



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS  
FACULTAD 2

# PERSONALIZACIÓN DE NOVA SERVER 2015 PARA EL SISTEMA PLATSI

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO  
DE  
INGENIERO EN CIENCIAS INFORMÁTICAS

**Autores:** Yirai Pérez Suárez  
                  Jiubel Herrera Ymas  
**Tutores:** Ing. Yor Alex Remond Recio  
                  Ing. Luis Eduardo Gallardo Concepción  
**Consultante:** Ing. Gustavo Quesada Arévalo

La Habana, Junio 2015  
"Año 57 de la Revolución"

## **Declaración de autoría**

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

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

**Firma del Autor**

Yirai Pérez Suárez

---

**Firma del Autor**

Jiubel Herrera Ymas

---

**Firma del Tutor**

Ing. Luís E. Gallardo Concepción

---

**Firma del Tutor**

Ing. Yor Alex Remond Recio

## DATOS DE CONTACTO

**Datos del Autor:**

Yirai Pérez Suárez  
Universidad de las Ciencias Informáticas, La Habana, Cuba.  
Correo: [ypsuares@estudiantes.uci.cu](mailto:ypsuares@estudiantes.uci.cu)

**Datos del Autor:**

Jiubel Herrera Ymas  
Universidad de las Ciencias Informáticas, La Habana, Cuba.  
Correo: [jimas@estudiantes.uci.cu](mailto:jimas@estudiantes.uci.cu)

**Datos del Tutor:**

Ing. Luís Eduardo Gallardo Concepción, pertenece al área de TLM, Dpto. Aplicaciones.  
Universidad de las Ciencias Informáticas, La Habana, Cuba.  
Correo: [legallardo@uci.cu](mailto:legallardo@uci.cu)

**Datos del Cotutor:**

Ing. Yor Alex Remond Recio, pertenece al área de TLM, Dpto. Telecomunicaciones.  
Universidad de las Ciencias Informáticas, La Habana, Cuba.  
Correo: [reymond@uci.cu](mailto:reymond@uci.cu)

## **AGRADECIMIENTOS**

### **Yirai Pérez Suárez**

A toda mi familia en especial a mi madre, a mis tías Elda y Onelia, a mi prima Leansy y mis padres de acogida: Pedro y Edelena. A mi compañero de Tesis Jiubel. A mis tutores Yor Alex y Luís Gallardo y profesores consultantes Gustavo Quesada Arévalo y a Ramon A. Anglada. A mis amigos y amistades de la universidad, en especial a Zaillet, Yisel, Julio, Iraldo y Yurina y a sus padres María Victoria y José. A todos los que han tenido que ver conmigo de una forma o de otra, muchas gracias por su apoyo...

## **DEDICATORIA**

**Yirai Pérez Suárez**

Dedico esta tesis a toda mi familia, en especial a mi madre, a mis tías: Elda y Onelia por ayudarme a construir mi futuro y creer en mí en todo momento y a mi prima Leansy, que me ha apoyado y se ha preocupado tanto por mí en estos años.

## **RESUMEN**

Con el fin de automatizar el proceso realizado por los especialistas de seguridad informática para las revisiones de seguridad, en la Universidad de las Ciencias Informáticas en el centro de Telemática (TLM) se desarrolla la Plataforma de Seguridad en Tecnologías de la Información (PlatSI) la cual está integrada por varias herramientas de prueba de seguridad que permiten la realización de auditorías a Aplicaciones Web. La instalación de PlatSI se convierte en un proceso engorroso que implica un gran esfuerzo debido a la instalación manual de un número elevado de dependencias con un orden específico dependiendo de un repositorio. También se deben modificar varios archivos de configuración, en PlatSI (aplicación web), en Bambu-Server y en Bambu-, el objetivo de esta modificación es establecer los parámetros de red necesarios para la comunicación entre estos 3 componentes y la base de datos. De ahí que el presente trabajo defina como problema a resolver: ¿Cómo incorporar los componentes del proceso de instalación del sistema PlatSI a una distribución GNU/Linux? Como objetivo general: desarrollar una personalización de la distribución NOVA Server 2015 para el sistema PlatSI, potenciando el proceso de instalación de software. El producto obtenido al final de la investigación, es decir, la personalización de la distribución cubana de GNU/Linux, Nova Server 2015 para el sistema PlatSI a través de una Imagen ISO DVD, facilita el trabajo de los especialistas de seguridad informática, tanto a la hora de instalar la aplicación, realizar las pruebas de seguridad, así como en la elaboración del informe final.

## **PALABRAS CLAVES**

Personalización de NOVA Server, Plataforma de Seguridad en las Tecnologías de la información.

## ÍNDICE DE CONTENIDO

INTRODUCCION .....	1
<b>CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA .....</b>	<b>6</b>
1.1    Conceptos fundamentales .....	6
1.2    Características generales de PlatSI .....	7
1.3    Características generales de Nova Server .....	8
1.4    Herramientas de personalización .....	9
1.5    Lenguaje de desarrollo .....	11
1.6    Metodología de desarrollo a utilizar .....	12
1.6.1    SXP, metodología seleccionada .....	14
1.7    Conclusiones parciales .....	16
<b>CAPITULO II. PROPUESTA DEL SISTEMA .....</b>	<b>17</b>
2.1    Introducción .....	17
2.2    Concepción del sistema .....	17
2.3    Captura de requisitos .....	21
2.3.1    Lista de reserva del producto .....	21
2.4    Historias de usuario .....	23
2.5    Fase de Planificación .....	27
2.6    Estimación de esfuerzo por historias de usuario .....	27
2.7    Lista de riesgos .....	28
2.8    Conclusiones parciales .....	29
<b>CAPITULO III. IMPLEMENTACIÓN Y PRUEBA .....</b>	<b>30</b>
3.1    Introducción .....	30
3.2    Arquitectura .....	30
3.3    Tareas de la ingeniería .....	33
3.4    Plan de iteraciones .....	42
3.5    Estándar de código .....	43
3.6    Estructura del Debian-Installer para la personalización de Nova server para el Sistema PlatSI .....	46
3.7    Como se generó el ISO de la personalización de Nova server 2015 para el sistema PlatSI .....	51
3.8    Pruebas .....	52
Conclusiones Generales .....	¡Error! Marcador no definido.
Recomendación .....	62

<b>Bibliografías .....</b>	<b>¡Error! Marcador no definido.</b>
<b>Anexo I .....</b>	<b>¡Error! Marcador no definido.</b>
<b>Anexo II .....</b>	<b>¡Error! Marcador no definido.</b>
<b>Anexo III .....</b>	<b>¡Error! Marcador no definido.</b>
<b>Glosario de términos .....</b>	<b>67</b>

## ÍNDICE DE TABLAS

Tabla 1. Lista de Reserva del Producto .....	21
Tabla 2. Instalar dependencias y paquetes .....	23
Tabla 3. Configurar dependencias y paquetes.....	24
Tabla 4. Configuración de PlatSI.....	24
Tabla 5. Crear la base de datos .....	25
Tabla 6. Cargar grupos y usuarios de prueba de PlatSI .....	25
Tabla 7. Configuración de Bambú-Server .....	26
Tabla 8. Configuración de Bambu-Client.....	26
Tabla 9. Estimación de esfuerzo por historia de usuario .....	27
Tabla 10. Lista de riesgos.....	28
Tabla 11.Tarea de ingeniería 1.....	33
Tabla 12.Tarea de ingeniería 2.....	34
Tabla 13.Tarea de ingeniería 3.....	34
Tabla 14.Tarea de ingeniería 4.....	35
Tabla 15.Tarea de ingeniería 5.....	35
Tabla 16.Tarea de ingeniería 6.....	36
Tabla 17.Tarea de ingeniería 7.....	36
Tabla 18.Tarea de ingeniería 8.....	37
Tabla 19.Tarea de ingeniería 9.....	37
Tabla 20.Tarea de ingeniería 10.....	38
Tabla 21.Tarea de ingeniería 11.....	38
Tabla 22.Tarea de ingeniería 12.....	39
Tabla 23.Tarea de ingeniería 13.....	39
Tabla 24.Tarea de ingeniería 14.....	40
Tabla 25.Tarea de ingeniería 15.....	40
Tabla 26.Tarea de ingeniería 16.....	41
Tabla 28. Plantilla de Releases.....	43
Tabla 29. Caso de Prueba de Aceptación HU01-P1 .....	52
Tabla 30. Caso de Prueba de Aceptación HU02-P1 .....	53
Tabla 31. Caso de Prueba de Aceptación HU03-P1 .....	54
Tabla 32. Caso de Prueba de Aceptación HU04-P1 .....	55
Tabla 33. Caso de Prueba de Aceptación HU05-P1 .....	55

<b>Tabla 34. Caso de Prueba de Aceptación HU06-P1 .....</b>	<b>56</b>
<b>Tabla 35. Caso de Prueba de Aceptación HU06-P2 .....</b>	<b>57</b>
<b>Tabla 36. Caso de Prueba de Aceptación HU07-P1 .....</b>	<b>58</b>
<b>Tabla 37. Caso de Prueba de Aceptación HU07-P2 .....</b>	<b>59</b>

## **ÍNDICE DE FIGURAS**

<b>Figura 1. Modelo del proceso de instalación manual de PlatSI .....</b>	<b>19</b>
<b>Figura 2. Propuesta de solución .....</b>	<b>21</b>
<b>Figura 3. Arquitectura en 3 Capas .....</b>	<b>31</b>
<b>Figura 4. Menú de configuración .....</b>	<b>32</b>
<b>Figura 5. Estructura del Debian-Installer de la personalización de Nova server ....</b>	<b>46</b>
<b>Figura 6. Código de la interfaz de las opciones de PlatSI .....</b>	<b>47</b>
<b>Figura 7. Código del archivo PlatSI-completo-install.sh .....</b>	<b>49</b>
<b>Figura 8. Estructura del directorio PlatSI en el Debian-Installer .....</b>	<b>50</b>
<b>Figura 10. Resultados de pruebas de aceptación.....</b>	<b>60</b>

## INTRODUCCIÓN

Las Tecnologías de la Información y la Comunicación, reconocidas por sus siglas TIC, son el conjunto de tecnologías desarrolladas para gestionar información. Abarcan un sin número de soluciones muy amplio, en las que se incluyen las tecnologías de almacenamiento y recuperación o simplemente procesamiento de datos para poder calcular resultados y elaborar informes. Esta revolución tecnológica ha sido propiciada por la aparición de la tecnología digital, la cual unida a la aparición de ordenadores cada vez más potentes, ha permitido a la humanidad progresar rápidamente en la ciencia y la técnica desplegando un arma más poderosa: la información y el conocimiento. [1]

Con la aparición de las TIC, Cuba 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 nuevos programas destinados a elevar el nivel cultural de la población y su calidad de vida. Como un centro de nuevo tipo, de alcance nacional, de características atípicas, con tareas concretas en el proyecto de informatización de la sociedad cubana y con énfasis en la producción de software, fue concebida, la Universidad de Ciencias Informáticas (UCI), gracias a la visión futurista del Comandante en Jefe Fidel Castro Ruz y en correspondencia con el caudal de transformaciones propiciado por las TIC. [2]

La UCI cuenta con un conjunto de centros vinculados a las actividades productivas, capaces de integrar los procesos docentes, productivos e investigativos de alto nivel; contando con un personal especializado en dichas áreas. Un ejemplo de ello es el Centro de Telemática, en el que se desarrollan sistemas informáticos y se ofertan servicios como son: consultorías a centro de datos, montaje de centros de llamadas y de datos, que inciden dentro de las ramas de las Telecomunicaciones y la Seguridad Informática.

El centro TLM desarrolla una herramienta denominada, Plataforma de Seguridad en las Tecnologías de la información (PlatSI) que permite la realización de pruebas de seguridad a aplicaciones web desde un único entorno que integra las herramientas Acunetix y Nikto con este objetivo. Este proceso facilita el trabajo de los especialistas de seguridad en cuanto a: accesibilidad,

debido a que las pruebas de seguridad se ejecutan de forma remota; usabilidad, ya que los reportes de las herramientas pertenecientes a una misma auditoria pueden ser consultados en la aplicación web facilitando la detección y análisis de las vulnerabilidades, así como la confección del informe final.

La instalación de la herramienta se convierte en un proceso engorroso que implica un gran esfuerzo debido a la instalación manual de un número elevado de dependencias con un orden específico dependiendo de un repositorio. En consecuencia el tiempo dedicado a la instalación de la herramienta es elevado lo cual hace complejo su despliegue. Además dichas dependencias tienen que estar disponibles en las versiones específicas que utiliza PlatSI para su correcto funcionamiento.

También se deben modificar varios archivos de configuración, específicamente en PlatSI (aplicación web) el archivo parameters.yml, en Bambu-Server el archivo server.conf y en Bambu-Client los archivos client.conf y main.conf, el objetivo de esta modificación es establecer los parámetros de red necesarios para la comunicación entre estos 3 componentes y la base de datos, ya sea en una misma PC o en diferentes PCs.

Teniendo en cuenta lo expuesto anteriormente, se plantea como **problema a resolver**:

¿Cómo incorporar los componentes del proceso de instalación del sistema PlatSI a una distribución GNU/Linux?

Se define como **objeto de estudio**: las distribuciones GNU/Linux y se enmarca como **campo de acción** los procesos asociados a la integración entre una distribución GNU/Linux y los componentes de instalación del sistema PlatSI.

Para dar solución al problema a resolver se define como **objetivo general**: desarrollar una personalización de la distribución NOVA Server 2015 para el sistema PlatSI, potenciando el proceso de instalación de software.

El proceso de investigación se estructuró a través de la proyección y cumplimiento de las siguientes tareas de investigación:

1. Determinación de los fundamentos y estado del arte acerca de las herramientas existentes para la personalización de distribuciones GNU/Linux que permita potenciar el proceso de instalación, profesionalización y alcance del software.
2. Determinación de las dificultades y potencialidades de los procesos asociados a la integración entre una distribución GNU/Linux y los componentes de instalación del sistema PlatSI para lograr el objetivo propuesto.
3. Estudio de las necesidades del proyecto PlatSI para identificar todas sus dependencias.
4. Análisis y selección de la metodología, herramientas y tecnologías a utilizar en la presente investigación para el desarrollo de la personalización de Nova server 2015.
5. Definición de la propuesta de solución de la investigación para definir las funcionalidades y requisitos que debe cumplir.
6. Definición de la estrategia de pruebas a seguir para verificar el correcto funcionamiento de la propuesta de solución.

Durante el desarrollo de la investigación se utilizan diferentes métodos de investigación, entre los que se encuentran: métodos del nivel empírico y métodos del nivel teórico.

Del nivel teórico: el análisis histórico-lógico, el analítico-sintético, el enfoque de sistema y la modelación.

Del nivel empírico: el análisis de documentos y la entrevista.

**Del nivel teórico:**

- Histórico-lógico: Se aplica para estudiar el desarrollo lógico histórico en la personalización de la distribución NOVA Server 2015 para el sistema PlatSI, potenciando el proceso de instalación.

- Analítico-sintético: Se emplea para determinar los fundamentos teóricos en la personalización de la distribución NOVA Server 2015 para el sistema PlatSI, así como para determinar y elaborar la vía de solución.
- Enfoque de sistema: Se pone en práctica en la investigación para determinar sus componentes, así como la relación entre ellos, a partir de la personalización de NOVA Server 2015 para el sistema PlatSI.

### **Del nivel empírico:**

Análisis de documentos: Se utilizó para revisar los documentos especializados en función de determinar los fundamentos teóricos que sustentan la investigación, así como para determinar y elaborar la vía de solución.

Entrevista: Se utilizó para la obtención de información del cliente sobre las dependencias que son empleadas en el proceso de instalación de PlatSI, así como los componentes que lo conforman.

El presente trabajo de diploma se ha estructurado en introducción, tres capítulos, conclusiones, recomendaciones, referencias bibliográficas, bibliografía, anexos y glosario de términos. Para una mejor comprensión se presenta una breve descripción de cada capítulo:

**Capítulo I. Fundamentación Teórica:** en este capítulo se presenta la definición del marco teórico de la investigación; además del estado del arte sobre herramientas que se usan para realizar personalizaciones, existentes en el mundo y en nuestro país, con el fin de seleccionar las correctas herramientas, tecnologías y lenguaje de desarrollo a utilizar en el proceso de la personalización propuesta.

**Capítulo II. Características del sistema:** Se definen las funcionalidades que debe cumplir la personalización, se elaboran las historias de usuario (HU) en conjunto con su planificación, se plasman las tablas de la lista de reserva del producto, la estimación de esfuerzo por historia de usuario y la lista de riesgos.

**Capítulo III. Implementación y Pruebas:** Se especifica la arquitectura a utilizar así como las tareas de ingenierías vinculadas a cada una de las historias de usuarios (HU) y se realizan las actividades de implementación y prueba de la solución propuesta.

## CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA

### 1.1 Introducción

Este capítulo comprende los principales conceptos que serán tratados durante la confección del trabajo, así como algunas herramientas, estándares y tecnologías existentes para el desarrollo de instaladores integrados a la personalización de NOVA Server 2015. Además se recogerán las características generales de PlatSI y del sistema operativo Nova, el lenguaje de desarrollo y metodología a utilizar.

### 1.2 Conceptos fundamentales

En este acápite se introducen los conceptos fundamentales que se estarán tratando a lo largo de la investigación.

#### **Distribución Linux**

Una distribución Linux (coloquialmente llamada distro) es una distribución de software basada en el núcleo Linux que incluye determinados paquetes de software para satisfacer las necesidades de un grupo específico de usuarios, dando así origen a ediciones domésticas, empresariales y para servidores. [3]

#### **Imagen ISO**

Es un tipo especial de archivo. Se le llama imagen porque es un "reflejo" exacto de todo lo que contenga el CD, DVD o BD (Blue-ray Disc) a partir del que se haya creado. El nombre ISO viene de las siglas en inglés de la Asociación Internacional de Estandarización, que fue quien definió sus características. Los archivos de este tipo tienen la extensión de archivo .iso.

Reduce a un solo archivo las múltiples carpetas y archivos originales, es una gran ventaja, sobre todo para transferir programas grandes o gran cantidad de datos a través de Internet, e-mail o una red doméstica. [4]

### 1.3 Características generales de PlatSI

#### **Bambu**

Bambú fue desarrollado en el centro TLM con vista a ser usado como marco de trabajo para la implementación de sistemas que trabajen de forma distribuida. Se implementó usando el lenguaje de programación Python, lo que facilita la asimilación y entendimiento de su código fuente. Es un sistema genérico que permite la ejecución de tareas de forma distribuida, así como la obtención y almacenamiento de sus resultados, está compuesto por el servidor y conectados a este uno o varios clientes Bambu. El servidor es el encargado de recibir cada tarea y asignársela a un agente que pueda realizarla. Por su parte los agentes son las PC que ejecutan finalmente las tareas y envían hacia el servidor el resultado obtenido. [5]

#### **PlatSI**

La Plataforma de Seguridad en las Tecnologías de la información (PlatSI) es una solución informática que permite realizar pruebas y detectar posibles brechas de seguridad en los sistemas web auditados. Para su funcionamiento se apoya en un conjunto de herramientas de seguridad usadas para la detección de vulnerabilidades de software, garantizando la integración de sus resultados y el conocimiento de los especialistas como un todo.

PlatSI se realizó haciendo uso del marco de trabajo para aplicaciones web Symfony2, se encuentra concebida para tres módulos, de ellos el Módulo de Auditorías a Sistemas Operativos (MASO) aún no se ha desarrollado y el Módulo de Auditorías a Bases de Datos (MABD) no se ha integrado a la Plataforma, aunque está terminado completamente. Actualmente solo se encuentra en funcionamiento el Módulo de Auditorías a Aplicaciones Web (MAAWeb).

Su funcionamiento básico consiste en la realización de pruebas de seguridad por un especialista en seguridad informática. Desde la Plataforma (aplicación web) se selecciona la herramienta que se quiere utilizar para la prueba (Acunetix o Nikto) y seguidamente se procede a introducir los datos que requiera la herramienta seleccionada, donde el especialista espera el resultado de vulnerabilidades detectadas.

Transparente a ese proceso, de forma interna la Plataforma contiene un sistema distribuidor de tareas (Bambu), que organiza las peticiones que pueden realizar varios clientes Bambu a la vez. Bambu utiliza ICE (Internet Communication Engine) para comunicarse con PlatSI (Aplicación Web) y entre el servidor y los clientes conectados. [5]

### **Relación entre PlatSI y Bambu**

Cuando un especialista ejecuta una herramienta de seguridad, la aplicación web de PlatSI, le encomienda a Bambu-Server su ejecución. Bambu-Server a su vez se encarga de reasignar la responsabilidad de la ejecución a uno de sus clientes (Bambu-Client) que esté habilitado y con capacidad para realizar el trabajo.

Una vez terminada la ejecución de la herramienta el agente en cuestión extrae los resultados relevantes y los envía a Bambu-Server, el cual los entrega a PlatSI web haciéndolos persistentes en la base de datos. Bambu entre otras cosas permite que se puedan integrar herramientas de plataformas diferentes (Linux/Windows) y el balanceo de la carga que supone tener muchos usuarios ejecutando herramientas de seguridad de forma simultánea. [5]

## **1.4 Características generales de Nova Server**

### **NOVA**

Nova es una distribución que utiliza el núcleo Linux e incluye determinados paquetes de aplicaciones informáticas para satisfacer las necesidades de la migración a plataformas de código abierto que experimenta Cuba como parte del proceso de Informatización de la Sociedad.

Se decidió utilizar en la personalización, Nova server 2015 ya que proporciona:

**Seguridad:** El modelo de desarrollo colaborativo, el acceso al código fuente y el exhaustivo proceso de revisión y auditoría de código garantiza un sistema seguro de virus y sin puertas traseras.

**Soberanía Tecnológica:** Mediante la formación de recursos humanos capacitados es un sistema operativo independiente, con capacidad decisional sobre las tecnologías reutilizadas y desarrolladas.

**Socio-adaptabilidad:** Es una distribución de GNU/Linux realizada en la Universidad de las Ciencias Informáticas, alineado a las políticas que orienta la informatización nacional y optimizado para las condiciones tecnológicas del país.

**Sostenibilidad:** La constante asimilación e investigación de las nuevas tecnologías, la planificación, los modelos novedosos de comercialización y el uso racional de los recursos humanos, materiales y naturales, garantiza vigencia y sostenibilidad a largo plazo. [6]

## 1.5 Metodología de desarrollo de software y Herramientas utilizadas

A continuación se describen las herramientas, tecnologías y metodología escogidas para guiar el proceso de desarrollo del software.

### 1.5.1 Herramientas de personalización

En esta sección se recoge algunas de las diferentes herramientas existentes y la usada para la personalización de NOVA Server 2015.

#### **Reconstructor**

Reconstructor es una herramienta para crear versiones personalizadas del sistema operativo GNU/Linux, en concreto Debían y Ubuntu, utiliza cualquier versión. Permite personalizar casi cualquier aspecto de la distribución: desde el software incluido hasta el aspecto visual. [7]

#### **ISO Master**

ISO Master es una aplicación gratuita y de código abierto multiplataforma con la que se puede extraer el contenido de CDs y DVDs a una imagen ISO, crear una nueva imagen de disco con datos almacenados en un ordenador o modificar los datos almacenados en un archivo ISO. Su interfaz

está especializada en la búsqueda, creación y edición de imágenes ISO, facilitando un panel de navegación para el sistema de archivo y otro panel con los datos almacenados (o que se vayan a almacenar) en la imagen ISO. [8]

### **Garfio**

Garfio es un sistema de infraestructura muy simple, desarrollado para crear distribuciones a medida, orientada a grupos concretos de usuarios, ya sea para universidades, colegios, cursos, trabajos, etc.

Garfio Genérico permite:

- Generar una distribución a partir de un sistema instalado. Distribución personalizada.
- Generar una copia de respaldo (Backup) completo del sistema, en un LiveCD.
- Generar copias de respaldo del entorno, es decir de la carpeta del usuario en /home y restaurarlas con un doble-clic. [9]

### **Debian-Installer**

Es el programa de instalación para Debian GNU/Linux. Es también el instalador oficial para Ubuntu y NOVA. Obtiene el núcleo y los paquetes de software apropiados de los repositorios relevantes, detecta el hardware, particiona el disco duro completo para aprovechar todo el espacio disponible, instala la distribución correspondiente y configura el gestor de arranque adecuado. Además permite instalar un sistema particular con aplicaciones empaquetadas y agregar programas en la etapa de pos-instalación. [10]

Las herramientas Reconstructor e ISO Master cumplen su cometido en general como se mencionó antes, realizan una serie de configuraciones en el sistema de fichero del sistema operativo, lo cual es ventajoso. Sin embargo, tienen como desventaja que toma el sistema base desde un Live CD/DVD previamente instalado, que puede o no estar actualizado y muchas veces no se conoce su origen y pone en riesgo la seguridad de determinada institución y la herramienta Garfio no permite

realizar modificaciones al código del sistema operativo para adicionarle las funcionalidades que se desean.

Por estas razones estas herramientas no se seleccionaron para realizar la personalización que dará solución al problema en cuestión. Por las opciones de Debian-Installer antes expuestas es que se selecciona esta herramienta para realizar la personalización que dará solución a la problemática planteada.

### **1.5.2 Sistema Gestor de Bases de Datos (SGBD):**

Un Sistema Gestor de Bases de Datos o SGBD, también llamado DBMS (Data Base Management System) es una colección de datos relacionados entre sí, estructurados y organizados, es un conjunto de programas que acceden y gestionan esos datos. La colección de esos datos se denomina Base de Datos. Algunos ejemplos de SGBD son Oracle, DB2, Postgresql, Mysql, MS SQL Server entre otros. [11]

Se utiliza Postgresql 9.3, pues el sistema PlatSI lo necesita para guardar los datos de las pruebas y los roles que intervienen. En la personalización es utilizado para crear la base de datos de PlatSI y los usuarios de prueba.

### **1.5.3 Herramienta case**

Herramienta CASE (del inglés, Computer Aided Software Engineering) con licencia gratuita, propicia un conjunto de ayudas para el desarrollo de programas informáticos dando soporte al modelado visual con UML (del inglés, Unified Modeling Language), desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. [12]

### **Visual Paradigm**

Es la herramienta CASE seleccionada que utiliza la notación BPMN para generar el diagrama de proceso del negocio sirviendo de apoyo en la confección de la propuesta de solución y permitiendo un mejor entendimiento de los procesos que se realizan en la instalación manual del sistema PlatSI.

#### **1.5.4 Notación para el Modelado de Procesos de Negocio (BPMN)**

Es una notación gráfica estandarizada que permite el modelado de procesos de negocio, en un formato de flujo de trabajo. El principal objetivo de BPMN es proporcionar una notación estándar que sea fácilmente legible y entendible por parte de todos los involucrados e interesados del negocio. Tiene la finalidad de servir como lenguaje común para cerrar la brecha de comunicación que frecuentemente se presenta entre el diseño de los procesos de negocio y su implementación. Se selecciona esta notación para modelar el proceso de negocio en el que se enmarca la presente investigación ya que provee una notación común para que los procesos puedan ser representados gráficamente de forma más clara, estandarizada y completa. [13]

#### **1.5.5 Lenguaje de desarrollo**

Bash: es un shell de Unix (intérprete de órdenes de Unix) escrito para el proyecto GNU. Además es el intérprete predeterminado en la mayoría de sistemas GNU/Linux y puede ejecutarse en la mayoría de los sistemas operativos tipo Unix. También se ha llevado a Microsoft Windows por el proyecto Cygwin. [14]

Se seleccionó Bash para el desarrollo de la personalización Nova server 2015 porque es el lenguaje que utiliza el Debian-Installer para manejar todo el proceso de post instalación.

#### **1.5.6 Metodología de desarrollo a utilizar**

Las metodologías se dividen en dos grupos: las metodologías pesadas o tradicionales y las metodologías ágiles, las cuales se rigen por los doce principios que forman el Manifiesto Ágil (Ver [Anexo I](#)). Estos doce principios se resumen en las 4 principales ideas que proyecta este tipo de metodología:

- Se encarga de valorar al individuo y las iteraciones del equipo más que a las herramientas o los procesos utilizados.
- Se hace mucho más importante crear un producto software que funcione que escribir mucha documentación.
- El cliente está en todo momento colaborando en el proyecto.
- Es más importante la capacidad de respuesta ante un cambio realizado que el seguimiento estricto de un plan. [15]

A continuación se muestran las descripciones de 3 metodologías ágiles, de las cuales se seleccionará una como guía en el desarrollo de la presente investigación.

### **SCRUM**

Es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

Se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

Scrum también se utiliza para resolver situaciones en que no se está entregando al cliente lo que necesita, cuando las entregas se alargan demasiado, los costes se disparan o la calidad no es aceptable, cuando se necesita capacidad de reacción ante la competencia, cuando la moral de los equipos es baja y la rotación alta, cuando es necesario identificar y solucionar ineficiencias

sistemáticamente o cuando se quiere trabajar utilizando un proceso especializado en el desarrollo de producto. [16]

### **XP**

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. Se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Es especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Consta de 4 fases, la primera es la planeación; dentro de la cual se encuentran las historias de usuario, la segunda es el diseño; que debe ser simple y aquí es donde se hacen las tarjetas CRC, la tercera es la codificación; donde la programación es en pareja y la integración es continua y la última es la fase de prueba. Esta tiene como objetivo: la satisfacción del cliente, potenciar el trabajado en grupo, minimizar el riesgo actuando sobre las variables del proyecto: coste, tiempo, calidad, alcance. [17]

### **SXP**

SXP, consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar el éxito del proyecto. Basada completamente en los valores y principios de las metodologías ágiles expuestos en el Manifiesto Ágil. Como método de estimación se utiliza la opinión de expertos y constan con métricas o indicadores para lograr una eficiente calidad.

Es una metodología compuesta por las metodologías SCRUM y XP que ofrece una estrategia tecnológica, a partir de la introducción de procedimientos ágiles que permitan actualizar los procesos de software para el mejoramiento de la actividad productiva fomentando el desarrollo de

la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del equipo, ayudando al líder del proyecto a tener un mejor control del mismo.

De SCRUM toma la forma de gestionar un equipo de manera que trabaje de forma eficiente y de tener siempre medidos los progresos, de forma que sepamos por donde andamos y de XP la manera de encaminar el desarrollo del software; que consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

SXP consta de 4 fases principales:

**Planificación – Definición:** Se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto.

**Desarrollo:** Se realiza la implementación del sistema hasta que esté listo para ser entregado.

**Entrega y Puesta en marcha:** Se entrega el software y se comienza a usar.

**Mantenimiento:** Se realiza el soporte para el cliente.

De cada una de estas fases se realizan numerosas actividades tales como el levantamiento de requisitos, la priorización de la Lista de Reserva del Producto, definición de las Historias de Usuario, pruebas, entre otras, de donde se generan artefactos para documentar todo el proceso. Las entregas son frecuentes y existe una refactorización continua, lo que nos permite mejorar el diseño cada vez que se le añada una nueva funcionalidad.

SXP está especialmente indicada para proyectos de pequeños equipos de trabajo, rápidos cambios de requisitos o requisitos imprecisos, muy cambiantes, donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad. Ayuda a que trabajen todos juntos, en la misma dirección, con un objetivo claro, permitiendo además seguir de forma clara el avance de las tareas a realizar, de forma que los jefes pueden ver día a día como progresa el trabajo. [18]

Se seleccionó como metodología a utilizar: SXP, debido a que el tiempo establecido para la realización de este trabajo es corto y el equipo de desarrollo es pequeño, constando con dos integrantes, al mismo tiempo se mantendrá una estrecha relación entre el cliente y el equipo de trabajo para la detección de los problemas que puedan existir, permitiendo reajustar el trabajo para eliminar dichos problemas. Además se adapta a la programación estructurada del lenguaje Bash, no siendo así en las metodologías XP y Scrum que están enfocadas a la programación orientada a objetos.

### **1.6 Conclusiones parciales**

En este capítulo se analizaron los principales conceptos para comprender el funcionamiento de los procesos asociados a la Plataforma de Seguridad de las Tecnologías de la Información y se realizó un estudio del arte de los sistemas existentes relacionados con la personalización de NOVA Server, llegando a la conclusión de que se utilizará Debian-Installer para realizar dicha personalización. Se hizo un análisis detallado de las herramientas y tecnologías a utilizar, definiendo a SXP como metodología de desarrollo y como lenguaje de programación se utilizará Bash. Para el manejo de diversas fuentes de datos en el software se hará uso del gestor de base de datos Postgresql. 9.3.

## CAPITULO II. PROPUESTA DEL SISTEMA

### 2.1 Introducción

En el presente capítulo se muestran los pasos a seguir para realizar el proceso de instalación de PlatSI manualmente, la propuesta de solución del trabajo presente, así como la fase de Planificación – Definición, la cual es la primera actividad propuesta por la metodología SXP. En esta fase se pueden observar los artefactos que se encuentran vinculados con la concepción del sistema, de modo que, se definen las Historias de Usuario asociadas a cada iteración, los requerimientos del sistema, la estimación de esfuerzos por Historias de Usuario, así como la lista de riesgos. Esta última permite definir aquellos elementos que representen una amenaza para el desarrollo del sistema.

### 2.2 Proceso de instalación manual de PlatSI

Con el objetivo de garantizar una mayor comprensión de cómo se realiza el proceso manual de instalación y configuración del sistema PlatSI en el Centro TLM, se muestra en la figura 1 el modelo de dicho proceso.

Actualmente el proceso de instalación de PlatSI es llevado a cabo por un especialista que se encarga de realizar los siguientes pasos:

- Instalar y configurar ICE para establecer la comunicación en la red de los diferentes componentes de PlatSI.
- Instalar y configurar las dependencias necesarias para el funcionamiento de PlatSI (Aplicación Web) entre las que se encuentra postgresql y apache2.
- Copiar los archivos de PlatSI (Aplicación Web) en el directorio público de apache /var/www.
- Modificar el archivo de configuración de PlatSI (Aplicación Web) para establecer la comunicación con el componente Bambu-Server y la base de datos.
- Crear la base de datos de PlatSI (Aplicación Web).
- Cargar los grupos y usuarios de prueba en la base de datos para así culminar con la instalación del primer componente.

- Para Bambu-Server es necesario instalar y configurar sus dependencias para su correcto funcionamiento entre las que se encuentra python-psycopg2.
- Copiar los archivos de Bambu-server en el directorio /usr/local/bambu.
- Configurar Bambu-Server como servicio.
- Modificar el archivo de configuración de Bambu-Server para establecer la comunicación con el componente PlatSI (Aplicación Web) y la base de datos.
- Iniciar el servicio bserver para culminar la instalación del tercer componente.
- Para Bambu-Client es necesario instalar y configurar sus dependencias para su correcto funcionamiento entre las que se encuentra sqlite y nikto.
- Copiar los archivos de Bambu-Client en el directorio /usr/local/bambu
- Configurar Bambu-Client como servicio
- Modificar los archivos de configuración de Bambu-Client para establecer la comunicación con Bambu-Server.
- Iniciar el servicio bclient para culminar la instalación del tercer componente.

# Capítulo 2 Propuesta del Sistema

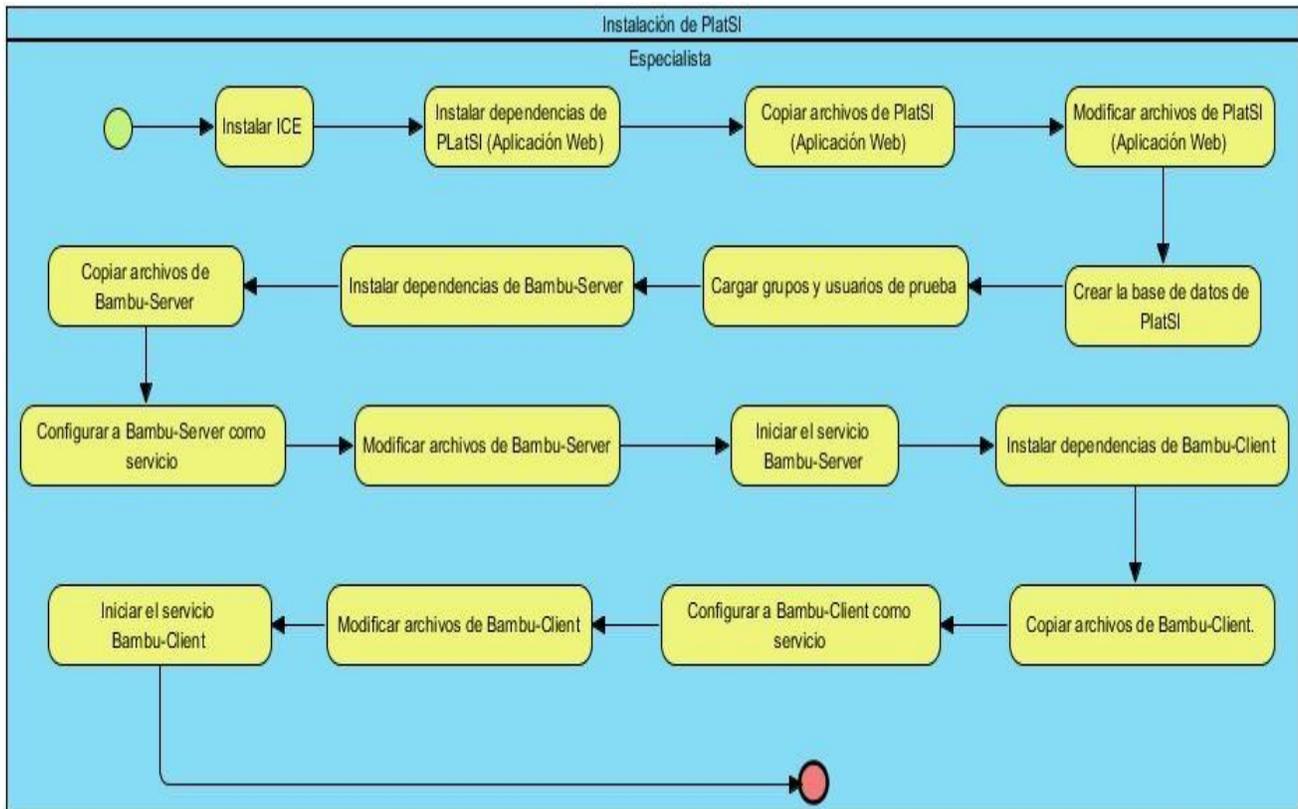


Figura 1. Modelo del proceso de instalación manual de PlatSI

## 2.3 Concepción del sistema

Los resultados del proceso investigativo concluyeron con la siguiente propuesta de solución, la personalización de Nova Server 2015 para el sistema PlatSI que se distribuirá a través de una Imagen ISO DVD dirigida al centro TLM que responde a las opciones de instalación de los siguientes módulos, PlatSI (Aplicación Web), Bambu-Server y Bambu-Client estos brindan la posibilidad de instalarse en la misma PC o en PC independientes de manera local. Para su correcto funcionamiento se instalan automáticamente paquetes y dependencias entre las que pueden encontrarse ICE como protocolo de comunicación y postgresql como gestor de Bases de datos.

Cuenta además con las disponibilidades de cargar usuarios de prueba con las autenticaciones definidas, sin la necesidad que los administradores posibiliten la creación de los usuarios para la

## Capítulo 2 Propuesta del Sistema

realización de pruebas. Se agrega además un repositorio interno que proporciona la instalación de paquetes y dependencias sin tener que conectarse a un repositorio en la red.

Para el desarrollo de la propuesta de solución el Debian Installer se abstrae de la arquitectura de Nova server permitiendo realizarle la personalización deseada a dicha distribución, influyendo solamente en el proceso de post instalación. Haciendo uso del Debian Installer se modificará el menú de instalación de Nova server 2015 el cual brindará las opciones de instalación de PlatSI. Además se crearán varios script que configuren y preparen al sistema para que instale automáticamente las dependencias desde el repositorio local y permitan además que el usuario modifique los parámetros de configuración de los tres módulos de PlatSI. Después será necesario añadir el directorio PlatSI donde se encuentran todos script necesarios para la instalación del sistema en la post instalación de nova server 2015 y por último generar el ISO de la personalización de Nova server 2015 para el sistema PlatSI, brindando el resultado final de la investigación.

En la figura 2 se muestra la organización de los archivos que serán creados dentro del ISO de NOVA Server para lograr la personalización.

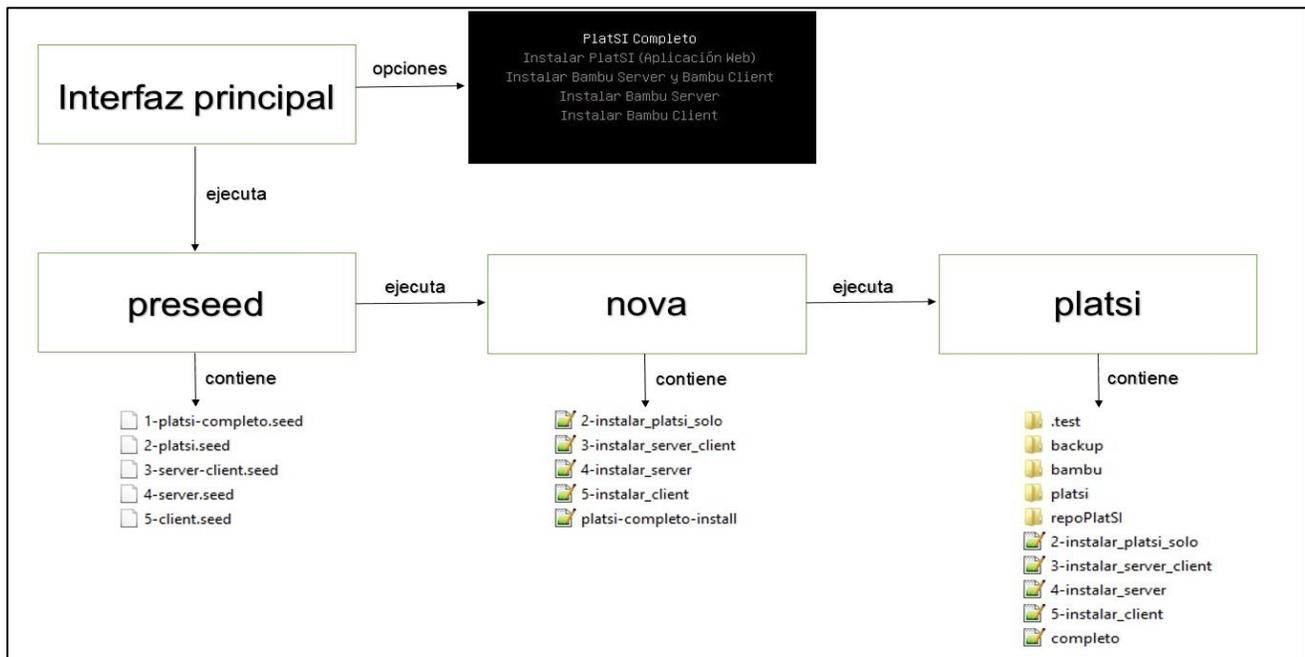


Figura 2. Propuesta de solución

## 2.4 Captura de requisitos

En este acápite se describen y especifican los requisitos funcionales y no funcionales que debe poseer la personalización Nova Server 2015 para el sistema PlatSI, los mismos fueron definidos en conjunto con el cliente.

### 2.4.1 Lista de reserva del producto

A continuación se muestra la plantilla de Lista de reserva del producto, es el primer artefacto generado en la etapa de Captura de requisitos, está conformada por una lista priorizada que define el trabajo que se va a realizar para la confección de la personalización. Realizando el llenado de esta plantilla con calidad, se garantizarán las ventajas siguientes:

- La organización de los requisitos, tanto funcionales como no funcionales, en dependencia de la prioridad que tengan para el desarrollo del sistema.
- Se facilitará el trabajo al confeccionar las historias de usuarios.

A continuación se muestran la LRP, perteneciente al trabajo en cuestión.

Tabla 1. Lista de Reserva del Producto

Asignado a	Ítem *	Descripción	Estimación	Estimado por
<b>Requisitos funcionales</b>				
<b>Prioridad</b>	<b>Alta</b>			
Jiubel Herrera Ymas y Yirai Pérez Suárez	1	Instalar dependencias y paquetes	2	Desarrollador
Jiubel Herrera Ymas y Yirai Pérez Suárez	2	Configurar dependencias	1	Desarrollador

## Capítulo 2 Propuesta del Sistema

Jiubel Herrera Ymas y Yirai Pérez Suárez	3	Configuración de PlatSI	2	Desarrollador
Jiubel Herrera Ymas y Yirai Pérez Suárez	4	Configuración de Bambu-Server	2	Desarrollador
Jiubel Herrera Ymas y Yirai Pérez Suárez	5	Configuración de Bambu-Client	2	Desarrollador
<b>Media</b>				
Jiubel Herrera Ymas y Yirai Pérez Suárez	6	Crear la base de datos de PlatSI	1	Desarrollador
Jiubel Herrera Ymas y Yirai Pérez Suárez	7	Cargar grupos y usuarios de prueba de PlatSI	1	Desarrollador
<b>Requisitos no funcionales</b>				
Jiubel Herrera Ymas y Yirai Pérez Suárez	1	Brindar un entorno de instalación fácil para el uso del usuario	---	Desarrollador
Jiubel Herrera Ymas y Yirai Pérez Suárez	2	Funcionar mínimo sobre computadoras de: Micro: 2.0 GB Ram: 2 GB Disco duro: 80 GB	---	Desarrollador

## 2.5 Historias de usuario

Las historias de usuario son la técnica utilizada en SXP para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento las historias de usuario pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas.

Las historias de usuario proporcionan ventajas, tales como:

- Están escritas en lenguaje del cliente, por lo que es muy fácil su comprensión.
- Especifican cada uno de los requisitos del sistema, sin necesidad de documentaciones extensas.
- Reflejan todas las características del sistema.
- Si se definen correctamente, guían el proceso de implementación.

A continuación se muestran las HU pertenecientes al trabajo en cuestión:

**Tabla 2. 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: Yirai Pérez Suárez, Jiubel Herrera Ymas	Iteración Asignada: 1
Prioridad en negocio: Alta	Puntos estimados:2
Riesgo en Desarrollo: Alto	Puntos Reales: 2

## Capítulo 2 Propuesta del Sistema

**Descripción:** Permite la instalación de dependencias y paquetes necesarios para el funcionamiento de ICE, PlatSI, Bambu-Server y Bambu-Client. Algunas de estas dependencias son: ice34-slice, libiceutil34, libqdbm14, php-zero-ice, ice34-translators, python-zero-ice, apache2, php-mdb2, postgresql, python-psycopg2, postgresql-common, sqlite, python-dev, nikto y libwhisker2-perl

**Tabla 3. Configurar dependencias y paquetes**

Historia de Usuario	
<b>Número:</b> 2	<b>Nombre de Historia de Usuario:</b> Configurar dependencias y paquetes
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Yirai Pérez Suárez, Jiubel Herrera Ymas	<b>Iteración Asignada:</b> 1
<b>Prioridad en negocio:</b> Alta	<b>Puntos estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Permite la configuración adecuada para el correcto funcionamiento de PlatSI. Ejemplo de esta configuración: postgres, apache2_ice	

**Tabla 4. Configuración de PlatSI**

Historia de Usuario	
<b>Número:</b> 3	<b>Nombre de Historia de Usuario:</b> Configuración de PlatSI
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Yirai Pérez Suárez, Jiubel Herrera Ymas.	<b>Iteración Asignada:</b> 2

<b>Prioridad en negocio:</b> Alta	<b>Puntos estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Permite copiar el código del sistema PlatSI para la raíz de apache e insertar los siguientes parámetros: IP del servidor de base de datos, puerto de conexión al servidor de base de datos, nombre de la base de datos, usuario por el cual se conectará a la base de datos, contraseña del usuario e IP donde se encuentra instalado Bambu-Server.	

Tabla 5. Crear la base de datos

Historia de Usuario	
<b>Número:</b> 4	<b>Nombre de Historia de Usuario:</b> Crear la base de datos de PlatSI
<b>Modificación de Historia de Usuario Número:</b> Ninguna.	
<b>Usuario:</b> Yirai Pérez Suárez, Jiubel Herrera Ymas.	<b>Iteración Asignada:</b> 2
<b>Prioridad en negocio:</b> Media	<b>Puntos estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Medio	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Permite al usuario de decidir si quiere crear la base de datos, crea el rol de administrador de la bases de datos e instala los plugin de Nikto y Acunetix.	

Tabla 6. Cargar grupos y usuarios de prueba de PlatSI

Historia de Usuario	
<b>Número:</b> 5	<b>Nombre de Historia de Usuario:</b> Cargar grupos y usuarios de prueba de PlatSI
<b>Modificación de Historia de Usuario Número:</b> Ninguna	

## Capítulo 2 Propuesta del Sistema

<b>Usuario:</b> Yirai Pérez Suárez, Jiubel Herrera Ymas.	<b>Iteración Asignada:</b> 2
<b>Prioridad en negocio:</b> Media	<b>Puntos estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Medio	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Permite seleccionar si se quiere instalar los usuarios y grupos por defecto para realizar las auditorias.	

**Tabla 7. Configuración de Bambú-Server**

Historia de Usuario	
<b>Número:</b> 6	<b>Nombre de Historia de Usuario:</b> Configuración de Bambu-Server
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Yirai Pérez Suárez, Jiubel Herrera Ymas.	<b>Iteración Asignada:</b> 3
<b>Prioridad en negocio:</b> Alta	<b>Puntos estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Permite al usuario insertar varios parámetros que sirven para conectar Bambu-Server y la base de datos: IP del servidor donde se encuentra la base de datos, usuario de la base de datos, contraseña para el usuario de la base de datos y nombre de la base de datos. Además copia archivos de Bambu-Server, pone a Bambu-Server como servicio e inicia el mismo.	

**Tabla 8. Configuración de Bambu-Client**

Historia de Usuario	
<b>Número:</b> 7	<b>Nombre de Historia de Usuario:</b> Configuración de Bambu-Client
<b>Modificación de Historia de Usuario Número:</b> Ninguna	

<b>Usuario:</b> Yirai Pérez Suárez, Jiubel Herrera Ymas.	<b>Iteración Asignada:</b> 3
<b>Prioridad en negocio:</b> Alta	<b>Puntos estimados:</b> 2
<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b> 2
<b>Descripción:</b> Permite al usuario insertar varios parámetros que sirven para conectar Bambu-Client y Bambu-Server: IP donde se encuentra instalado Bambu-Server. Además copia archivos de Bambu-Client, pone a Bambu-Client como servicio. También permite al usuario modificar la configuración principal de la herramienta Ontoger e inicia el servicio de Bambu-Client.	

## 2.6 Fase de Planificación

En la fase de planificación se plasma la estimación del esfuerzo que costará implementar cada HU, 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.

## 2.7 Estimación de esfuerzo por historias de usuario.

Tabla 9. Estimación de esfuerzo por historia de usuario

Historia de Usuario	Puntos de estimación
1 Instalar dependencias y paquetes	2
2 Configuración de dependencias	1
3 Configuración de PlatSI	1
4 Crear la base de datos de PlatSI	1
5 Cargar grupos y usuarios de prueba de PlatSI	1
6 Configuración de Bambu-Server	1
7 Configuración de Bambu-Client	2

### 2.8 Lista de riesgos

A continuación se muestra la plantilla de Lista de riesgos que es el documento que se genera de la actividad de valoración de riesgos. En ella quedan definidos los posibles riesgos que actuarán sobre el proceso de desarrollo de software y como se podrán mitigar. Esta plantilla propicia algunas ventajas, tales como:

- Se definen los posibles riesgos, así como la forma de mitigarlos, lo que disminuye el efecto de los mismos, si ocurrieran.
- Se lleva un control de todos los problemas que han influido en el proyecto, así como de la manera que fueron enfrentados y el impacto que tuvieron en el proceso de desarrollo.

A continuación se muestra la lista de riesgos pertenecientes al trabajo en cuestión:

**Tabla 10. Lista de riesgos**

Riesgo	Tipos de riesgos	Descripción	Probabilidad	Efectos	Mitigación del riesgo
Interferencia del proceso docente en la ejecución del proyecto	Organización	Pruebas y evaluaciones docentes en períodos del desarrollo	Baja	Tolerable	Programar tareas en horario fuera de la docencia
Retraso del proceso de desarrollo por eventos	Organización	Eventos masivos (políticos, culturales y de extensión universitaria)	Media	Tolerable	Programar tareas en horario que no se efectúen eventos
Pérdida de información importante.	Tecnológico	Rotura o descomposición de una computadora por afectaciones del fluido eléctrico u otros.	Alta	Serio	Copiar salvadas en diferentes máquinas o dispositivos
Presencia de virus informáticos.	Tecnológico	La PC es infectada por virus informáticos.	Baja	Tolerable	Tener el antivirus bien

					actualizado
Afectación del proceso de desarrollo por falta de preparación de los autores en el desarrollo de personalizaciones en el sistema operativo Nova Server.	Personal	El equipo de desarrollo no tiene experiencia en materias de personalizaciones de distribuciones GNU/Linux.	Alta	Serio	Estudiar documentos y artículos que para aumentar la preparación del tema

### 2.9 Conclusiones parciales

En este capítulo se describió la propuesta del sistema a desarrollar. Se identificaron los requisitos que el sistema debe cumplir, así como la descripción de las HU divididas por cada requisito y la estimación del esfuerzo dedicado a la realización de cada una de ellas en el orden en que se les dará cumplimiento según las necesidades del cliente. Con la obtención del plan de entregas y la planificación se logró delimitar el ciclo de desarrollo del sistema; además también se determinó la lista de riesgos con el fin de poder mitigarlos en caso de alguno de ellos se manifieste.

## CAPITULO III. IMPLEMENTACIÓN Y PRUEBA

### 3.1 Introducción

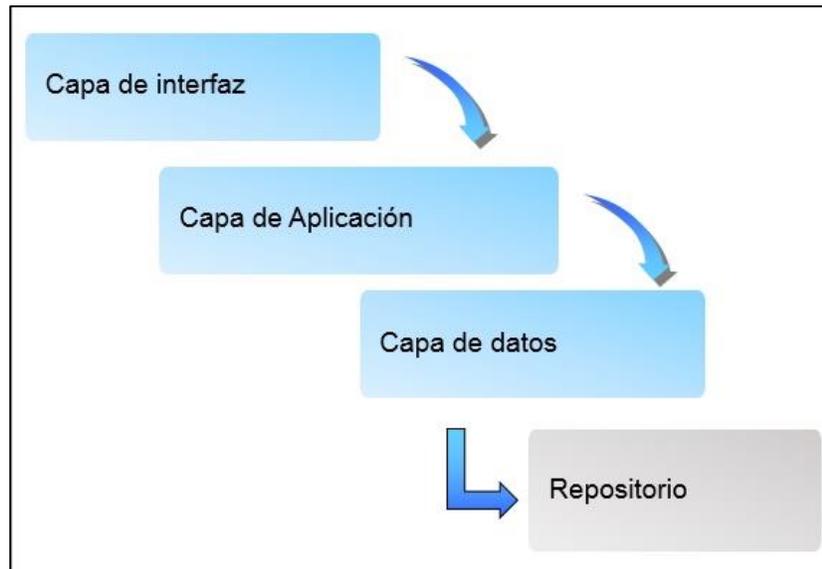
En el presente capítulo se generan los artefactos relacionados con la planificación de las iteraciones y tareas de ingeniería a realizar durante la implementación. Además se genera el código fuente en la etapa de implementación, se muestra la estructura final del Debian-Installer y se dejan plasmados los resultados arrojados por las pruebas realizadas.

### 3.2 Arquitectura

Una arquitectura de software es la estructura o las estructuras, que incluyen los componentes del software, las propiedades visibles externamente de esos componentes y las relaciones entre ellos. Es una representación que permite que un ingeniero de software, analice la efectividad del diseño para cumplir con los requisitos establecidos, considere opciones arquitectónicas en una etapa en que aún resulta relativamente fácil hacer cambios al diseño y reduzca los riesgos asociados con la construcción del software. [19]

Para la realización de la personalización de la distribución cubana de GNU/Linux Nova Server 2015 se utilizará la arquitectura N-Capas, específicamente de 3 Capas como se muestra en la figura 3, pues en la capa de interfaz o capa de servicios de usuario o presentación, se encontrarán los scripts de las vistas a través las cuales el usuario autorizado podrá decidir que opción escoger.

Una vez hecha la selección la capa de aplicación o capa de servicios de negocios, se encargará de ejecutarla, haciendo uso de los recursos que le ofrezca la capa de datos o capa de servicios de datos, que es donde se situarán los scripts que instalan las dependencias que utiliza el sistema PlatSI.



**Figura 3. Arquitectura en 3 Capas**

Esta arquitectura de 3 capas consiste en dividir software de gran tamaño en partes más pequeñas, esto puede hacer más simples los procesos de generarlo, reutilizarlo y modificarlo. Aunque, algunas veces, los niveles residen físicamente en máquinas diferentes debe enfatizarse en la distribución lógica de los mismos.

Los nombres de estos niveles difieren de acuerdo a la fuente, no obstante es bastante extendido el uso de las siguientes referencias en el modelo de 3 capas, el cual constituye el diseño más usado en la actualidad:

- Capa de servicios de usuario o presentación.
- Capa de servicios de negocios.
- Capa de servicios de datos.

A continuación se describen las características de estas 3 capas en la personalización de NOVA Server 2015 para el sistema PlatSI:

### **Capa de servicios de usuario o presentación**

# Capítulo 3 Implementación y Prueba

En una aplicación de 3 niveles, esta capa reúne todos los aspectos del software que tiene que ver con las interfaces y la interacción con los diferentes tipos de usuarios humanos. En el directorio raíz del Debian-Installer, se encuentra el directorio isolinux, que contiene el archivo de configuración del menú; en el cual se le muestra al usuario las opciones que puede seleccionar, dichas opciones se muestran en la figura 4:



**Figura 4. Menú de configuración**

## Capa de servicios de negocios

Esta capa reúne todos los aspectos del software que automatizan o apoyan los procesos de negocio que llevan a cabo los usuarios. Estos aspectos típicamente incluyen las tareas que forman parte de los procesos, las reglas y restricciones que aplican. La lógica de negocios construida en componentes lógicos personalizados enlaza los ambientes clientes y el nivel de servicios de datos.

Esta capa también recibe el nombre de la capa de la Lógica de la Aplicación. Las responsabilidades de esta capa se pueden sintetizar en:

- Recibir la entrada del nivel de presentación.
- Interactuar con los servicios de datos para poder ejecutar las operaciones de negocios que la aplicación automatiza.

En esta capa se encuentran 15 archivos, 5 de ellos de extensión .seed y los 10 restantes de extensión .sh, estos archivos se encargan de copiar todos los archivos y de ejecutar el instalador en la post instalación de Nova server 2015.

### Capa de servicios de datos

Esta capa reúne todos los aspectos del software que tienen que ver con el manejo de los datos, además contiene el repositorio de PlatSI con todas las dependencias necesarias para la instalación.

### 3.3 Tareas de la ingeniería

Las tareas de ingeniería es el primer artefacto de la fase de Desarrollo. Esta plantilla posee una gran importancia, pues permite definir cada una de las actividades que estarán asociadas a las historias de usuario y que permitirán su implementación. También posibilita conocer qué programador está asignado a cada tarea, así como el tiempo que se necesita para su realización, lo que facilita la estimación del tiempo que se llevará cada historia de usuario en implementarse, de acuerdo a su complejidad.

- Esta plantilla proporciona ventajas tales como:
- Permite organizar el proceso de implementación, pues las tareas se van implementando de acuerdo a su prioridad.
- Posibilita conocer el grado de complejidad de cada historia de usuario, teniendo en cuenta la cantidad de tareas asociadas.

A continuación se presentan las tareas de ingeniería estas se describen a partir de las historias de usuario descritas en el capítulo anterior.

Tabla 11.Tarea de ingeniería 1

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> 1
<b>Nombre Tarea:</b> Instalar ICE	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.4

# Capítulo 3 Implementación y Prueba

<b>Fecha Inicio:</b> 2-2-15	<b>Fecha Fin:</b> 4-2-15
<b>Programador Responsable:</b> Jiubel Herrera Ymas y Yirai Pérez Suárez	
<b>Descripción:</b> Instalar php-zero-ice y python-zero-ice con todas sus dependencias.	

Tabla 12.Tarea de ingeniería 2

Tarea de Ingeniería	
<b>Número Tarea:</b> 2	<b>Número Historia de Usuario:</b> 1
<b>Nombre Tarea:</b> Instalar apache2	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.8
<b>Fecha Inicio:</b> 4-2-15	<b>Fecha Fin:</b> 9-2-15
<b>Programador Responsable:</b> Jiubel Herrera Ymas y Yirai Pérez Suárez	
<b>Descripción:</b> Instalar php5, php5-gd, php-apc, php5-pgsql, php-mdb2-driver-pgsql, apache2 y libapache2-mod-php5 con todas sus dependencias.	

Tabla 13.Tarea de ingeniería 3

Tarea de Ingeniería	
<b>Número Tarea:</b> 3	<b>Número Historia de Usuario:</b> 1
<b>Nombre Tarea:</b> Instalar postgresql	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.4

# Capítulo 3 Implementación y Prueba

<b>Fecha Inicio:</b> 10-2-15	<b>Fecha Fin:</b> 11-2-15
<b>Programador Responsable:</b> Jiubel Herrera Ymas y Yirai Pérez Suárez	
<b>Descripción:</b> Instalar postgresql con todas sus dependencias.	

Tabla 14.Tarea de ingeniería 4

Tarea de Ingeniería	
<b>Número Tarea:</b> 4	<b>Número Historia de Usuario:</b> 1
<b>Nombre Tarea:</b> Instalar dependencias de Bambu-Server y Bambu-Client	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.4
<b>Fecha Inicio:</b> 12-2-15	<b>Fecha Fin:</b> 13-2-15
<b>Programador Responsable:</b> Jiubel Herrera Ymas y Yirai Pérez Suárez	
<b>Descripción:</b> Instalar python-psycpg2, sqlite, python-dev y nikto con todas sus dependencias.	

Tabla 15.Tarea de ingeniería 5

Tarea de Ingeniería	
<b>Número Tarea:</b> 5	<b>Número Historia de Usuario:</b> 2
<b>Nombre Tarea:</b> Configurar apache2-ice	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.4

# Capítulo 3 Implementación y Prueba

<b>Fecha Inicio:</b> 16-2-15	<b>Fecha Fin:</b> 17-2-15
<b>Programador Responsable:</b> Jiubel Herrera Ymas y Yirai Pérez Suárez	
<b>Descripción:</b> Se configura el apache2 e ice además de iniciar el servicio apache2.	

Tabla 16.Tarea de ingeniería 6

Tarea de Ingeniería	
<b>Número Tarea:</b> 6	<b>Número Historia de Usuario:</b> 2
<b>Nombre Tarea:</b> Configurar postgresql	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.6
<b>Fecha Inicio:</b> 18-2-15	<b>Fecha Fin:</b> 20-2-15
<b>Programador Responsable:</b> Jiubel Herrera Ymas y Yirai Pérez Suárez	
<b>Descripción:</b> Se configura postgresql creando usuario y contraseña.	

Tabla 17.Tarea de ingeniería 7

Tarea de Ingeniería	
<b>Número Tarea:</b> 7	<b>Número Historia de Usuario:</b> 3
<b>Nombre Tarea:</b> Copiar archivos de PlatSI	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.4

# Capítulo 3 Implementación y Prueba

<b>Fecha Inicio:</b> 23-2-15	<b>Fecha Fin:</b> 24-2-15
<b>Programador Responsable:</b> Jiubel Herrera Ymas y Yirai Pérez Suárez	
<b>Descripción:</b> copia los archivos que utilizan PlatSI en el sistema operativo y el código del proyecto para la raíz de apache.	

Tabla 18.Tarea de ingeniería 8

Tarea de Ingeniería	
<b>Número Tarea:</b> 8	<b>Número Historia de Usuario:</b> 3
<b>Nombre Tarea:</b> Configurar archivos de PlatSI	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.6
<b>Fecha Inicio:</b> 25-2-15	<b>Fecha Fin:</b> 27-2-15
<b>Programador Responsable:</b> Jiubel Herrera Ymas y Yirai Pérez Suárez	
<b>Descripción:</b> Configura el puerto de conexión al servidor de base de datos, nombre de la base de datos, usuario por el cual se conectará a la base de datos, contraseña del usuario y IP donde se encuentra instalado Bambu-Server. Además inicia PlatSI como servicio.	

Tabla 19.Tarea de ingeniería 9

Tarea de Ingeniería	
<b>Número Tarea:</b> 9	<b>Número Historia de Usuario:</b> 4

# Capítulo 3 Implementación y Prueba

<b>Nombre Tarea:</b> Crear la base de datos de PlatSI	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 2-3-15	<b>Fecha Fin:</b> 6-3-15
<b>Programador Responsable:</b> Jiubel Herrera Ymas y Yirai Pérez Suárez	
<b>Descripción:</b> Crea la base de datos de PlatSI e instala los plugins: Acunetix y Nikto.	

Tabla 20.Tarea de ingeniería 10

Tarea de Ingeniería	
<b>Número Tarea:</b> 10	<b>Número Historia de Usuario:</b> 5
<b>Nombre Tarea:</b> Cargar grupos y usuarios de prueba de PlatSI	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 9-3-15	<b>Fecha Fin:</b> 13-3-15
<b>Programador Responsable:</b> Jiubel Herrera Ymas y Yirai Pérez Suárez	
<b>Descripción:</b> Cargando grupos y usuarios de prueba: Administrador, planificador y especialista.	

Tabla 21.Tarea de ingeniería 11

Tarea de Ingeniería	
<b>Número Tarea:</b> 11	<b>Número Historia de Usuario:</b> 6

# Capítulo 3 Implementación y Prueba

<b>Nombre Tarea: Copiar archivos de Bambu-Server.</b>	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.2
<b>Fecha Inicio:</b> 16-3-15	<b>Fecha Fin:</b> 17-3-15
<b>Programador Responsable:</b> Jiubel Herrera Ymas y Yirai Pérez Suárez	
<b>Descripción:</b> Copia los archivos que utilizan Bambu-Server en el sistema operativo	

Tabla 22.Tarea de ingeniería 12

Tarea de Ingeniería	
<b>Número Tarea:</b> 12	<b>Número Historia de Usuario:</b> 6
<b>Nombre Tarea: Poner a Bambu-Server como servicio.</b>	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.2
<b>Fecha Inicio:</b> 18-3-15	<b>Fecha Fin:</b> 19-3-15
<b>Programador Responsable:</b> Jiubel Herrera Ymas y Yirai Pérez Suárez	
<b>Descripción:</b> Pone a Bambu-Server como servicio.	

Tabla 23.Tarea de ingeniería 13

Tarea de Ingeniería	
<b>Número Tarea:</b> 13	<b>Número Historia de Usuario:</b> 6
<b>Nombre Tarea: Configurar el archivo de bambú server.</b>	

# Capítulo 3 Implementación y Prueba

<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.6
<b>Fecha Inicio:</b> 19-3-15	<b>Fecha Fin:</b> 20-3-15
<b>Programador Responsable:</b> Jiubel Herrera Ymas y Yirai Pérez Suárez	
<b>Descripción:</b> Configura el IP del servidor donde se encuentra la base de datos, usuario de la base de datos, contraseña para el usuario de la base de datos y nombre de la base de datos. Además inicia Bambu-Server como servicio.	

Tabla 24.Tarea de ingeniería 14

Tarea de Ingeniería	
<b>Número Tarea:</b> 14	<b>Número Historia de Usuario:</b> 7
<b>Nombre Tarea:</b> Copiar archivos de Bambu-Client.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.4
<b>Fecha Inicio:</b> 23-3-15	<b>Fecha Fin:</b> 25-3-15
<b>Programador Responsable:</b> Jiubel Herrera Ymas y Yirai Pérez Suárez	
<b>Descripción:</b> Se copian los archivos de Bambu-Client para el sistema operativo.	

Tabla 25.Tarea de ingeniería 15

Tarea de Ingeniería	
<b>Número Tarea:</b> 15	<b>Número Historia de Usuario:</b> 7

<b>Nombre Tarea: Poner a Bambu-Client como servicio.</b>	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.4
<b>Fecha Inicio:</b> 25-3-15	<b>Fecha Fin:</b> 27-3-15
<b>Programador Responsable:</b> Jiubel Herrera Ymas y Yirai Pérez Suárez	
<b>Descripción:</b> Se inicia Bambu-Client como servicio para su posterior utilización.	

Tabla 26.Tarea de ingeniería 16

<b>Número Tarea:</b> 16	<b>Número Historia de Usuario:</b> 7
<b>Nombre Tarea: Configurar el archivo de bambú client.</b>	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.8
<b>Fecha Inicio:</b> 30-3-15	<b>Fecha Fin:</b> 1-4-15
<b>Programador Responsable:</b> Jiubel Herrera Ymas y Yirai Pérez Suárez	
<b>Descripción:</b> Configura el IP donde se encuentra instalado Bambu-Server.	

**Tabla 27. Tarea de ingeniería 17**

Tarea de Ingeniería	
<b>Número Tarea:</b> 17	<b>Número Historia de Usuario:</b> 7
<b>Nombre Tarea:</b> Configurar el archivo de la herramienta Ontoger.	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.4
<b>Fecha Inicio:</b> 1-4-15	<b>Fecha Fin:</b> 3-4-15
<b>Programador Responsable:</b> Jiubel Herrera Ymas y Yirai Pérez Suárez	
<b>Descripción:</b> Configura la herramienta Ontoger e inicia el servicio Bambu-Client.	

### 3.4 Plan de iteraciones

Después de ser identificadas y descritas las Historias de Usuario (HU) y estimar el esfuerzo dedicado a la realización de cada una de ellas, se procede a la planificación de la fase de implementación estableciendo una división de tres iteraciones.

- En la iteración 1 se realizará el desarrollo de las HU 1, 2.
- En la iteración 2 se realizará el desarrollo de las HU 3, 4, 5.
- En la iteración 3 se realizará el desarrollo de las HU 6 y 7.

La plantilla de Plan de releases, es un documento donde el cliente define el valor que posee el negocio según las características deseadas, es decir las historias de usuario. Los programadores proporcionan estimaciones de 1, 2 ó 3 puntos, y las historias mayores de los 3 puntos deben dividirse en otras más pequeñas. Esta plantilla proporciona ventajas, tales como:

- Define cuales son las historias de usuario más significativas, y las ubican en las iteraciones según esta prioridad.

# Capítulo 3 Implementación y Prueba

- Divide el proceso de desarrollo de software en iteraciones, planificando el trabajo a realizar en cada una de ellas.

A continuación se muestra en una tabla la plantilla de reléase correspondiente al trabajo en cuestión:

**Tabla 28. Plantilla de Releases**

Release	Descripción de la iteración	Orden de la HU a implementar	Duración total
1	En esta iteración se descargan las dependencias y se instalan para ver su correcto funcionamiento.	1,2	3 semanas
2	En esta iteración se deja ya funcionando correctamente PlatSI (Aplicación Web).	3, 4, 5	3 semanas
3	En esta iteración se deja ya funcionando correctamente Bambu- Server y Bambu- Client.	6, 7	3 semanas

### 3.5 Estándar de código

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez.

El propósito fundamental de este estándar de codificación, es que la personalización de nova server para el sistema PlatSI tenga un estilo consistente, independiente del autor, para que el sistema resulte fácil de entender y por supuesto fácil de mantener.

- Para la implementación del sistema, se decidió utilizar el siguiente estándar de código:
- Las declaraciones de funciones abren llaves en la línea inferior de la declaración, y cierran llaves con el mismo nivel de margen de la declaración, una línea después de la última sentencia.

Ejemplo:

```
function instalar_ice
{
    if [ "$1" = "php" ]; then
        echo "Instalando php-zero-ice"
        apt-get -y --force-yes install php-zero-ice
    elif [ "$1" = "python" ]; then
        echo "Instalando python-zero-ice"
        apt-get -y --force-yes install python-zero-ice
    else
        echo "Instalando php-zero-ice y python-zero-ice"
        apt-get -y --force-yes install php-zero-ice python-zero-ice
    fi
}
```

- Los nombres de las variables serán en letra mayúscula exceptuando las variables que se pasan por parámetros en las funciones, que irán en letra minúscula.

Ejemplo de variable global:

```
DIR="$PWD"
```

Ejemplo de variable por parámetros en una función:

```
instalar_ice python
```

- Los nombres de las funciones serán en letra minúscula separando los nombres compuestos con guion bajo.

Ejemplo:

```
function instalar_apache2
```

```
{
```

```
...
```

```
}
```

- En las instrucciones se plasman comentarios y mensajes en los que se explican la acción que se va a ejecutar.

Ejemplo:

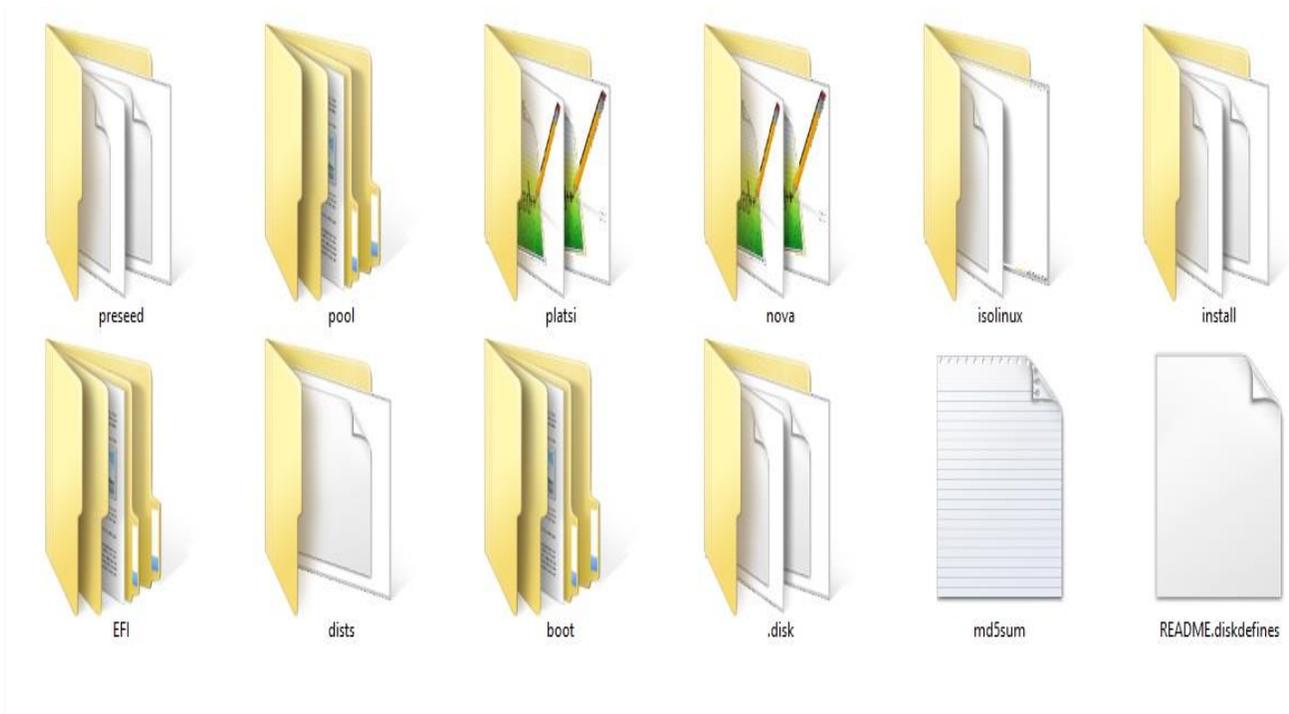
```
1-cp -r PlatSI /var/www/ # Copiando el código del proyecto para la raíz de apache
```

```
2-echo "Instalando plugin de Nikto"
```

```
php app/console PlatSI:bamplus:install_tool Nikto
```

## 3.6 Estructura del Debian-Installer para la personalización de Nova server para el Sistema PlatSI

Para la creación de personalización fue necesario añadir directorios y archivos al Debian-Installer para lograr la correcta instalación de PlatSI en la pos instalación de Nova server 2015, en la figura 5 se muestra como quedó estructurado por directorios.



**Figura 5. Estructura del Debian-Installer de la personalización de Nova server**

- `isolinux/txt.cfg`

Este archivo es donde se crea la interfaz que brinda las opciones de instalación PlatSI completo, PlatSI (Aplicación Web), (Bambu-Server y Bambu-Client), Bambu-Server, Bambu-Client, además de ejecutar los archivos del directorio Preseed según la opción que se seleccione. En la figura 6 se muestra el código de `txt.cfg`:

# Capítulo 3 Implementación y Prueba

```
default platsi-completo
label platsi-completo
    menu label PlatSI Completo
    kernel /install/vmlinuz
    append file=/cdrom/preseed/1-platsi-completo.seed vga=normal initrd=/install/initrd.gz -- quiet

label platsi
    menu label Instalar PlatSI (Aplicación Web)
    kernel /install/vmlinuz
    append file=/cdrom/preseed/2-platsi.seed vga=normal initrd=/install/initrd.gz -- quiet

label server-client
    menu label Instalar Bambu Server y Bambu Client
    kernel /install/vmlinuz
    append file=/cdrom/preseed/3-server-client.seed vga=normal initrd=/install/initrd.gz -- quiet

label server
    menu label Instalar Bambu Server
    kernel /install/vmlinuz
    append file=/cdrom/preseed/4-server.seed vga=normal initrd=/install/initrd.gz -- quiet

label client
    menu label Instalar Bambu Client
    kernel /install/vmlinuz
    append file=/cdrom/preseed/5-client.seed vga=normal initrd=/install/initrd.gz -- quiet
```

**Figura 6. Código de la interfaz de las opciones de PlatSI**

- Preseed

Este directorio contiene archivos encargados de preparar la instalación de PlatSI en la pos instalación de nova server 2015, copiar el directorio nova para el sistema operativo y ejecutar los archivos de la misma según la opción que se seleccione.

A continuación se muestra los archivos contenidos en el directorio Preseed:

- 1-PlatSI-completo.seed

- 2-PlatSI.seed
- 3-server-client.seed
- 4-server.seed
- 5-client.seed

Ejemplo de código personalizado del archivo PlatSI-completo.seed:

- d-i preseed/late\_command string \  
cp -r /target/media/cdrom/nova/ /target/tmp/ && \  
chroot /target sh /tmp/nova/PlatSI-completo-install.sh
- nova

Este directorio contiene archivos encargados de copiar el directorio PlatSI y crear el archivo configure-PlatSI en el sistema operativo. Además configura el sistema para que descargue e instale las dependencias del repositorio de PlatSI previamente copiado. Ejecuta el archivo de instalación de PlatSI y reconfigura el repositorio.

A continuación se muestra los archivos contenidos en el directorio Preseed:

- PlatSI-completo-install.sh
- 2-instalar\_PlatSI\_solo.sh
- 3-instalar\_server\_client.sh
- 4-instalar\_server.sh
- 5-instalar\_client.sh

En la figura 7 se muestra el código del archivo PlatSI-completo-install.sh que cumple las funcionalidades antes propuestas:

# Capítulo 3 Implementación y Prueba

```
#!/bin/bash
# copy *.deb files from CD to hard disk
PKG_DIR=/var/tmp/platsi
cp -av /media/cdrom/platsi $PKG_DIR 2> /dev/tty4
cat > /etc/rc.local << EOF
#!/bin/bash
service plymouth stop
chvt 1
echo "#!/bin/bash" >/usr/bin/configure-platsi.sh
echo "/etc/init.d/bclient stop" >>/usr/bin/configure-platsi.sh
echo "/etc/init.d/bserver stop" >>/usr/bin/configure-platsi.sh
echo "nano /var/www/platsi/app/config/parameters.yml" >>/usr/bin/configure-platsi.sh
echo "nano /usr/local/bambu/server/server.conf" >>/usr/bin/configure-platsi.sh
echo "nano /usr/local/bambu/client/client.conf" >>/usr/bin/configure-platsi.sh
echo "nano /usr/local/bambu/client/Tools/Ontoger/src/.main.conf" >>/usr/bin/configure-platsi.sh
echo "/etc/init.d/bclient start" >>/usr/bin/configure-platsi.sh
echo "/etc/init.d/bserver start" >>/usr/bin/configure-platsi.sh
chmod +x /usr/bin/configure-platsi.sh

echo "deb [ arch=amd64 ] file://$PKG_DIR/repoPlatSI/nova 2015 principal extendido" > /etc/apt/sources.list
apt-get update &>/dev/tty1
cd /var/tmp/platsi
bash completo.sh &>/dev/tty1

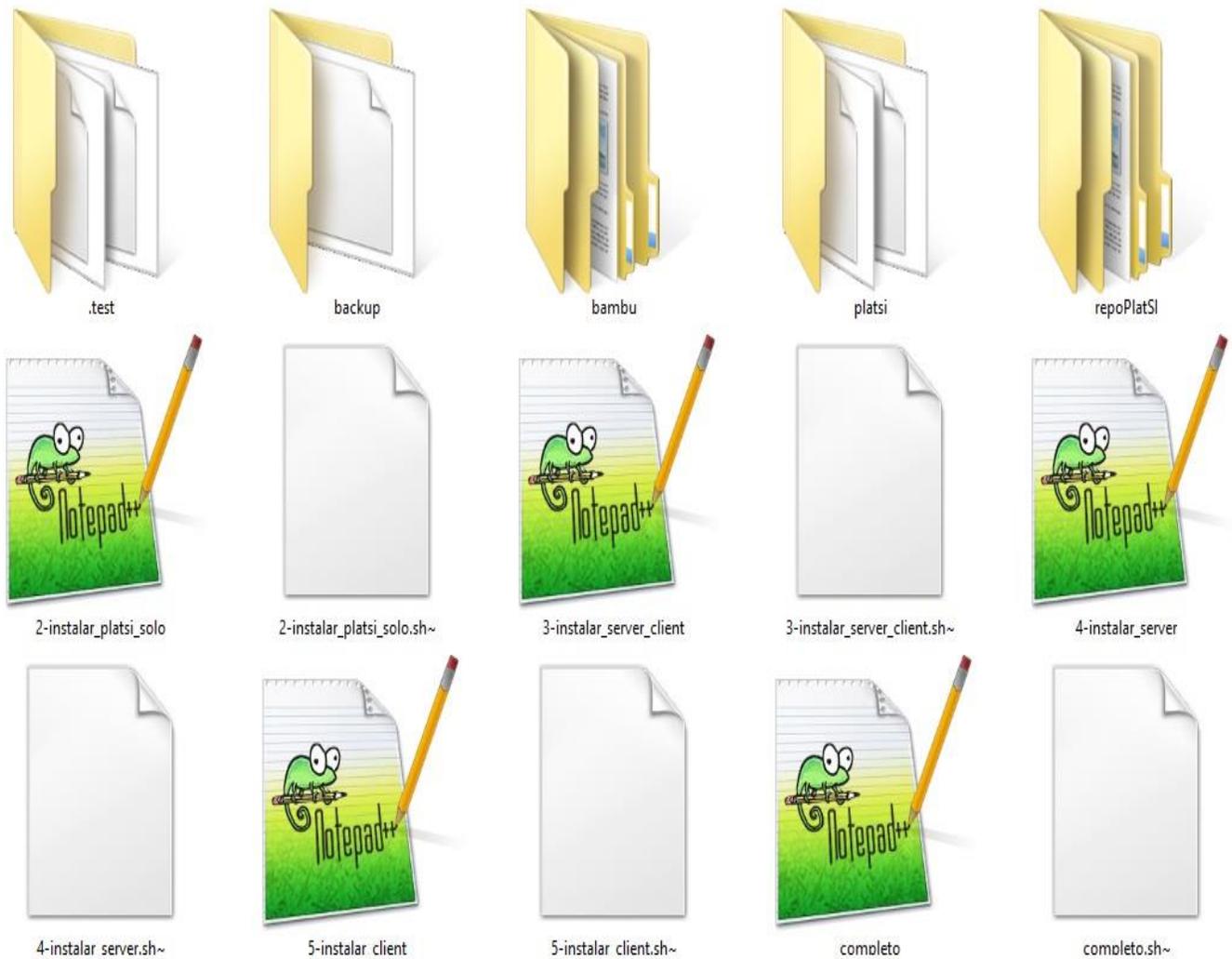
echo "deb http://novarepo.uci.cu/nova 2015 principal extendido" > /etc/apt/sources.list
echo "#!/bin/sh -e" > /etc/rc.local
echo "#>> /etc/rc.local
echo "# rc.local">> /etc/rc.local
echo "#>> /etc/rc.local
echo "# This script is executed at the end of each multiuser runlevel.">> /etc/rc.local
echo "# Make sure that the script will "exit 0" on success or any other">> /etc/rc.local
echo "# value on error.">> /etc/rc.local
echo "#>> /etc/rc.local
echo "# In order to enable or disable this script just change the execution">> /etc/rc.local
echo "# bits.">> /etc/rc.local
echo "#>> /etc/rc.local
echo "# By default this script does nothing.">> /etc/rc.local
echo "#>> /etc/rc.local
echo "exit 0" >> /etc/rc.local
EOF
### CUSTOM_ACTION ###
sync
```

Figura 7. Código del archivo PlatSI-completo-install.sh

# Capítulo 3 Implementación y Prueba

- PlatSI

Este directorio se encuentra todos los archivos por los que está compuesto PlatSI incluyendo los script de instalación y el repositorio interno, en la figura 8 se muestra como quedo estructurado dicho directorio.



**Figura 8. Estructura del directorio PlatSI en el Debian-Installer**

A continuación se muestra la figura 9 con un fragmento de código del archivo `completo.sh` que es el encargado de la instalación de PlatSI (aplicación web):

# Capítulo 3 Implementación y Prueba

```
function instalar_platsi
{
    instalar_apache2 #Se instala php5 y las dependencias necesarias
    instalar_postgresql
    cd "$DIR"
    echo "Copiando archivos de PlatSI"
    cp -r platsi /var/www/ # Copiando el código del proyecto para la raiz de apache
    cd /var/www/platsi
    sed -i "s/labsi_bamplus.iceconnection.host: 10.128.50.236/labsi_bamplus.iceconnection.host:
$( ifconfig eth0 | grep 'inet:10' | awk -F '[:]+' '{print $4}']/g"
/var/www/platsi/app/config/parameters.yml
#Corriendo los ASSETS para habilitar css, images y javascript
php app/console assets:install --symlink
php app/console doctrine:database:create
php app/console doctrine:schema:create
echo "Cargando datos (fixtures)..."
php app/console doctrine:fixtures:load --no-interaction
php app/console fos:user:create admin admin@uci.cu admin
php app/console fos:user:promote admin ROLE_ADMIN
echo "Instalando plugin de Nikto"
php app/console platsi:bamplus:install_tool Nikto
echo "Instalando plugin de Acunetix"
php app/console platsi:bamplus:install_tool Acunetix
echo "Cargando grupos y usuarios de prueba: planificador, especialista, solicitante."
su -c "psql -d platsidb < $DIR/backup/userbackup.sql" postgres
echo "Eliminando Cache..."
rm -r app/cache/*
rm -r app/logs/*
chown -R www-data:www-data /var/www/platsi
cd "$DIR"
service apache2 reload
}
```

Figura 9. Fragmento de código del archivo completo.sh

## 3.7 Como se generó el ISO de la personalización de Nova server 2015 para el sistema PlatSI

Después de hacer cualquier cambio en el Debian-Installer es necesario volver a generar el ISO para realizar las pruebas pertinentes, el siguiente código es utilizado para realizar dicha tarea:

# Capítulo 3 Implementación y Prueba

```
mkisofs -r -V "Nova para Servidores 2015 amd64" -cache-inodes -J -l -b isolinux/isolinux.bin -c  
isolinux/boot.cat -no-emul-boot -boot-load-size 4 -boot-info-table -o ../build/nova-servidores-2015-  
amd64-beta2-$(date +%A-%F-%H-%M-%S).iso
```

## 3.8 Pruebas

La metodología SXP, propone la realización de Caso de prueba de aceptación, el objetivo de dichas pruebas es validar que un sistema cumple con el funcionamiento esperado y permitir al usuario de dicho sistema que determine su aceptación, desde el punto de vista de su funcionalidad y rendimiento. Las pruebas de aceptación son definidas por el cliente y preparadas por el equipo de desarrollo, aunque la ejecución y aprobación final corresponden al cliente. La utilización de estas proporcionan ventajas, tales como:

- Son el termómetro de los desarrolladores, fundamentalmente de los programadores a la hora de medir la calidad de su trabajo.
- Garantizan la entrega de un producto con calidad, que responde a las necesidades del cliente.

Los casos de pruebas de aceptación se presentan a continuación:

**Tabla 29. Caso de Prueba de Aceptación HU01-P1**

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> HU01-P1	<b>Nombre Historia de Usuario:</b> Instalar dependencias y paquetes
<b>Nombre de la persona que realiza la prueba:</b> Yirai Pérez Suárez, Jiubel Herrera Ymas	
<b>Descripción de la Prueba:</b> Se comprueba que todas los paquetes y dependencias estén instaladas correctamente	
<b>Condiciones de Ejecución:</b> El usuario debe tener los permisos suficientes para realizar esta operación.	

# Capítulo 3 Implementación y Prueba

<p><b>Entrada / Pasos de ejecución:</b></p> <p>1-Entrar al nova server donde está instalado PlatSI como sudo su</p> <p>2-Verificar que cada dependencia se instaló correctamente:</p> <ul style="list-style-type: none"> <li>- php-zero-ice</li> <li>- python-zero-ice</li> <li>- php5, php5-gd, php5-apc, php5-pgsql, php-mdb2-driver-pgsql</li> <li>- apache2, libapache2-mod-php5</li> <li>- postgresql</li> <li>- python-psycopg2</li> <li>- sqlite, python-dev, nikto</li> </ul>
<p><b>Resultado Esperado:</b> Que se instalen correctamente todas las dependencias</p>
<p><b>Evaluación de la Prueba:</b> Satisfactoria.</p>

**Tabla 30. Caso de Prueba de Aceptación HU02-P1**

Caso de Prueba de Aceptación	
<p><b>Código Caso de Prueba:</b></p> <p>HU02-P1</p>	<p><b>Nombre Historia de Usuario:</b> Configurar dependencias y paquetes</p>
<p><b>Nombre de la persona que realiza la prueba:</b> Yirai Pérez Suárez, Jiubel Herrera Ymas</p>	
<p><b>Descripción de la Prueba:</b> Se comprueba que postgres y apache2_ice estén configuradas correctamente.</p>	
<p><b>Condiciones de Ejecución:</b> El usuario debe tener los permisos suficientes para realizar esta operación.</p>	

# Capítulo 3 Implementación y Prueba

<p><b>Entrada / Pasos de ejecución:</b></p> <p>1-Conectar con el gestor de base de datos postgresql, con las credenciales que están definidas en la instalación del sistema.</p> <p>2-Entrar al nova server donde se instaló PlatSI y registrarse como administrador.</p> <p>3-Comprobar que el apache este corriendo correctamente.</p> <p>4-Comprobar que se pueda acceder la aplicación web desde otra pc.</p>
<p><b>Resultado Esperado:</b> Que se configuraren correctamente el postgres y el apache</p>
<p><b>Evaluación de la Prueba:</b> Satisfactoria.</p>

**Tabla 31. Caso de Prueba de Aceptación HU03-P1**

Caso de Prueba de Aceptación	
<p><b>Código Caso de Prueba:</b></p> <p>HU03-P1</p>	<p><b>Nombre Historia de Usuario:</b> Configuración de PlatSI</p>
<p><b>Nombre de la persona que realiza la prueba:</b> Yirai Pérez Suárez, Jiubel Herrera Ymas</p>	
<p><b>Descripción de la Prueba:</b> Se comprueba que los archivos de PlatSI se hayan copiado adecuadamente, además de verificar que el archivo configure-PlatSI se haya creado en el sistema operativo Nova server anteriormente instalado, además de comprobar que los parámetros se modifiquen correctamente.</p>	
<p><b>Condiciones de Ejecución:</b> El usuario debe tener los permisos suficientes para realizar esta operación.</p>	
<p><b>Entrada / Pasos de ejecución:</b></p> <p>1-Entrar al nova server donde se instaló PlatSI y registrarse como administrador</p> <p>2-Comprobar que los archivos de PlatSI se copiaron correctamente.</p> <p>3-Ejecutar configure-PlatSI.sh</p>	

# Capítulo 3 Implementación y Prueba

4-Revisar manualmente si se guardaron los cambios en los archivos de configuración.
<b>Resultado Esperado:</b> Que se copien correctamente todos los archivos de PlatSI, se genere el archivo configure-PlatSI y se modifiquen los archivos de configuración.
<b>Evaluación de la Prueba:</b> Satisfactoria.

**Tabla 32. Caso de Prueba de Aceptación HU04-P1**

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> HU04-P1	<b>Nombre Historia de Usuario:</b> Crear la base de datos.
<b>Nombre de la persona que realiza la prueba:</b> Yirai Pérez Suárez, Jiubel Herrera Ymas	
<b>Descripción de la Prueba:</b> Se comprueba que la base de datos se ha creado correctamente.	
<b>Condiciones de Ejecución:</b> El usuario debe tener los permisos suficientes para realizar esta operación.	
<b>Entrada / Pasos de ejecución:</b>  1-Conectarse al servidor de base de datos y comprobar que se haya creado la base de datos.  2-Acceder mediante una computadora por el navegador al ip donde está montado el PlatSI aplicación web y comprobar que los datos insertados se guarden en la base de datos.	
<b>Resultado Esperado:</b> Que se genere correctamente la base de datos y se guarden correctamente los datos insertados.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

**Tabla 33. Caso de Prueba de Aceptación HU05-P1**

Caso de Prueba de Aceptación
------------------------------

# Capítulo 3 Implementación y Prueba

<b>Código Caso de Prueba:</b> HU05-P1	<b>Nombre Historia de Usuario:</b> Cargar grupos y usuarios de prueba de PlatSI.
<b>Nombre de la persona que realiza la prueba:</b> Yirai Pérez Suárez, Jiubel Herrera Ymas	
<b>Descripción de la Prueba:</b> Se comprueba que los usuarios de prueba se hayan creado correctamente.	
<b>Condiciones de Ejecución:</b> El usuario debe tener los permisos suficientes para realizar esta operación.	
<b>Entrada / Pasos de ejecución:</b>  1-Comprobar que los usuarios estén creados correctamente en la base de datos.  2-Acceder mediante una computadora por el navegador al ip donde está montado el PlatSI aplicación web y loguearse con el usuario y contraseña predefinido.	
<b>Resultado Esperado:</b> Que se generen correctamente los usuarios de prueba en la base de datos y se acceda a la aplicación con los usuarios predefinidos.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

Tabla 34. Caso de Prueba de Aceptación HU06-P1

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> HU06-P1	<b>Nombre Historia de Usuario:</b> Configuración de Bambu-Server.
<b>Nombre de la persona que realiza la prueba:</b> Yirai Pérez Suárez, Jiubel Herrera Ymas	
<b>Descripción de la Prueba:</b> Se comprueba que los archivos de Bambu-Server se hayan copiado adecuadamente, se verifica que el archivo configure-PlatSI se haya creado, se ejecuta y se comprueba si se guardaron los cambios en los archivos de configuración. Además que el servicio bserver este corriendo adecuadamente en el sistema operativo nova-server anteriormente instalado.	

# Capítulo 3 Implementación y Prueba

<p><b>Condiciones de Ejecución:</b> El usuario debe tener los permisos suficientes para realizar esta operación.</p>
<p><b>Entrada / Pasos de ejecución:</b></p> <ol style="list-style-type: none"> <li>1-Entrar al nova server donde se instaló Bambu-Server y registrarse como administrador.</li> <li>2-Comprobar que los archivos de Bambu-Server se copiaron correctamente.</li> <li>3-Ejecutar configure-PlatSI.sh</li> <li>4- Revisar manualmente si se guardaron los cambios en los archivos de configuración.</li> <li>5-Verificar que bserver este corriendo.</li> <li>6-Revisar los log para ver si no aparece ningún error.</li> </ol>
<p><b>Resultado Esperado:</b> Que se copien correctamente todos los archivos de Bambu-Server, se genere el archivo configure-PlatSI, se guarden los cambios en los archivos de configuración, que bserver esté corriendo correctamente y los log no muestren error.</p>
<p><b>Evaluación de la Prueba:</b> Satisfactoria.</p>

**Tabla 35. Caso de Prueba de Aceptación HU06-P2**

Caso de Prueba de Aceptación	
<p><b>Código Caso de Prueba:</b> HU06-P2</p>	<p><b>Nombre Historia de Usuario:</b> Configuración de Bambu-Server.</p>
<p><b>Nombre de la persona que realiza la prueba:</b> Yirai Pérez Suárez, Jiubel Herrera Ymas</p>	
<p><b>Descripción de la Prueba:</b> Se utiliza un PlatSI (aplicación web) y un cliente previamente instalado con el Bambu-Server para ver el comportamiento en sus los log.</p>	
<p><b>Condiciones de Ejecución:</b> El usuario debe tener los permisos suficientes para realizar esta operación.</p>	
<p><b>Entrada / Pasos de ejecución:</b></p>	

# Capítulo 3 Implementación y Prueba

<p>1-Acceder mediante una computadora por el navegador al ip donde está montado el PlatSI aplicación web y mandar a ejecutar una prueba.</p> <p>2-Verificar los log de Bambu-Server para comprobar que está registrado la tarea que le envió al cliente de bambu.</p> <p>3-Abrir el log del cliente de bambu y revisar que recibió la información de la tarea.</p>
<p><b>Resultado Esperado:</b> Que los log del cliente y el servidor no muestren errores.</p>
<p><b>Evaluación de la Prueba:</b> Satisfactoria.</p>

**Tabla 36. Caso de Prueba de Aceptación HU07-P1**

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> HU07-P1	<b>Nombre Historia de Usuario:</b> Configuración de Bambu-Client.
<b>Nombre de la persona que realiza la prueba:</b> Yirai Pérez Suárez, Jiubel Herrera Ymas	
<b>Descripción de la Prueba:</b> Se comprueba que los archivos de Bambu-Client se hayan copiado adecuadamente, se verifica que el archivo configure-PlatSI se haya creado, se ejecuta y se comprueba si se guardaron los cambios en los archivos de configuración. Además que el servicio bclient este corriendo adecuadamente en el sistema operativo nova-server anteriormente instalado.	
<b>Condiciones de Ejecución:</b> El usuario debe tener los permisos suficientes para realizar esta operación.	
<b>Entrada / Pasos de ejecución:</b>	
<p>1-Entrar al nova server donde se instaló Bambu-Client y registrarse como administrador.</p> <p>2-Comprobar que los archivos de Bambu-Client se copiaron correctamente.</p> <p>3-Ejecutar configure-PlatSI.sh</p> <p>4- Revisar manualmente si se guardaron los cambios en los archivos de configuración.</p>	

# Capítulo 3 Implementación y Prueba

5-Verificar que bclient este corriendo.
6-Revisar los log para ver si no aparece ningún error.
<b>Resultado Esperado:</b> Que se copien correctamente todos los archivos de Bambu-Client, se genere el archivo configure-PlatSI, se guarden los cambios en los archivos de configuración, bclient esté corriendo correctamente y los log no muestren error.
<b>Evaluación de la Prueba:</b> Satisfactoria.

**Tabla 37. Caso de Prueba de Aceptación HU07-P2**

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> HU07-P2	<b>Nombre Historia de Usuario:</b> Configuración de Bambu-Client.
<b>Nombre de la persona que realiza la prueba:</b> Yirai Pérez Suárez, Jiubel Herrera Ymas	
<b>Descripción de la Prueba:</b> Se utiliza un PlatSI (aplicación web) y un Bambu-Server previamente instalado con el Bambu-Client para ver el comportamiento de sus log.	
<b>Condiciones de Ejecución:</b> El usuario debe tener los permisos suficientes para realizar esta operación.	
<b>Entrada / Pasos de ejecución:</b>  1-Acceder mediante una computadora por el navegador al ip donde está montado el PlatSI aplicación web y mandar a ejecutar una prueba.  2-Verificar los log de Bambu-Server para comprobar que está registrado la tarea que le envió al cliente de bambu.  3-Abrir el log del cliente de bambu y revisar que recibió la información de la tarea.	
<b>Resultado Esperado:</b> Que los log no muestren errores.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

# Capítulo 3 Implementación y Prueba

Al culminar las pruebas de aceptación se obtuvo como resultado 4 no conformidades en la primera entrega tales como errores en la instalación de dependencias por problemas de versiones, reflejadas en las HU 1, las cuales fueron resueltas. En la segunda entrega se detectaron 2 no conformidades, no se guardaron los cambios en los archivos de configuración y no se crearon los grupos de usuarios de prueba pertenecientes a la HU 3 y a la HU 4 respectivamente, las cuales fueron resueltas. En la tercera y última entrega, luego de realizar las pruebas no se detectaron no conformidades logrando resultados satisfactorios y con ello la satisfacción del cliente. Lo anteriormente expuesto se evidencia en la figura 10.

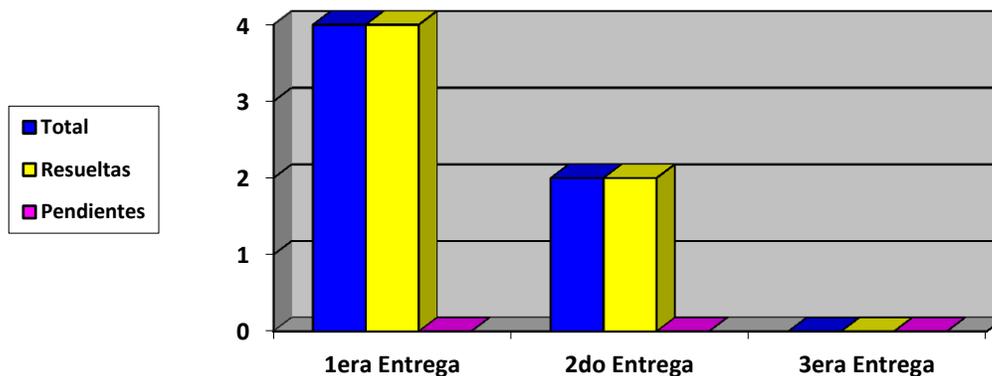


Figura 10. Resultados de pruebas de aceptación

### 3.9 Conclusiones parciales

En este capítulo se describió la arquitectura a utilizar y como se estructuró la misma en el presente trabajo. Se realizaron las tareas de ingeniería correspondientes a cada una de las historias de usuarios. También se realizó el plan de iteraciones, con el fin de determinar qué se hará en cada momento de la investigación. Además se dejó plasmado el estándar de código utilizado, la estructura específica del Debian-Installer para la presente personalización y cómo se generó el iso de la personalización. Igualmente se realizaron los casos de pruebas de aceptación, los cuales de la misma manera quedaron figurados en el documento presente.

# CONCLUSIONES GENERALES

## CONCLUSIONES GENERALES

A partir del desarrollo de la presente investigación se arribaron a las siguientes conclusiones:

- ✓ Se desarrolló un producto final que personaliza en tiempo real una imagen que se caracteriza por ser de fácil instalación, se ejecuta en un corto tiempo y se ajusta a un estándar de comercialización, para la Plataforma de Seguridad en las Tecnologías de la Información del centro Telemática de la Universidad de las Ciencias Informáticas.
- ✓ El análisis de las herramientas similares permitió la identificación de nuevas soluciones partiendo de las características y potencialidades que ofrecen las tecnologías utilizadas en la investigación.
- ✓ Se demostró que las funcionalidades de la imagen brinda resultados satisfactorios que le otorgan a la investigación alto grado de validez.

Por todo lo anteriormente expuesto, se concluye que el objetivo propuesto para el presente trabajo de diploma se ha cumplido satisfactoriamente, poniendo en práctica todas y cada una de las tareas propuestas para el desarrollo de la Personalización de NOVA Server 2015 para el sistema PlatSI.

# RECOMENDACIÓN

## RECOMENDACIÓN

A partir de la personalización realizada a NOVA Server 2015 para el sistema PlatSI, se realice otras personalizaciones para este misma aplicación, pero en las distribuciones GNU/Linux: Debian y CentOS.

# REFERENCIAS BIBLIOGRÁFICAS

## REFERENCIAS BIBLIOGRÁFICAS

1. **Beit.** ServiciosTIC. [En línea] 2015. <http://www.serviciostic.com/las-tic/definicion-de-tic.html>.
2. **Universidad de las Ciencias Informáticas.** Universidad de las Ciencias Informáticas. [En línea] 2012. <http://www.uci.cu/?q=historia>.
3. **alegsa.com.ar. alegsa.com.ar.** [En línea] 1998 - 2015. <http://www.alegsa.com.ar/Dic/distribucion%20de%20linux.php>.
4. **About.com.** About en español. [En línea] 2015. <http://windowsespanol.about.com/od/RedesYDispositivos/f/Que-Es-ISO.htm>.
5. **Morales, Delis Ise.** *Integración de herramientas de pruebas de seguridad para aplicaciones web en XILEMA-PlatSI.* La Habana : Universidad de las Ciencias Informáticas, 2014.
6. **Fuentes, Allan Pierra.** *Nova, ditribución cubana de GNU/Linux.* La Habana : Universidad de las Ciencias Informáticas, 2011.
7. **Total Publishing Network S.A. Muy Linux.** [En línea] 2008. <http://www.muylinux.com/2010/07/21/reconstructor-crea-un-livecd-a-tu-medida>.
8. **Hernández, Luis , y otros.** Up to down. [En línea] <http://iso-master.uptodown.com/ubuntu>.
9. **Doñe, José.** *Creación de una Distro.*
10. **Alegsa.com.ar.** [En línea] 1998 - 2015. <http://www.alegsa.com.ar/Dic/debian.php>.
11. **Pérez, María T. Garzón.** *Sistemas gestores de bases de datos.* s.l. : Revista digital Innovación y experiencias educativas, 2010.
12. **Revista Cubana de Ciencias Informáticas.** Gracia Mora, Rider Oscar y Valera González, Yusleidy.
13. **Gallardo Concepción, Luis Eduardo y Carbonell Sánchez, Reynier.** *Módulo Base de Datos de Vulnerabilidades de Xilema-PlatSI.* Ciudad de La Habana : Universidad de las Ciencias Informáticas, 2014.
14. **doc.ubuntu-es. doc.ubuntu-es.** [En línea] junio de 2012. [http://doc.ubuntu-es.org/Manual\\_de\\_bash](http://doc.ubuntu-es.org/Manual_de_bash).
15. **Fabiola, Ticon Condori Shirley.** *Metodologías tradicionales, metodologías ágiles, metodologías para juegos, metodologías educativas, y metodologías para aplicaciones móviles.*

# REFERENCIAS BIBLIOGRÁFICAS

16. proyectos agiles.org. [En línea] <http://www.proyectosagiles.org/que-es-scrum>.
17. Penadés, M<sup>a</sup> Carmen y Letelier, Patricio . *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*.
18. Peñalver, G., Meneses, A. y García, S. *SXP, Metodología Ágil para el desarrollo de software*. Ciudad de La Habana, Cuba : Universidad de las Ciencias Informáticas, 2010.
19. Llanes, Daniel Sánchez. SGIEPC. Sistema de gestión de informática en la empresa procesadora de café Eladio Machín. Cienfuegos, Cuba : Universidad de Cienfuegos “Carlos Rafael Rodríguez”, 2008.

# BIBLIOGRAFÍAS

## BIBLIOGRAFÍA

1. *Guía de instalación de Debian GNU/Linux. 2004-2015.*
2. Doñe, José. *Creación de una Distro.*
3. Fabiola, Ticon Condori Shirley. *Metodologías tradicionales, metodologías ágiles, metodologías para juegos, metodologías educativas, y metodologías para aplicaciones móviles.*
4. Fuentes, Allan Pierra. *Nova, distribución cubana de GNU/Linux.* La Habana : Universidad de las Ciencias Informáticas, 2011.
5. Gallardo Concepción, Luis Eduardo y Carbonell Sánchez, Reynier. *Módulo Base de Datos de Vulnerabilidades de Xilema-PlatSI.* Ciudad de La Habana : Universidad de las Ciencias Informáticas, 2014.
6. Hitpass, Bernhard. *BPMN 2.0 Manual de Referencia y Guía Práctica.* Santiago de Chile : s.n., 2014.
7. Morales, Delis Ise. *Integración de herramientas de pruebas de seguridad para aplicaciones web en XILEMA-PlatSI.* La Habana : Universidad de las Ciencias Informáticas, 2014.
8. Penadés, M<sup>a</sup> Carmen y Letelier, Patricio . *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP).*
9. Peñalver, G., Meneses, A. y García, S. *SXP, Metodología Ágil para el desarrollo de software.* Ciudad de La Habana, Cuba : Universidad de las Ciencias Informáticas, 2010.
10. Pressman, Roger S. *Ingeniería de software: un enfoque práctico.* s.l. : Mikel Angoar, 1997.
11. Sisinta, Ailema. *Propuesta de técnicas de estimación y métricas para la metodología ágil SXP.* [En línea] 2009.
12. Press, Krasis. *Guía de arquitectura N-Capas orientada al dominio con. NET 4.0.* 2010.
13. Mora Ferrer, Joan. *Instalar Debian GNU/Linux a SGI Indy.* 2004.
14. Hernández Meléndez, Edelsys. *Cómo escribir una tesis.* La Habana : Ciencias Médicas, 2006.

# ANEXO I

15. *Delfdroid y su comparación evaluativa con XP y Scrum mediante el método 4-DAT*. Ávila Domenech, Ernesto y Meneses Abad, Abel. s.l. : Revista Cubana de Ciencias Informáticas, 2013.
16. HENRÍQUEZ, Santiago Domingo Moquillaza; HUERTA, Hugo Vega; GRADOS, Luis Angel Guerra. Programación en N capas. Revista de investigación de Sistemas e Informática, 2010, vol. 7, no 2, p. 57-67.
17. TENTOR, Julio. Arquitectura de N-Capas y N-Niveles.[Imagen]. Obtenido de: <http://www.jtentor.com.ar/post/Arquitectura-de-N-Capas-yN-Niveles.aspx>.
18. DONÉSTEVEZ, Abel Alfonso Fírvida. Nova 3.0, avances y expectativas de la distribución cubana de GNU/Linux. Serie Científica, 2011, vol. 4, no 6.
19. LEYVA-OCHOA, Yoency, et al. Plataforma Cubana de Migración a Código Abierto. Revista Cubana de Ciencias Informáticas, 2014, vol. 8, p. 78-91.
20. PINO MOSCOSO, Verónica Alejandra. Análisis de Alternativas de Distribuciones Linux para Personalizar un Live CD para le EIS de la ESPOCH. 2011.
21. DE LINUX, EL Rincón. Distribuciones de Linux. Recuperado el 2 de Noviembre de 2011. 2011.
22. WHITE, Stephen A. Introduction to BPMN. IBM Cooperation, 2004, vol. 2, no 0, p. 0.
23. FOX, B.; RAMEY, C. BASH (1). páginas de manuales de BASH v2. 5b, vol. 2.
25. RIVERO, Gonzalo. Introduccion a la programacion en bash.
26. CHÁVEZ, Jorge Domínguez. Más sobre BASH en Linux.
28. KNIBERG, Henrik. Scrum y XP desde las trincheras. C4Media Inc. InfoQ, 2007.
29. LOZANO, Guillermo Valdés. Software libre. 2008.
30. FOUCES, Oscar Díaz. Creación de un entorno de trabajo para la formación de profesionales de los servicios lingüísticos con GNU/Linux.. Hermeneus: Revista de la Facultad de Traducción e Interpretación de Soria, 2007, no 9, p. 21-38.
31. MOLINA, Jesús, et al. Ulanux/ulanix: Software académico a la medida. Primer encuentro venezolano sobre tecnologías de la información e ingeniería del software, evetis' 07, 2007.

# ANEXO I

## ANEXO I

### Manifiesto Ágil

1. La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de SW que le aporte un valor.
2. Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva.
3. Entregar frecuentemente software que funcione desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas.
4. La gente del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.
5. Construir el proyecto en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir finalizar el trabajo.
6. El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.
7. El software que funciona es la medida principal de progreso.
8. Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.
9. La atención continua a la calidad técnica y al buen diseño mejora la agilidad.
10. La simplicidad es esencial.
11. Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos.
12. En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo, y según esto ajusta su comportamiento

Figura 11. Manifiesto Ágil de las Metodologías Ágiles.

# GLOSARIO DE TÉRMINOS

## ANEXO II

Entrevista realizada al Ing. Luís Eduardo Gallardo Concepción, Centro TLM, departamento de aplicaciones, desarrollador principal de PlatSI:

### **¿Cuáles son las dependencias que utiliza PlatSI para su correcto funcionamiento?**

Las dependencias que utiliza PlatSI son:

- php-zero-ice
- python-zero-ice
- php5, php5-gd, php-apc, php5-pgsql, php-mdb2-driver-pgsql
- apache2, libapache2-mod-php5
- postgresql
- python-psycopg2
- sqlite, python-dev y nikto

Estas dependencias se descargan desde el repositorio y ellas a su vez dependen de otras.

### **¿Cuáles son los componentes de PlatSI y cómo funcionan entre ellos?**

PlatSI está compuesto por tres componentes: PlatSI (Aplicación Web), Bambu-Server y Bambu-Client, de forma resumida se realiza una petición de prueba en PlatSI (Aplicación Web), este le envía la petición al Bambu-Server para que la procese y le envíe dicha petición al Bambu-Client, una vez llegada la petición Bambu-Client, este realiza las pruebas y le envía el resultado al servidor de Bambu que es el encargo de mandar estos resultados a PlatSI (Aplicación web) para mostrárselos al usuario.

### **¿Cuáles son los requerimientos mínimos para que funcione PlatSI en una PC?**

PlatSI no ha sido probado en PCs con prestaciones menores a las siguientes: Microprocesador con 3.0 GHz de velocidad, memoria Ram de 2 Gb y un disco duro de 80 Gb.

# GLOSARIO DE TÉRMINOS

## GLOSARIO DE TÉRMINOS

**Software:** Conjunto de programas y rutinas que permiten a la computadora realizar determinadas tareas.

**Consultoría:** Entidad dedicada a asesorar empresas en asuntos como el marketing, la organización, la fiscalidad, etc.

**Accesibilidad:** Posibilidad de acceder a cierta cosa o facilidad para hacerlo.

**Usabilidad:** Cualidad de la página web o del programa informático que son sencillos de usar porque facilitan la lectura de los textos, descargan rápidamente la información y presentan funciones y menús sencillos, por lo que el usuario encuentra satisfechas sus consultas y cómodo su uso.

**Distribución:** Acción y efecto de distribuir (dividir algo entre varias personas, dar a algo el destino conveniente, entregar una mercancía). El término, que procede del latín *distributio*, es muy habitual en el comercio para nombrar al reparto de productos.

**Dependencias:** Las dependencias de software se utilizan para identificar relaciones entre partes de software. La información de dependencia se define como *requisitos* y *capacidades*.

**Distro:** Coloquialmente son las distribuciones de GNU/Linux. Son distribuidoras independientes de material alternativo.

**Código abierto:** Expresión con la que se conoce al software o hardware distribuido y desarrollado libremente. Se focaliza más en los beneficios prácticos (acceso al código fuente) que en cuestiones éticas o de libertad que tanto se destacan en el software libre.

**Servidor:** Aplicación en ejecución (software) capaz de atender las peticiones de un cliente y devolverle una respuesta en concordancia.

**Sostenible:** Que es compatible con los recursos de que dispone una región, una sociedad, etc.

# GLOSARIO DE TÉRMINOS

**Personalización:** Adaptar algo a las características, al gusto o a las necesidades de una persona: Juan ha personalizado totalmente su despacho. Dar carácter personal a una cosa.

**Hardware:** Conjunto de elementos físicos o materiales que constituyen una computadora o un sistema informático.

**Shell de Unix:** es el término usado en informática para referirse a un intérprete de comandos, el cual consiste en la interfaz de usuario tradicional de los sistemas operativos basados en Unix y similares como GNU/Linux.

**Paquetes:** Los paquetes informáticos son programas que permiten a los usuarios realizar diversos trabajos en la automatización de tareas complicadas como la contabilidad, diseño gráfico y/o finanzas, diferenciándose principalmente de los sistemas operativos por las utilidades y el lenguaje de programación que utilizan.

**Script:** En informática un script, archivo de órdenes, archivo de procesamiento por lotes o guiones un programa usualmente simple, que por lo regular se almacena en un archivo de texto plano.

**Live CD:** Es un sistema operativo que normalmente contiene un conjunto de aplicaciones almacenado en un medio extraíble, que puede ser un CD o un DVD, que puede ejecutarse desde éste sin necesidad de instalarlo en el disco duro de una computadora, para lo cual usa la memoria RAM como disco duro virtual y el propio medio como sistema de archivos.

**BPMN:** Del inglés Business Process Modeling Notation, en español Notación para el Modelado de Procesos de Negocio, se utilizó para modelar el proceso de negocio en la detección de vulnerabilidades.

**Vulnerabilidades:** Son puntos débiles del software que permiten que un atacante comprometa la integridad, disponibilidad o confidencialidad del mismo.