diseño con mayor cohesión y un bajo acoplamiento.

Creador: Guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. Una de sus ventajas es el bajo acoplamiento, lo cual supone facilidad de mantenimiento y reutilización. (15)

La clase GSERVER será la responsable de la creación de las instancias de Base de Datos, uWSGI, Instalaciones, configuraciones y permisos.

Bajo Acoplamiento: El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Acoplamiento bajo significa que una clase no depende de muchas clases. (15)

A las clases Base de Datos, uWSGI, Instalaciones, Configuraciones y permisos se les asignan responsabilidades de forma tal que solo se comuniquen con la clase GSERVER,

con el uso de este patrón se fortalece la reutilización de código y se disminuye la dependencia entre las clases.

3.2.2 Patrones GOF

Los Patrones GOF (Gang of Four, Banda de Cuatro) proporcionan a los programadores una estructura de código común a todos los proyectos que implemente una funcionalidad genérica. La utilización de estos patrones de diseño, permite ahorrar tiempo en la construcción del software, así como hacerlo más fácil de comprender, mantener y extender (16).

Durante el diseño del sistema se emplearon patrones GOF como:

Patrones creacionales

Método de fabricación: Centraliza en una clase constructora la creación de objetos de un subtipo de un tipo determinado, ocultando al usuario la diversidad de casos particulares que se pueden prever (10).

La clase Gserver es la responsable de crear todos los objetos de tipo Base de Datos, uWSGI, Instalaciones, configuraciones y permisos.

3.3 Tarjetas Clase – Responsabilidad – Colaborador

Las tarjetas CRC, propias de la metodología XP, son una técnica simple e informal pero efectiva que ha sido propuesta para el diseño detallado de sistemas Orientado a Objetos. Se analizan basándose en sus responsabilidades con respecto al sistema y permiten que el equipo completo contribuya en la tarea del diseño.

Una tarjeta CRC representa un objeto, el nombre de la clase se coloca a modo de título en la tarjeta, las responsabilidades se colocan a la izquierda, y las clases que se implican en cada responsabilidad a la derecha, en la misma línea que su requerimiento correspondiente.

Las responsabilidades son los atributos y las operaciones relevantes para la clase. Dicho de una manera más simple una responsabilidad es "cualquier cosa que la clase sabe o hace". Los colaboradores son aquellas clases que se requieren para que una clase reciba la información necesaria para completar una responsabilidad (21).

A continuación se muestra una de las tarjetas CRC confeccionadas, correspondiente a una de las clases más relevantes en el funcionamiento del sistema, el resto de las mismas se encuentran visibles en la sección Anexos II.

Tarjeta CRC

Tabla 11: uWSGI

Clase: uWSGI	
Descripción: Es la encargada de tener un control de los procesos en ejecución.	
Responsabilidad	Colaboración
Crea una aplicación uWSGI.	Gserver
Carga los procesos en ejecución en un momento determinado.	Gserver

3.4 Tareas de la Ingeniería

Las tareas de la ingeniería son escritas por el equipo de desarrollo a partir de las HU elaboradas por el cliente, brindando un detalle más profundo para realizar una implementación de las mismas y estimando un tiempo más cercano a la realidad para cada una de ellas.

Tabla 15: Muestra de una Tarea de Ingeniería.

Tarea de Ingeniería	
Número Tarea: (Número de la Tarea, incremental en el tiempo)	Número Historia de Usuario: (Número de la HU a la que pertenece la tarea)
Nombre Tarea: (Nombre que identifica a la tarea)	
Tipo de Tarea: (Las tareas pueden ser de: Desarrollo, Corrección, Mejora y Otra (especificar))	Puntos Estimados: (Tiempo en días que se le asignará al desarrollo de la tarea (estimado))
Fecha Inicio: (Fecha en que inicia el desarrollo de la tarea)	Fecha Fin: (Fecha en que finaliza el desarrollo de la tarea)
Programador Responsable: (Nombre y apellidos del programador)	
Descripción: (Breve descripción de la tarea)	

A continuación se exponen unas de las tareas de ingeniería correspondientes a las HU representadas anteriormente, el resto de las tareas de ingeniería definidas por el equipo de desarrollo se encuentran en la sección Anexos III.

Tabla 16: Instalar dependencias.

Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: 1
Nombre Tarea: Instalar dependencias.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 3/3
Fecha Inicio:- 1/02/2015	Fecha Fin:- 4/02/2015
Programador Responsable: Diego Miguel Ferreiro Garcia.	
Descripción: Instalar las dependencias desde los repositorios de sus respectivas distribuciones. Entre estas dependencias se encuentran: python-setuptools; django-bootstrap-form-master.zip; requests-1.2.0.tar.gz; URLObject-2.3.4.tar.gz; python-dnspython; python-dateutil; python-kombu.	

3.5 Pruebas

La metodología XP propone la constante realización de pruebas, los desarrolladores dirigen el trabajo a la búsqueda de errores, someten sistemáticamente a pruebas las funcionalidades del sistema lo que permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección.

XP divide las pruebas del sistema en dos grupos: pruebas unitarias, diseñada por los programadores con el objetivo de verificar el código y pruebas de aceptación destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñadas por el cliente final.

3.5.1 Pruebas unitarias

Las pruebas unitarias son una de las piedras angulares de XP. Todos los módulos deben de pasar las pruebas unitarias antes de ser liberados o publicados (18).

En el transcurso de la implementación del sistema cada uno de los desarrolladores prueba constantemente lo que va obteniendo para garantizar que las funcionalidades exigidas por el cliente estén siendo implementadas correctamente. Cuando se encuentra un error, éste debe ser corregido inmediatamente, y se deben tener precauciones para que errores similares no vuelvan a ocurrir (18).

Las pruebas unitarias no generan artefactos y no son directamente palpables para el cliente, las mismas se desarrollaron constantemente cada vez que se concluía la implementación de alguna funcionalidad probándola directamente en el sistema.

3.5.2 Pruebas de aceptación

Las pruebas de aceptación son creadas en base a las historias de usuarios. Una historia de usuario no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación (19). Estas pruebas se enfocan en las características generales y las funcionalidades del sistema, son las encargadas de comprobar que las funcionalidades desarrolladas sean las esperadas por el cliente. Luego de haber superado las pruebas de aceptación podrá considerarse que la aplicación es apta para el uso y despliegue dentro del proyecto.

Los diseños de casos de pruebas se encuentran en la sección Anexos IV, los mismos fueron divididos en las siguientes secciones para una mayor organización:

- Clases Válidas: describe cada uno de los pasos seguidos durante el desarrollo de la prueba, se tiene en cuenta cada una de las entradas válidas que hace el usuario con el objetivo de verificar si se obtiene el resultado esperado.
- Clases Inválidas: describe cada uno de los pasos seguidos durante el desarrollo de la prueba, se tiene en cuenta cada una de las posibles entradas inválidas que hace el usuario con el objetivo de verificar si se obtiene el resultado esperado y cómo responde el sistema.
- Resultado Esperado: se describe el resultado que se espera ya sea para entradas válidas o inválidas.
- Resultado de la Prueba: se describe el resultado que se obtiene.
- Observaciones: algún señalamiento o advertencia que sea necesario hacerle a la sección que se está probando.

Las pruebas de aceptación correspondientes a las funcionalidades de la Personalización de Ubuntu 12.04 para el módulo Gserver llevadas a cabo por el cliente arrojaron los siguientes resultados: en la primera iteración se obtienen un total de 5 no conformidades, de las cuales todas fueron resueltas, de esta forma no quedó ninguna pendiente por resolver, en la segunda iteración se encontraron 2 no conformidades. A continuación se ilustran los resultados en una gráfica:

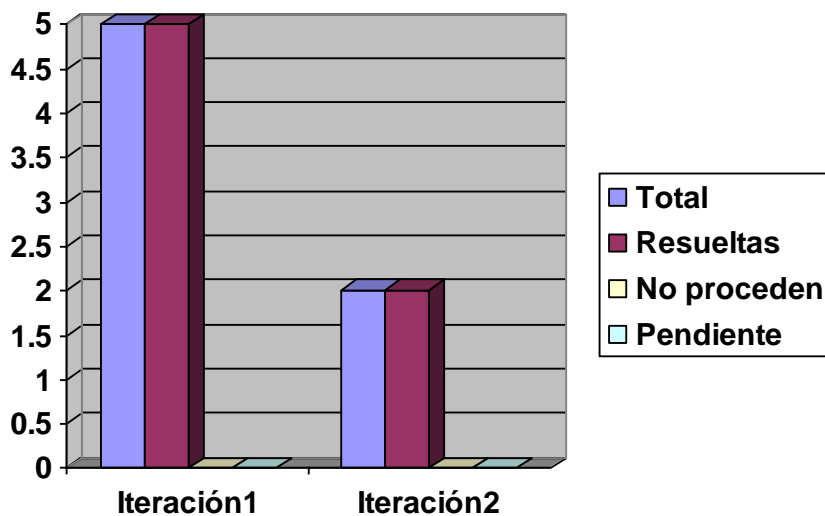


Figura 3: Tabla de solución de no conformidades

Conclusiones parciales

En el presente capítulo se definió la arquitectura del sistema, para así diseñar y guiar el proceso de desarrollo de la solución. Se tuvo en cuenta la utilización de patrones de diseño para lograr una mejor reutilización de código y una correcta implementación de la aplicación. Además se definieron las tarjetas CRC permitiendo identificar y organizar las clases orientadas a objetos. Se implementaron las Tareas de la Ingeniería definidas para cada HU y se llevaron a cabo las pruebas de aceptación logrando la satisfacción del cliente con el software desarrollado.

CONCLUSIONES

A través del desarrollo de la investigación se puede llegar a las siguientes conclusiones:

- Se definió los principales referentes teóricos - metodológicos que deben sustentar la gestión de la información en la realización de una distribución personalizada de Ubuntu Server 12.04 para el módulo Gserver.
- Se identificó las necesidades existentes en la realización de una solución que permita la integración del módulo Gserver a una personalización de Ubuntu Server 12.04 en la UCI.
- Se seleccionó un conjunto de herramientas y metodologías de desarrollo de software que facilitaron el desarrollo de la solución propuesta.
- Se desarrolló el análisis y diseño de la solución propuesta.
- Se desarrolló una distribución personalizada de Ubuntu Server 12.04 para el módulo Gserver.
- Se realizó un conjunto de pruebas a cada una de las funcionalidades desarrolladas para comprobar su correcto funcionamiento.

RECOMENDACIONES

- Realizar la solución para otras distribuciones como: Debian 7, Centos, Ubuntu 14.04.

REFERENCIA BIBLIOGRÁFICA

1. Ubuntu Server. <http://recursos.cepindalo.es>. [En línea] [Citado el: 16 de enero de 2015.]
<http://recursos.cepindalo.es/mod/book/tool/print/index.php?id=548&chapterid=152>.
2. Software. definicion.de. [En línea] [Citado el: 6 de enero de 2015.]
definicion.de/software/.
3. Sistema Operativo. definicion.de. [En línea] [Citado el: 16 de enero de 2015.]
<http://definicion.de/sistema-operativo/>.
4. Hardware. definicion.de. [En línea] [Citado el: 6 de enero de 2015.]
<http://definicion.de/hardware/>.
5. Martínez Faneyth, Luis Alejandro. [En línea] 10 de octubre de 2012. [Citado el: 6 de enero de 2015.] <http://canaima.softwarelibre.gob.ve/novedades/ayudan-os-a-probar-el-instalador-de-la-nueva-version-del-sistema-operativo-canaima-gnu-linux-3.1>.
6. Ubuntu. Ubuntu: software libre y de código abierto. [En línea] [Citado el: 7 de enero de 2015.] <https://soubuntublog.wordpress.com/listado-de-eventos-principales/>.
7. Ubuntu Desktop. Universidad Nacional Abierta y a Distancia. [En línea] [Citado el: 6 de enero de 2015.]
http://datateca.unad.edu.co/contenidos/201490/Exe_Lecturas_por_Tema/Act3_Reconocimiento_Tema1/Unidad1/caso_de_uso_conociendo_gnulinux_ubuntu.html.
8. Distribuciones basadas en Ubuntu. Software que incluye Ubuntu. [En línea] [Citado el: 7 de enero de 2015.]
<http://linoit.com/users/angelakane09/canvases/SOFTWARE%20QUE%20INCLUYE%20UBUNTU>.
9. Linux. <http://www.baquia.com>. [En línea] 23 de marzo de 2013. [Citado el: 4 de febrero de 2015.] <http://www.baquia.com/tecnologia-y-negocios/entry/emprendedores/2013-03-25-china-prepara-un-sistema-operativo-nacional-basado-en-ubuntu>.
10. Montero Vals, Adrián y Sospedra López, Diannet .
<http://semanatecnologica.fordes.co.cu>. [En línea] 9 de 2013. [Citado el: 6 de febrero de 2015.] http://semanatecnologica.fordes.co.cu/sites/default/files/public/p29_0.pdf.
11. Python. <http://e-logicasoftware.com/>. [En línea] [Citado el: 7 de febrero de 2015.]
grupolinux/directorios/acis/2015/folletos-2015/cursos-acis-pdf/LE-python-RN-lenguaje-de-programacion-python.pdf.
12. ImgBurn. <http://es.recuweb.co>. [En línea] [Citado el: 7 de febrero de 2015.]
<http://es.recuweb.co/xbox/imgburn>.
13. Muylinux. [En línea] 24 de abril de 2012. [Citado el: 15 de mayo de 2015.]
<http://www.muylinux.com/2012/04/24/nginx-1-2-0-estable-disponible>.
14. Cristalab. [En línea] enero de 2014. [Citado el: 15 de mayo de 2015.]
<http://www.cristalab.com/blog/7-razones-para-usar-nginx-web-server-c112338/>.
15. Caballero G., Leonardo J. Plone. [En línea] [Citado el: 15 de mayo de 2015.]
<http://plone-spanish-docs.readthedocs.org/es/latest/zope/wsgi/index.html>.

16. The GNU Bourne-Again SHell. [En línea] [Citado el: 16 de mayo de 2015.]
<http://tiswww.case.edu/php/chet/bash/bashtop.html>.
17. Cesareox. [En línea] [Citado el: 16 de marzo de 2015.]
<https://www.cesareox.com/docencia/dfs/el-lenguaje-bash>.
18. Diseño Arquitectonico. [En línea] [Citado el: 10 de febrero de 2015.]
<http://disenoarquitectonicousta.blogspot.com/2014/09/quien-lo-realiza-aunque-un-ingeniero-de.html>.
19. Hall, Larman - Prentice. UML y Patrones. Introducción al análisis y diseño orientado a objetos. págs. 189-196.
20. Rosanigo, Zulema Beatriz. Maximizando reuso en software para Ingeniería Estructural. 2000.
21. Modelado del analisis. <https://inge2software.files.wordpress.com>. [En línea] [Citado el: 15 de febrero de 2015.]
<https://inge2software.files.wordpress.com/2012/.../modelado-de-analisis.....>
22. Joskowicz, José . Reglas y Prácticas en XP. [En línea] 10 de febrero de 2008. [Citado el: 12 de febrero de 2015.]
<http://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>.
23. Virtual. <http://virtual.urbe.edu>. [En línea] [Citado el: 23 de marzo de 2015.]
[tesispub/0092784/cap03.pdf](http://virtual.urbe.edu/tesispub/0092784/cap03.pdf).

BIBLIOGRAFÍA

1. Ubuntu Server. *http://recursos.cepindalo.es*. [En línea] [Citado el: 16 de enero de 2015.]
<http://recursos.cepindalo.es/mod/book/tool/print/index.php?id=548&chapterid=152>.
2. Software. *definicion.de*. [En línea] [Citado el: 6 de enero de 2015.]
definicion.de/software/.
3. Sistema Operativo. *definicion.de*. [En línea] [Citado el: 16 de enero de 2015.]
<http://definicion.de/sistema-operativo/>.
4. Hardware. *definicion.de*. [En línea] [Citado el: 6 de enero de 2015.]
<http://definicion.de/hardware/>.
5. Martínez Faneyth, Luis Alejandro. [En línea] 10 de octubre de 2012. [Citado el: 6 de enero de 2015.] <http://canaima.softwarelibre.gob.ve/novedades/ayudanos-a-probar-el-instalador-de-la-nueva-version-del-sistema-operativo-canaima-gnu-linux-3.1>.
6. Ubuntu. *Ubuntu: software libre y de código abierto*. [En línea] [Citado el: 7 de enero de 2015.] <https://soubuntublog.wordpress.com/listado-de-eventos-principales/>.
7. Ubuntu Desktop. *Universidad Nacional Abierta y a Distancia*. [En línea] [Citado el: 6 de enero de 2015.]
http://datateca.unad.edu.co/contenidos/201490/Exe_Lecturas_por_Tema/Act3_Reconocimiento_Tema1/Unidad1/caso_de_uso_conociendo_gnulinix_ubuntu.html.
8. Distribuciones basadas en Ubuntu. *Software que incluye Ubuntu*. [En línea] [Citado el: 7 de enero de 2015.]
<http://linoit.com/users/angelakane09/canvases/SOFTWARE%20QUE%20INCLUYE%20UBUNTU>.
9. Linux. <http://www.baquia.com>. [En línea] 23 de marzo de 2013. [Citado el: 4 de febrero de 2015.] <http://www.baquia.com/tecnologia-y-negocios/entry/emprendedores/2013-03-25-china-prepara-un-sistema-operativo-nacional-basado-en-ubuntu>.
10. Montero Vals, Adrián y Sospedra López, Diannet .
<http://semanatecnologica.fordes.co.cu>. [En línea] 9 de 2013. [Citado el: 6 de febrero de 2015.] http://semanatecnologica.fordes.co.cu/sites/default/files/public/p29_0.pdf.
11. Python. <http://e-logicasoftware.com/>. [En línea] [Citado el: 7 de febrero de 2015.]
grupolinux/directorios/acis/2015/folletos-2015/cursos-acis-pdf/LE-python-RN-lenguaje-de-programacion-python.pdf.
12. ImgBurn. <http://es.recuweb.co>. [En línea] [Citado el: 7 de febrero de 2015.]
<http://es.recuweb.co/xbox/imgburn>.
13. *Muylinux*. [En línea] 24 de abril de 2012. [Citado el: 15 de mayo de 2015.]
<http://www.muylinux.com/2012/04/24/nginx-1-2-0-estable-disponible>.
14. *Cristalab*. [En línea] enero de 2014. [Citado el: 15 de mayo de 2015.]
<http://www.cristalab.com/blog/7-razones-para-usar-nginx-web-server-c112338/>.
15. Caballero G., Leonardo J. Plone. [En línea] [Citado el: 15 de mayo de 2015.]
<http://plone-spanish-docs.readthedocs.org/es/latest/zope/wsgi/index.html>.

16. *The GNU Bourne-Again SHell*. [En línea] [Citado el: 16 de mayo de 2015.] <http://tiswww.case.edu/php/chet/bash/bashtop.html>.
17. *Cesareox*. [En línea] [Citado el: 16 de marzo de 2015.] <https://www.cesareox.com/docencia/dfs/el-lenguaje-bash>.
18. *Diseño Arquitectónico*. [En línea] [Citado el: 10 de febrero de 2015.] <http://disenoarquitectonicousta.blogspot.com/2014/09/quien-lo-realiza-aunque-un-ingeniero-de.html>.
19. Hall, Larman - Prentice. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. págs. 189-196.
20. Rosanigo, Zulema Beatriz. *Maximizando reuso en software para Ingeniería Estructural*. 2000.
21. *Modelado del análisis*. <https://inge2software.files.wordpress.com>. [En línea] [Citado el: 15 de febrero de 2015.] <https://inge2software.files.wordpress.com/2012/.../modelado-de-analisis.....>
22. Joskowicz, José . *Reglas y Prácticas en XP*. [En línea] 10 de febrero de 2008. [Citado el: 12 de febrero de 2015.] <http://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>.
23. *Virtual*. <http://virtual.urbe.edu>. [En línea] [Citado el: 23 de marzo de 2015.] <http://virtual.urbe.edu/tesispub/0092784/cap03.pdf>.
24. Softonic Internacional S.A. *canaima.softonic*. [En línea] [Citado el: 16 de enero de 2015.] <http://canaima.softonic.com/linux>.
25. BBC. *fayerwayer*. *fayerwayer*. [En línea] 22 de marzo de 2013. <https://www.fayerwayer.com/2013/03/china-y-canonical-lanzaran-ubuntu-kylin-un-sistema-operativo-disenado-para-ese-pais/>.
26. *Home of the Patterns Library*. [En línea] 2010. [Citado el: 14 de febrero de 2015.] [http://hillside.net/..](http://hillside.net/)
27. *Metodologías de Desarrollo de Software*. <http://www.academia.edu>. <http://www.academia.edu>. [En línea] [Citado el: 2 de febrero de 2015.] http://www.academia.edu/4984909/Metodologia_de_desarrollo_de_software.
28. LIGHTNING UK. *imgburn*. [En línea] 16 de junio de 2013. [Citado el: 5 de febrero de 2015.] <http://www.imgburn.com>.
29. Universidad Técnica Federico Santa María. *inf.utfsm.cl*. *inf.utfsm.cl*. [En línea] 9 de abril de 2013. <http://www.inf.utfsm.cl/~visconti/ili236>.
30. Universidad de Valladolid. *infor.uva*. *infor.uva*. [En línea] 23 de mayo de 2013. . Félix Prieto, Universidad de Valladolid. [En línea] 2009. [Citado el: mayo de 23 de 2013.] <http://www.infor.uva.es>.
31. *masadelante.com*. *masadelante.com*. [En línea] s.f. <https://www.masadelante.com/faqs/software-hardware>.
32. Microsoft Developer Network. ¿Qué es un Patrón? <http://msdn.microsoft.com>. [En línea] [Citado el: 1 de abril de 2015.] <http://msdn.microsoft.com/es-es/library/bb972240.aspx..>

33. SoftPedia. ochobitshacenunbyte. *ochobitshacenunbyte*. [En línea] 4 de febrero de 2015. <http://www.ochobitshacenunbyte.com>.
34. Python Software Foundation. *python*. [En línea] 10 de diciembre de 2014. [Citado el: 3 de febrero de 2015.] <https://www.python.org/>.
35. Creative Commons Attribution-NonCommercial-ShareAlike. *recursos.cepindalo*. [En línea] [Citado el: 2 de febrero de 2015.] <http://recursos.cepindalo.es/mod/book/tool/print/index.php?id=548#ch154>.
36. Creative Commons. *um*. *um*. [En línea] 30 de diciembre de 2014. <http://www.um.es/docencia/barzana/IAGP/lagp2.html>.
37. *visual-paradigm*. [En línea] 3 de febrero de 2015. [Citado el: 5 de febrero de 2015.] <http://www.visual-paradigm.com/>.
38. Brodtkin, Jon. *arstechnica*. *arstechnica*. [En línea] 22 de marzo de 2013. http://arstechnica.com/information-technology/2013/03/goodbye-windows-china-to-create-home-grown-os-based-on-ubuntu/?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%3A+arstechnica%2Findex+%28Ars+Technica+-+All+content%29.
39. Cendejas Valdéz, José Luis. *http://www.eumed.net*. [En línea] 2014. [Citado el: 2 de febrero de 2015.] <http://www.eumed.net/tesis-doctorales/2014/jlcv/software.htm>.
40. Pressman, Roger S. Ingeniería del software. Un enfoque práctico. Capítulo 8: Modelado del Análisis. *Ingeniería del software*. s.l. : 6ta Edición 2006. 2006., 2006.
41. *hipertextual*. [En línea] [Citado el: 12 de febrero de 2015.] <http://hipertextual.com/archivo/2014/04/comandos-basicos-terminal/>.
42. *StackExchange*. [En línea] [Citado el: 15 de abril de 2015.] <http://askubuntu.com/questions/409607/how-to-create-a-customized-ubuntu-server-iso>.
43. *B.5. Advanced options*. [En línea] [Citado el: 10 de enero de 2015.] <https://www.debian.org/releases/stable/mips/apbs05.html.en>.
44. *Askubuntu*. [En línea] [Citado el: 12 de enero de 2015.] <http://askubuntu.com/questions/122505/how-do-i-create-a-completely-unattended-install-of-ubuntu>.
45. *Ubuntu-es*. [En línea] 6 de junio de 2008. [Citado el: 6 de enero de 2015.] <http://www.ubuntu-es.org/node/155912>.
46. DesdeLinux. [En línea] [Citado el: 16 de febrero de 2015.] <http://blog.desdelinux.net/como-construir-una-version-personalizada-de-ubuntu/>.
47. SF-Xpt's. *SF-Xpt's*. [En línea] 9 de junio de 2013. [Citado el: 2 de febrero de 2015.] <https://sfxpt.wordpress.com/2013/06/09/get-the-debianubuntu-ready-and-customized-the-way-you-like-in-10-minutes/>.
48. One kind of Life. [En línea] [Citado el: 12 de marzo de 2015.] <https://doutdex.wordpress.com/2008/04/28/comprimir-y-descomprimir-rargzip2-lha-zoo-en-linux-comando-consola/>.
49. López Asaña, Daniel. *Los 15 comandos Linux más útiles para mantener sistemas de ficheros*. [En línea] [Citado el: 26 de marzo de 2015.]

<http://www.daniloaz.com/es/los-comandos-linux-mas-utiles-para-mantener-sistemas-de-ficheros/>.

ANEXOS

ANEXOS I:

Tabla 3: HU # 2 Instalación y Configuración.

Historia de Usuario	
Número: 2	Nombre de Historia de Usuario: Instalación y Configuración.
Modificación de Historia de Usuario Número: Ninguna.	
Usuario: Diego Miguel Ferreiro Garcia, Julio Cesar Quesada Arias.	Iteración Asignada: 1
Prioridad en negocio: Alta.	Puntos estimados:2
Riesgo en Desarrollo: Alto.	Puntos Reales: 2
Descripción: Se le facilita el trabajo al usuario tanto de almacenamiento cómo de obtención de información, pues precisamente se habilita el Sistema Gestor de Bases Datos.	

Tabla 4: HU # 3 Creación de directorios, trazas y asignación de permisos.

Historia de Usuario	
Número: 3	Nombre de Historia de Usuario: Creación de directorios, trazas y asignación de permisos.
Modificación de Historia de Usuario Número: Ninguna.	
Usuario: Diego Miguel Ferreiro Garcia, Julio Cesar Quesada Arias.	Iteración Asignada: 2
Prioridad en negocio: Alta.	Puntos estimados:2
Riesgo en Desarrollo: Alto.	Puntos Reales: 2
Descripción: Permitir al usuario tener un control de las acciones realizadas en la PC según los permisos asignados, apoyándose en los directorios de almacenamiento y trazas, así como en la asignación de permisos anteriormente realizada.	

Tabla 5: HU # 4 Enrutamiento y configuración de permisos.

Historia de Usuario	
Número: 4	Nombre de Historia de Usuario: Enrutamiento y configuración de permisos.
Modificación de Historia de Usuario Número: Ninguna.	
Usuario: Diego Miguel Ferreiro Garcia, Julio Cesar Quesada Arias.	Iteración Asignada: 1
Prioridad en negocio: Alta.	Puntos estimados:2
Riesgo en Desarrollo: Alto.	Puntos Reales: 2
Descripción: Permitirá al usuario realizar diversas acciones en dependencia del rol que desempeña en el proyecto, basándose en el enrutamiento de la carpeta de instalación de Gserver y configuración de permisos a los usuarios anteriormente realizado.	

Tabla 6: HU # 5 Configuración y sincronización.

Historia de Usuario	
Número: 5	Nombre de Historia de Usuario: Configuración y sincronización.
Modificación de Historia de Usuario Número: Ninguna.	
Usuario: Diego Miguel Ferreiro Garcia, Julio Cesar Quesada Arias.	Iteración Asignada: 2
Prioridad en negocio: Alta.	Puntos estimados:1
Riesgo en Desarrollo: Alto.	Puntos Reales: 1
Descripción: Permitir una interacción adecuada por parte del usuario con la base de datos (BD), apoyados en la correcta configuración de los parámetros de conexión a la BD y sincronización de la misma.	

Tabla 7: HU # 6 Creación y configuración de la aplicación de uWSGI.

Historia de Usuario	
Número: 6	Nombre de Historia de Usuario: Creación de la aplicación de uWSGI y configuración de su archivo.
Modificación de Historia de Usuario Número: Ninguna.	
Usuario: Diego Miguel Ferreiro Garcia, Julio Cesar Quesada Arias.	Iteración Asignada: 2
Prioridad en negocio: Alta.	Puntos estimados: 2
Riesgo en Desarrollo: Alto.	Puntos Reales: 2
Descripción: Permite al usuario tener un mayor control de la cantidad de procesos en ejecución en un momento determinado basándose en la creación de la aplicación de uWSGI y configuración de su archivo.	

Tabla 8: HU # 7 Realizar enlaces y eliminar aplicaciones.

Historia de Usuario	
Número: 7	Nombre de Historia de Usuario: Realizar enlaces y eliminar aplicaciones.
Modificación de Historia de Usuario Número: Ninguna.	
Usuario: Diego Miguel Ferreiro Garcia, Julio Cesar Quesada Arias.	Iteración Asignada: 3
Prioridad en negocio: Alta.	Puntos estimados: 2
Riesgo en Desarrollo: Alto.	Puntos Reales: 2
Descripción: Permite al usuario un mejor interacción con el sistema a través de del servidor web nginx. Además permite hacer uso del enlace simbólico con las aplicaciones habilitadas y trabajar sin la aplicación predeterminada de la instalación de nginx después de su eliminación.	

Tabla 9: HU # 8 Configuración de la conexión SSL.

Historia de Usuario	
Número: 8	Nombre de Historia de Usuario: Configuración de la conexión SSL.
Modificación de Historia de Usuario Número: Ninguna.	
Usuario: Diego Miguel Ferreiro Garcia, Julio Cesar Quesada Arias.	Iteración Asignada: -
Prioridad en negocio: Alta.	Puntos estimados:1
Riesgo en Desarrollo: Alto.	Puntos Reales: 1
Descripción: Permite al usuario tener una mayor seguridad, integridad y autenticidad de la información pues las conexiones SSL garantizan el establecimiento de comunicaciones seguras.	

Tabla 10: HU # 9 Creación y configuración de una nueva aplicación de nginx.

Historia de Usuario	
Número: 9	Nombre de Historia de Usuario: Creación y configuración de una nueva aplicación de nginx.
Modificación de Historia de Usuario Número: Ninguna.	
Usuario: Diego Miguel Ferreiro Garcia, Julio Cesar Quesada Arias.	Iteración Asignada: -
Prioridad en negocio: Alta.	Puntos estimados:1
Riesgo en Desarrollo: Alto.	Puntos Reales: 1
Descripción: Facilita al usuario el trabajo con el sistema de forma sencilla e interactiva, pues es precisamente nginx quien brinda una interfaz gráfica al usuario.	

ANEXOS II

Tarjetas CRC

Tabla 12: BasedeDatos

Clase: BasedeDatos	
Descripción: Es la encargada de manejar todo el trabajo de almacenamiento.	
Responsabilidad	Colaboración
Crea una base de datos vacía	Gserver

Tabla 13: ConfiguraciónPermisos

Clase: ConfiguraciónPermisos	
Descripción: Es la encargada de manejar todos los eventos relaciones con las configuraciones y permisos.	
Responsabilidad	Colaboración
Recoge los parámetros necesarios para la conexión y configuración de la Base de Datos.	Gserver
Configura la aplicación uWSGI para su buen funcionamiento.	Gserver

Tabla 14: Instalaciones

Clase: Instalaciones	
Descripción: Es la encargada de manejar todos los eventos relaciones con las instalaciones.	
Responsabilidad	Colaboración
Instala el Sistema Gestor de Bases de Datos postgresQL.	Gserver
Instala todas las dependencias y paquetes correspondientes.	Gserver

ANEXOS III

Tarea # 2 Instalar paquetes.

Tarea de Ingeniería	
Número Tarea: 2	Número Historia de Usuario: 1
Nombre Tarea: Instalar paquetes.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 3/3
Fecha Inicio: - 4/02/2015	Fecha Fin: - 7/02/2015
Programador Responsable: Julio César quesada Arias, Diego Miguel Ferreiro Garcia.	
Descripción: Se realiza la instalación de paquetes necesarios que permitan la correcta instalación de Gserver.	

Tarea # 3 Instalación del Sistema Gestor de Bases Datos (SGBD).

Tarea de Ingeniería	
Número Tarea: 3	Número Historia de Usuario: 2
Nombre Tarea: Instalación del Sistema Gestor de Bases Datos (SGBD).	
Tipo de Tarea: Desarrollo	Puntos Estimados: 3/3
Fecha Inicio: - 7/02/2015	Fecha Fin: - 11/02/2015
Programador Responsable: Julio César quesada Arias, Diego Miguel Ferreiro Garcia.	
Descripción: Se le facilita el trabajo tanto de almacenamiento como de obtención de información. Se realiza utilizando el comando apt-get install postgresql.	

Tarea # 4 Configuración del Sistema Gestor de Bases Datos (SGBD).

Tarea de Ingeniería	
Número Tarea: 4	Número Historia de Usuario: 2
Nombre Tarea: Configuración del Sistema Gestor de Bases Datos (SGBD).	
Tipo de Tarea: Desarrollo	Puntos Estimados: 3/3
Fecha Inicio: - 15/02/2015	Fecha Fin: - 18/02/2015
Programador Responsable: Julio César quesada Arias, Diego Miguel Ferreiro Garcia.	
Descripción: Se garantiza el correcto funcionamiento del anterior instalado SGBD. Para ello es necesario insertar ciertos datos como son, 'NAME': 'NOMBREBD', 'USER': 'USUARIO', 'PASSWORD': 'PASSWORD', 'HOST': 'SERVIDORBD' y 'PORT': 'PUERTO'.	

Tarea # 5 Enrutamiento de la carpeta de instalación de Gserver

Tarea de Ingeniería	
Número Tarea: 5	Número Historia de Usuario: 3
Nombre Tarea: Enrutamiento de la carpeta de instalación de Gserver.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 4/4
Fecha Inicio: - 5/03/2015	Fecha Fin: - 9/03/2015
Programador Responsable: Diego Miguel Ferreiro Garcia, Julio César quesada Arias.	
Descripción: Se garantizara dar una dirección a dicha carpeta de enrutamiento haciendo uso del comando <code>cp -R carpetaGServer /usr/local/gserver.</code>	

Tarea # 6 Configuración de permisos a los usuarios.

Tarea de Ingeniería	
Número Tarea: 6	Número Historia de Usuario: 3
Nombre Tarea: Configuración de permisos a los usuarios.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 4/4
Fecha Inicio: - 10/03/2015	Fecha Fin: - 14/03/2015
Programador Responsable: Diego Miguel Ferreiro Garcia, Julio César quesada Arias.	
Descripción: Permitirá al usuario realizar diversas acciones en dependencia del rol que desempeña en el proyecto haciendo uso el comando <code>chown -R www-data:www-data /usr/local/gserver/</code> .	

Tarea # 7 Creación de directorios de almacenamiento de archivos estáticos y trazas.

Tarea de Ingeniería	
Número Tarea: 7	Número Historia de Usuario: 4
Nombre Tarea: Creación de directorios de almacenamiento de archivos estáticos y trazas.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 6/6
Fecha Inicio: - 15/03/2015	Fecha Fin: - 21/03/2015
Programador Responsable: Diego Miguel Ferreiro Garcia, Julio César quesada Arias.	
Descripción: Garantizará tener el control de todo lo que ocurra en cada PC utilizando los comandos <code>mkdir -p /var/gserver/media/</code> , <code># mkdir -p /var/gserver/static/</code> y <code># chown -R www-data:www-data /var/gserver/media/ /var/gserver/static/</code> .	

Tarea # 8 Asignación de permisos y movimiento de archivos estáticos.

Tarea de Ingeniería	
Número Tarea: 8	Número Historia de Usuario: 4
Nombre Tarea: Asignación de permisos y movimiento de archivos estáticos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 6/6
Fecha Inicio: - 22/03/2015	Fecha Fin: - 28/03/2015
Programador Responsable: Diego Miguel Ferreiro Garcia, Julio César Quesada Arias.	
Descripción: Permitir tener un control de las acciones realizadas en la PC según los permisos asignados utilizando los comandos : mkdir /var/log/gserver/ chown -R www-data:www-data /var/log/gserver/	

Tarea # 9 Sincronización de la BD.

Tarea de Ingeniería	
Número Tarea: 9	Número Historia de Usuario: 5
Nombre Tarea: Sincronización de la BD.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 3/3
Fecha Inicio: - 2/04/2015	Fecha Fin: - 5/04/2015
Programador Responsable: Julio César quesada Arias, Diego Miguel Ferreiro Garcia.	
Descripción: Garantizará el correcto uso de la BD, facilitando así el uso de la misma. Se realiza usando el comando python manage.py syncdb.	

Tarea # 10 Creación de la aplicación de uWSGI.

Tarea de Ingeniería	
Número Tarea: 10	Número Historia de Usuario: 6
Nombre Tarea: Creación de la aplicación de uWSGI.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 3/3
Fecha Inicio: - 5/04/2015	Fecha Fin: - 8/04/2015
Programador Responsable: Julio César quesada Arias, Diego Miguel Ferreiro Garcia.	
Descripción: Garantizará tener un mayor control de la cantidad de procesos en ejecución en un momento determinado usando el comando touch /etc/uwsgi/apps-available/gserver.ini y así garantizar lo antes mencionado.	

Tarea # 12 Configuración del archivo de uWSGI.

Tarea de Ingeniería	
Número Tarea: 12	Número Historia de Usuario: 6
Nombre Tarea: Configuración de su archivo.	
Tipo de Tarea: Desarrollo	Puntos Estimados:2/3
Fecha Inicio: - 8/04/2015	Fecha Fin: - 10/04/2015
Programador Responsable: Julio César quesada Arias, Diego Miguel Ferreiro Garcia.	
Descripción: Garantizar que tenga un correcto funcionamiento el uWSGI. S garantizará que el usuari tenga una interfaz gráfica que le permitirá interactuar con el módulo Gserver como tal.	

Tarea # 12 Realizar enlace simbólico con las aplicaciones habilitadas y eliminar la aplicación predeterminada de la instalación de nginx.

Tarea de Ingeniería	
Número Tarea: 12	Número Historia de Usuario: 7
Nombre Tarea: Realizar enlace simbólico con las aplicaciones habilitadas y eliminar la aplicación predeterminada de la instalación de nginx.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 7/7
Fecha Inicio: - 10/04/2015	Fecha Fin: -13/04/2015
Programador Responsable: Julio César quesada Arias, Diego Miguel Ferreiro Garcia.	
Descripción: Garantizará una mejor interacción con la solución a través del servidor web nginx. Se realizara haciendo uso del comando <code>ln -s /etc/uwsgi/apps-available/gserver.ini /etc/uwsgi/apps-enabled/gserver.ini</code>	

Tarea # 13 Configuración de la conexión SSL.

Tarea de Ingeniería	
Número Tarea: 14	Número Historia de Usuario: 8
Nombre Tarea: Configuración de la conexión SSL.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 7/7
Fecha Inicio: - 10/04/2015	Fecha Fin: -14/04/2015
Programador Responsable: Julio César quesada Arias, Diego Miguel Ferreiro Garcia.	
Descripción: Garantizará el correcto funcionamiento de la conexión y así tener una mayor seguridad, integridad y autenticidad de la información. Se realizara en varios pasos que son : - Crear una llave privada SSL: # openssl genrsa -des3 -out self-ssl.key 2048 - Crear una petición de certificado firmado: # openssl req -new -key self-ssl.key -out self-ssl.csr - Crear el certificado: # openssl x509 -req -days 365 -in self-ssl.csr -signkey self-ssl.key -out self-ssl.crt	

Tarea # 15 Creación de una nueva aplicación de nginx.

Tarea de Ingeniería	
Número Tarea: 15	Número Historia de Usuario: 9
Nombre Tarea: Creación de una nueva aplicación de nginx.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 7/7
Fecha Inicio: -13/04/2015	Fecha Fin: -16/04/2015
Programador Responsable: Diego Miguel Ferreiro Garcia, Julio César quesada Arias.	
Descripción: Garantizará un trabajo pleno con el servidor web. Se garantizará lo antes mencionado haciendo uso del comando touch /etc/nginx/sites-available/gserver	

Tarea # 16 Configuración de la nueva aplicación de nginx.

Tarea de Ingeniería	
Número Tarea: 16	Número Historia de Usuario: 9
Nombre Tarea: Configuración de la nueva aplicación de nginx.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 7/7
Fecha Inicio: -16/04/2015	Fecha Fin: -20/04/2015
Programador Responsable: Diego Miguel Ferreiro Garcia, Julio César Quesada Arias.	
Descripción: Garantizará el correcto funcionamiento de la nueva aplicación antes creada de nginx. Lo antes mencionado se garantizará usando asignando valores específicos como por ejemplo <code>ssl_certificate /etc/nginx/self-ssl.crt</code> , <code>ssl_certificate_key /etc/nginx/self-ssl.key</code> etc.	

ANEXOS IV

Caso de prueba de Aceptación	
Código: HU1-P1	Historia de Usuario: Instalar dependencias y paquetes.
Nombre de la persona que realiza la prueba: Diego Miguel Ferreiro Garcia	
Descripción: Se comprueba si la personalización instala todas las dependencias y paquetes necesarios para el funcionamiento de Gserver.	
Condiciones de ejecución: Comenzar la instalación de la personalización.	
Entrada/Condiciones de ejecución:	
Resultado esperado: El cliente pueda hacer uso de todas las dependencias anteriormente instaladas como php5.py.	
Evaluación de la Prueba: Satisfactoria.	

Caso de prueba de Aceptación	
Código: HU2-P2	Historia de Usuario: Instalación y Configuración.
Nombre de la persona que realiza la prueba: Julio Cesar Quesada Arias.	
Descripción: Se le facilita el trabajo al usuario tanto de almacenamiento cómo de obtención de información, pues precisamente se habilita el Sistema Gestor de Bases Datos.	
Condiciones de ejecución: La previa instalación de las dependencias necesarias.	
Entrada/Condiciones de ejecución:	
Resultado esperado: El cliente puede comenzar a tratar la información ya sea almacenarla u obtenerla.	
Evaluación de la Prueba: Satisfactoria.	

Caso de prueba de Aceptación	
Código: HU3-P3	Historia de Usuario Enrutamiento y configuración de permisos.
Nombre de la persona que realiza la prueba: Julio Cesar Quesada Arias.	
Descripción: Permitirá al usuario realizar diversas acciones en dependencia del rol que desempeña en el proyecto, basándose en el enrutamiento de la carpeta de instalación de Gserver y configuración de permisos a los usuarios anteriormente realizado.	
Condiciones de ejecución: La asignación previa de los permisos a los usuarios.	
Entrada/Condiciones de ejecución:	
Resultado esperado: Poder controlar las acciones realizadas por los usuarios, haciendo uso de la carpeta de instalación de Gserver.	
Evaluación de la Prueba: Satisfactoria.	

Caso de prueba de Aceptación	
Código: HU4-P4	Historia de Usuario Configuración y sincronización.
Nombre de la persona que realiza la prueba: Diego Miguel Ferreiro Garcia	
Descripción: Permitir una interacción adecuada por parte del usuario con la base de datos (BD), apoyados en la correcta configuración de los parámetros de conexión a la BD y sincronización de la misma.	
Condiciones de ejecución: La creación previa de la BD.	
Entrada/Condiciones de ejecución:	
Resultado esperado: Interacción por parte del usuario con la BD.	
Evaluación de la Prueba: Satisfactoria.	

Caso de prueba de Aceptación

Código:HU5-P5 **Historia de Usuario** Creación de la aplicación de uWSGI y configuración de su archivo.

Nombre de la persona que realiza la prueba: Diego Miguel Ferreiro Garcia.

Descripción: Permite al usuario tener un mayor control de la cantidad de procesos en ejecución en un momento determinado basándose en la creación de la aplicación de uWSGI y configuración de su archivo.

Condiciones de ejecución: Creación de la aplicación uWSGI.

Entrada/Condiciones de ejecución:

Resultado esperado: Controlar la cantidad de procesos en ejecución.

Evaluación de la Prueba: Satisfactoria.

Código:HU6-P6 **Historia de Usuario** Realizar enlaces y eliminar aplicaciones.

Nombre de la persona que realiza la prueba: Julio Cesar Quesada Arias.

Descripción: Permite al usuario un mejor interacción con el sistema a través de del servidor web nginx. Además permite hacer uso del enlace simbólico con las aplicaciones habilitadas y trabajar sin la aplicación predeterminada de la instalación de nginx después de su eliminación.

Condiciones de ejecución: Creación de la aplicación uWSGI.

Entrada/Condiciones de ejecución:

Resultado esperado: Interacción por parte del cliente con el sistema.

Evaluación de la Prueba: Satisfactoria.

Código:HU7-P7 **Historia de Usuario** Configuración de la conexión SSL.

Nombre de la persona que realiza la prueba: Julio Cesar Quesada Arias.

Descripción: Permite al usuario tener una mayor seguridad, integridad y autenticidad de la información.

Condiciones de ejecución: Instalación previa del servidor web nginx.

Entrada/Condiciones de ejecución: Certificados SSL.

Resultado esperado: Seguridad de la información para el cliente.

Evaluación de la Prueba: Satisfactoria.

Código: HU8-P8	Historia de Usuario Creación y configuración de una nueva aplicación de nginx.
Nombre de la persona que realiza la prueba: Diego Miguel Ferreiro Garcia.	
Descripción: Facilita al usuario el trabajo con el sistema de forma sencilla e interactiva, pues es precisamente nginx quien brinda una interfaz gráfica al usuario.	
Condiciones de ejecución: Haber eliminado la aplicación predeterminada.	
Entrada/Condiciones de ejecución	
Resultado esperado: Obtener una interfaz gráfica para el cliente.	
Evaluación de la Prueba: Satisfactoria.	