Universidad de las Ciencias Informáticas

Facultad 1

Herramienta para automatizar la instalación cifrada de GNU/Linux Nova

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor

Alejandro Guilarte Larin

Tutores

MSc. Anié Bermudez Peña

Ing. Yennis Sánchez Figueredo

La Habana, abril del 2020

Declaración de autoría

Declaro por este medio que yo Alejandro Guilarte Larín, con carné de identidad 96110910028 soy el autor principal del trabajo titulado Herramienta para automatizar la instalación cifrada de GNU/Linux Nova y autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

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

_____

Firma del autor

Alejandro Guilarte Larín

_____ _____

Firma de la tutora Firma de la tutora

Msc. Anié Bermudez Peña Ing. Yennis Sánchez Figueredo

Datos de contacto

Msc. Anié Bermudez Peña

Universidad de las Ciencias Informáticas, La Habana, Cuba

Correo: abp@uci.cu

Ing. Yennis Sánchez Figueredo

Universidad de las Ciencias Informáticas, La Habana, Cuba

Correo: ysfigueredo@uci.cu

Dedicatoria:

Dedico este trabajo a toda mi familia, especialmente a los pilares de mi vida. A Daliena, mi hermanita pequeña-mayor va dedicado este trabajo, que a pesar de yo ser un indeseable, siempre encuentra la forma de hacerme llegar su amor y cariño. A mi madre Ania, la mejor madre del mundo, la madre de dios y del diablo, y a veces hasta de nadie, a ella va dedicado este trabajo. A mi padre Roberto; uno que no puede ser ni será nunca normal, ya que es el mejor padre que se puede pedir, a él va dedicado este trabajo.

Este trabajo va dedicado, a mis abuelos Orlinda y Constancio, y a mis primos Rosi y Eduardo, que no solo por sus esfuerzos gigantes, sino también constantes hoy logro este gran objetivo para mí. A ellos va dedicado este trabajo.

Agradecimientos:

Quiero agradecer a las personas que de una forma u de otra me apoyaron a lo largo de esta travesía, a mis padres y hermana que sin ellos no hubiera sido posible mi educación, sin ellos hubiese sido imposible convertirme en la persona que soy ahora.

A mis abuelos quiero agradecer por cada segundo de paciencia y enseñanza que me han regalado en la vida, cada sacrificio que han hecho por mí y por mi hermana.

A mis primos, esas personas que sé, que aun sin tener obligación de hacerlo, cuando me ha hecho falta la mínima cosa, son los primeros en bajarme las estrellas y estar aún más contentos que yo, de mi propia felicidad.

A mi tío Amir que aunque lejano físicamente siempre atento, y al decir de mi familia, no me podría parecer más a él.

Al resto de mi familia, a todos y cada uno que la conforman, que la hacen especial, única y perfecta.

Agradecer a esos amigos que desde la infancia va a mi lado, ellos con quien he compartido la dicha de cada día, Elías, Mosquera, Ernesto y Gabriel. Agradecerles por todas las historias y todo el apoyo en la vida.

A esos ahora viejos amigos, con que empezó la travesía hace ya 5 años, el grupo FI08. A los que siguen, a los que se quedaron, a los que le ha ido no tan bien, pero aun lo siguen intentando. Especialmente a mis compañeros de apartamento, por las horas juntos y los momentos inolvidables. Por las boberías, las penas y las alegrías que nos hicieron hermanos de fuego, por eso les agradezco.

A mis tutoras Yennis y Anié, que no sé cómo se las arreglaron, para escoger el peor tesista posible. Gracias por toda la ayuda que me han brindado a lo largo de este proceso.

A todas las personas que me faltan, pero que saben que hacen mi vida algo genial, a todos ustedes les estoy infinitamente agradecido.

Resumen

La Distribución Cubana GNU/Linux Nova es el sistema operativo seleccionado para ser desplegado en los Organismos de Administración Central del Estado por tener grandes capacidades de procesamiento y administración de sus recursos, manteniendo siempre como pilar importante la seguridad. Este sistema operativo, para su instalación y uso por diferentes entidades, en muchos casos necesita un cifrado de datos, que consume mucho tiempo debido a que se desarrolla de forma manual. El objetivo del presente trabajo de diploma es desarrollar una herramienta informática que permita automatizar el proceso de instalación cifrada de la distribución cubana GNU/Linux Nova. Para guiar el proceso de construcción de la propuesta de solución se utilizó la metodología de desarrollo de software Variación de AUP para la UCI y en la implementación se emplearon tecnologías y herramientas de software libre. La evaluación de la propuesta de solución se realizó mediante la aplicación de pruebas que garantizan el correcto funcionamiento de la herramienta y demostraron la satisfacción del cliente hacia el sistema desarrollado. Al culminar la presente investigación se obtuvo una herramienta informática que permite automatizar la instalación cifrada de la distribución cubana GNU/Linux Nova.

Palabras claves: cifrado, GNU/Linux Nova, herramienta informática, instalador, sistema operativo.

Abstract

The Cuban GNU / Linux Nova Distribution is the operating system selected to be deployed in the Central State Administration Bodies for having great capacities for processing and managing their resources, always keeping security as an important pillar. This operating system, for its installation and use by different entities, in many cases requires data encryption, which is time-consuming because it is developed manually. The objective of this diploma work is to develop a computer tool that allows automating the encrypted installation process of the Cuban GNU / Linux Nova distribution. To guide the process of building the solution proposal, the AUP Variation software development methodology for the UCI was used, and free software technologies and tools were used in the implementation. The evaluation of the solution proposal was carried out through the application of tests that guarantee the correct operation of the tool and demonstrated customer satisfaction towards the developed system. Upon completion of the present investigation, a computer tool was obtained that allows automating the encrypted installation of the Cuban GNU / Linux Nova distribution.

Keywords: computer tool, encryption, GNU / Linux Nova, installer, operating system.

Índice

INTRODUCCIÓN

Desde los orígenes del hombre, ha sido una cuestión importante el mantener en secreto información de otros de su especie. En la actualidad, con la era digital, la información toma un nuevo estatus de especial importancia. No solo sigue siendo primordial mantener la confidencialidad de la información, sino que se ha convertido en una tarea cada vez más difícil para usuarios de la tecnología(Ernesto Rodriguez Ortiz & George Sánchez Amaya, 2009).

Un sistema operativo es el software principal o conjunto de programas de un sistema informático. Gestiona los recursos de hardware y provee servicios a los programas de aplicación de software. Sus funciones básicas son administrar recursos, coordinar el hardware, organizar archivos y directorios en dispositivos de almacenamiento (Piñeiro, 2015).

GNU/Linux es un ejemplo de sistema operativo, se ha caracterizado por su estabilidad de operación, velocidad, seguridad como pilar importante y capacidad para la administración eficiente de los recursos de la computadora como son la memoria, el poder de la unidad central de procesamiento y espacio en disco (EPSILONMAG, 2017).

Estas características lo han llevado a diferenciarse del resto de los sistemas operativos del mundo, pero sin dudas, la más interesante de todas es que se compone de código abierto, es decir, su desarrollo se hace abiertamente y el código puede ser modificado y distribuido libremente. Esto ha propiciado que en la actualidad las distribuciones GNU/Linux hayan alcanzado una significativa popularidad ampliando su uso a varias esferas de la sociedad, convirtiéndose en una alternativa a nivel mundial (Debian, 2016).

El uso del Software Libre (SWL) representa para Cuba una herramienta fundamental para llevar a cabo la informatización de la sociedad, pues se lograría la soberanía tecnológica. De igual manera el bloqueo económico, financiero, comercial y tecnológico impuesto por el gobierno de Estados Unidos impide el desarrollo progresivo de la informatización, resultando una utopía para el país el uso o acceso legal a las aplicaciones tecnológicas o productos desarrollados por empresas norteamericanas. (NOVA, 2019).

En América Latina existen países como Venezuela, Brasil y Bolivia que han hecho del software libre y sus distribuciones, proyectos bandera en el desarrollo científico y tecnológico. Estos países coinciden ideológicamente en su oposición a la privatización, a favor de la autonomía tecnológica y el ahorro económico (Hipertextual, 2015).

Como parte del proceso de informatización de la sociedad cubana en el 2004 se publica por consenso del Consejo de Ministros de la República de Cuba el acuerdo 084, el cual orienta la migración paulatina de los Organismos de la Administración Central del Estado (OACE) hacia aplicaciones de código abierto. Un proceso que retomó fuerza tras el acuerdo del 23 de abril del año 2015, cuando la Comisión de Informatización y Ciberseguridad indicó el desarrollo de un diagnóstico de migración en varios OACE. Durante este proceso, se obtuvo la información sobre el hardware y software de las computadoras institucionales, dicha información dio paso al análisis de la misma y la elaboración de informes de diagnóstico que facilitan el posterior desarrollo de la ejecución del proceso de migración (Paz, 2019).

La Universidad de las Ciencias Informáticas (UCI) ha sido uno de los principales protagonistas en este proceso de migración. La UCI cuenta con varios proyectos en desarrollo, entre ellos el Centro de *Software* Libre (CESOL) encargado de desarrollar la distribución cubana GNU/Linux Nova y definir las directrices, lineamientos y soluciones que guiarán la migración nacional. Esta distribución está basada en cuatros principios: soberanía tecnológica, socio-adaptabilidad, sostenibilidad y seguridad; este último principio propone un modelo de desarrollo colaborativo que permita el acceso al código fuente y el exhaustivo proceso de revisión y auditoría de código para garantizar un sistema seguro (Aroche, Eugenio Durán 2014).

Mediante una entrevista realizada a especialistas del centro CESOL (ver Anexo 1), se evidencia que en la actualidad una de las principales barreras para mantener la seguridad en el proceso de migración, es que la distribución GNU/Linux Nova durante su instalación, permite cifrar solamente una porción, no así todo el disco duro.

Cifrar solo una porción del disco duro trae aparejado dos riesgos fundamentales: si le realizan un ataque informático, el atacante tendrá más información a su disposición que no estará cifrada y; en caso de robo o acceso físico al ordenador por algún individuo no autorizado será mucho más fácil recopilar la información contenida en este.

Si bien existen los métodos para cifrar completamente el disco y realizar la instalacion de dicho sistema operativo, existen dos incovenientes muy importantes; para realizarlo es necesario alto grado de conocimientos específicos y, que se realizaría de forma manual aumentando el coste en tiempo. En estos momentos en la distribución GNU/Linux Nova no existe una herramienta destinada a automatizar el proceso de instalación cifrada de dicho sistema operativo.

Partiendo de la problemática anterior se representa el siguiente **problema de la investigación:** ¿cómo garantizar la instalación cifrada de la Distribución Cubana GNU/Linux Nova de forma automática? Para

orientar la investigación se identifica como **objeto de estudio:** el cifrado de datos en las instalaciones de las Distribuciones GNU/Linux, y para enmarcar el alcance del mismo se cuenta con el **campo de acción:** el cifrado de datos en la instalación de la Distribución Cubana GNU/Linux Nova.

Para realizar la investigación se trazó como **objetivo general:** Desarrollar una herramienta informática que permita automatizar la instalación cifrada de la Distribución Cubana GNU/Linux Nova.

El objetivo general se desglosa en los **objetivos específicos:**

1. Elaborar el marco teórico-metodológico para la propuesta de desarrollo de la herramienta informática para automatizar la instalación cifrada de la Distribución Cubana GNU/Linux Nova.
2. Realizar el análisis y diseño de la herramienta informática para automatizar la instalación cifrada de la Distribución Cubana GNU/Linux Nova.
3. Implementar la herramienta informática para automatizar la instalación cifrada de la Distribución Cubana GNU/Linux Nova.
4. Evaluar la herramienta informática para automatizar la instalación cifrada de la Distribución Cubana GNU/Linux Nova.

Se definen como **preguntas científicas:**

1. ¿Cuáles son los principales referentes teórico-metodológicos que sustentan el desarrollo de la herramienta informática para la automatización de la instalación cifrada de la Distribución Cubana GNU/Linux Nova?
2. ¿Cuáles metodologías, tecnologías y herramientas, conforman el ambiente de desarrollo de la herramienta informática para la automatización de la instalación cifrada de la distribución cubana GNU/Linux Nova?
3. ¿Cómo generar los artefactos a partir del análisis y diseño de la herramienta informática para la automatización de la instalación cifrada de la distribución cubana GNU/Linux Nova?
4. ¿Qué pruebas aplicar en la evaluación de la herramienta informática para la automatización de la instalación cifrada de la distribución cubana GNU/Linux Nova?

Para el desarrollo de la investigación se utilizaron los siguientes **métodos de la investigación científica:**

Métodos teóricos:

- **Analítico-Sintético:** se utiliza para la realización del estudio y el análisis de un grupo de herramientas de cifrado en sistemas operativos GNU/Linux, posibilitando la comprensión de sus características generales.

- **Deductivo-Inductivo:** permite llevar a cabo un procedimiento que a partir de conocimientos generales se concluyeron casos particulares por un razonamiento lógico.

- **Modelación:** se utiliza para modelar los diagramas que, permitan una mejor comprensión de la temática, a lo largo de las etapas de análisis, diseño e implementación en la construcción de la herramienta para la automatización de la instalación cifrada de la distribución cubana GNU/Linux Nova.

- **Histórico Lógico:** permite conocer la evolución de las herramientas para la automatización de la instalación cifrada de la distribución cubana GNU/Linux Nova, así como los elementos que la caracterizan.

Métodos Empíricos:

- **Observación:** se utiliza para analizar cómo se realiza actualmente el cifrado en una instalación de la Distribución Cubana GNU/Linux Nova, obteniendo conocimientos que serían usados en la implementación.

- **Entrevista:** se utiliza para adquirir conocimientos importantes sobre las características del proceso de cifrado en instalaciones de la Distribución Cubana GNU/Linux Nova.

El presente trabajo está compuesto por: resumen, introducción, tres capítulos, conclusiones, recomendaciones, referencias bibliográficas y anexos.

Capítulo1. Fundamentación teórica de la herramienta informática para la automatización de la instalación cifrada de la distribución cubana GNU/Linux Nova: se presentan los conceptos fundamentales, la definición de la metodología de desarrollo de software que se utiliza y se realiza la descripción de la propuesta de solución. Además se describen las principales herramientas que se utilizan, lenguajes de programación y tecnologías informáticas con que se implementa la herramienta informática para la automatización de la instalación cifrada de GNU/Linux Nova.

Capítulo2. Análisis y Diseño de la herramienta informática para la automatización de la instalación cifrada de la distribución cubana GNU/Linux Nova: se realiza el análisis y diseño de la propuesta de solución haciendo uso de la metodología de desarrollo seleccionada. Además se especifica los requisitos funcionales y no funcionales que requieren la herramienta para su desarrollo y la arquitectura utilizada.

Capítulo3. Implementación, pruebas y evaluación de la herramienta informática para la automatización de la instalación cifrada de la distribución cubana GNU/Linux Nova: se documenta el proceso de implementación de la herramienta informática para la automatización de la instalación cifrada de GNU/Linux Nova a partir de los resultados del Análisis y Diseño, además de elaborar y aplicarle las pruebas al sistema.

CAPÍTULO1. FUNDAMENTACIÓN TEÓRICA DE LA HERRAMIENTA INFORMÁTICA PARA AUTOMATIZAR LA INSTALACIÓN CIFRADA DE LA DISTRIBUCIÓN CUBANA GNU/LINUX NOVA.

1.1 Introducción

En el presente capítulo se presentan los conceptos fundamentales al tema en cuestión, el estudio sobre las principales herramientas existente que permiten hacer una instalación cifrada de la distribución GNU/Linux, así como la metodología de desarrollo de software a utilizar y las distintas herramientas y tecnologías que serán empleadas en el desarrollo de la solución.

1.2 Conceptos y definiciones asociados al dominio del problema

Distribución de GNU/Linux

Según el autor del libro *The Debian administrator's Handbook* de Raphael Hertzog y otros autores donde plantean que: un sistema operativo basado en una distribución de GNU/Linux puede definirse como 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. Las mismas pueden contener o no controladores privativos (Hertzog, et al., 2015).

Por lo anteriormente plateado se toma como concepto en la presente investigación que una distribución de GNU/Linux es el conjunto de pasos y modificaciones realizadas a una distribución para agregar, quitar o modificar características al sistema dependiendo de la necesidad y los usuarios finales.

Instalación

Instalación es el acto y la consecuencia de instalar: establecer, situar algo en el sitio debido. El término también puede aludir al conjunto de los elementos instalados y al espacio que dispone de todo lo necesario para el desarrollo de una determinada actividad. En el terreno de la informática, por último, la noción de instalación refiere a transferir un software a un ordenador y a prepararlo para que funcione correctamente (Capril, 2017).

En la presente investigación se asume como concepto de instalación como el proceso fundamental por el cual los nuevos programas son transferidos a un computador con el fin de ser configurados, y preparados para ser desarrollados,teniendo como sujeto de la instalación un sistema operativo de distribución GNU/Linux.

Ciberseguridad

Es la práctica de defender las computadoras y los servidores, los dispositivos móviles, los sistemas electrónicos, las redes y los datos de ataques maliciosos. También se conoce como seguridad de tecnología de la información o seguridad de la información electrónica. El término es amplio y se aplica a numerosos elementos, desde seguridad informática hasta recuperación ante desastres y educación del usuario final (Karspesky 2019).

Cifrado de datos

Es el proceso por el que una información puede pasar a ser ilegible o secreta, a través de un algoritmo informático. Por tanto, los datos cifrados que sean enviados a un destinatario serán mucho menos vulnerables y tendrán menos riesgos de ser interceptados por terceros. La persona o empresa destinataria puede volver a descifrar el mensaje recibido a través de una clave («Cifrado de datos. Qué es y cómo hacerlo» 2018).

Instalación Cifrada

Es la acción de instalar un programa informático, en un sistema o dispositivo que cumple con los requisitos de cifrado de la información contenida y/o a contener, garantizando la integridad y confidencialidad del proceso de instalación («Cifrado de datos. Qué es y cómo hacerlo» 2018).

En la presente investigación se asume como concepto de instalación cifrada como el proceso en el que se realiza la instalación de un sistema informático en un dispositivo de almacenamiento el cual ha sufrido previamente un proceso de cifrado.

1.3 Análisis de sistemas homólogos

Hoy en día existen un gran número de aplicaciones que hacen posible el cifrado de la información y que son accesibles a todos los usuarios, desde herramientas libres hasta propietarias, con interfaces gráficas, sin ellas y que se pueden utilizar en diferentes sistemas operativos, desde las diferentes distribuciones de GNU/Linux, hasta Windows y Mac.

Después de realizar una amplia búsqueda de las mejores y más usadas herramientas de cifrado, fue seleccionada una muestra formada solamente por las herramientas de cifrado de código abierto, con licencias libres como GPL y que funcionen sobre GNU/Linux, debido a que el país se encuentra en un proceso de migración al software libre y no es factible desarrollar una solución con herramientas propietarias. Además estas herramientas libres se encuentran a disponibilidad de todos y se entregan con el código fuente de las mismas, evitando así que existan puertas traseras o fallos de seguridad que hayan

pasado por alto los desarrolladores, ya que todo aquel que tenga conocimientos para hacerlo, puede revisar el código, cambiarlo y redistribuirlo.

Debido a la manera intencional con que se escogió la muestra, siguiendo criterios específicos, pudo haberse pasado por alto alguna herramienta que no cumpliera con los requisitos anteriores y que sin embargo fuera superior en cuanto a funcionalidades.

A continuación se realiza una descripción de las diferentes herramientas seleccionadas en la muestra con el objetivo de determinar cuál sería más factible usar en el desarrollo de la solución.

1.3.1 Loop-AES (Advanced Encryption Standard/Estándar de Encriptación Avanzada)

Es un esquema de cifrado por bloques adoptado como un estándar de cifrado por el gobierno de los Estados Unidos, creado en Bélgica. El AES fue anunciado por el Instituto Nacional de Estándares y Tecnología (NIST) de los Estados Unidos el 26 de noviembre de 2001 después de un proceso de estandarización que duró 5 años. Se transformó en un estándar efectivo el 26 de mayo de 2002. Desde 2006, el AES es uno de los algoritmos más populares usados en criptografía simétrica. (Mahiques Carrasco, 2013).

Utiliza hasta 65 llaves distintas, las cuales crea añadiendo información aleatoria. Cada llave cifra un sector determinado de unos 512 bytes que se va repitiendo continuamente. Este modelo de llaves múltiples se denomina multi-key-v3 (Multi-llaves versión 3) (Mahiques Carrasco, 2013).

Otras características («Loop-aes vs DM-crypt» 2017)**:**

- Las llaves por la forma en que las genera y usa, hace que el sistema sea inmune a los ataques de diccionario.
- Para aumentar la seguridad al máximo se debe portar un fichero con las 65 llaves siempre encima, por ejemplo una memoria USB, lo que realmente hace que se pierda la seguridad ganada, al portar uno mismo las claves que deberían estar almacenadas en un lugar seguro.

- Cuando se hiberna o se suspende el equipo las claves que están siendo utilizadas en el momento y que se encuentran en la memoria RAM se copian al disco para poder reanudar el sistema después, lo que hace inseguros y poco recomendados los mecanismos de hibernación o suspensión.

1.3.2 Dm-crypt / LUKS

LUKS de sus siglas en ingles *"Linux Unified Key Setup" (configuración de la llave unificada de LINUX),* es un estándar formal implementado por la herramienta Cryptsetup-LUKS en el cual se eliminan los problemas de la segregación de los datos de DMCrypt/Cryptsetup. Este último es una versión del Cryptsetup original. Las principales diferencias entre los dos están dadas por la generación de claves y por la manera en que LUKS protege la clave maestra con la que cifra los datos en la partición. («Loop-aes vs DM-crypt» 2017).

LUKS añade los parámetros que necesita Cryptsetup-LUKS para generar la clave desde una contraseña introducida por el usuario a la cabecera de la partición cifrada. Cada clave contiene una copia cifrada de la clave maestra que DM-Crypt utiliza para proteger los datos. («Loop-aes vs DM-crypt» 2017).

1.3.3 eCryptfs

Es un sistema de cifrado de archivos para GNU/Linux que aprovecha el servicio de llavero recientemente introducido en el núcleo, la API (Application Programming Interface/ interfaz de programas de aplicación) de cifrado del núcleo CryptoAPI («eCryptfs in Launchpad» 2016).

eCryptfs es único entre la mayoría de las soluciones de cifrado de sistema de archivos ya que en él almacena un conjunto completo de los metadatos de cifrado, junto con cada archivo de manera individual. Esto permite que los archivos cifrados se puedan transferir a través de dominios de confianza, manteniendo la capacidad de que las personas con las credenciales adecuadas puedan acceder a esos archivos. Debido a que el cifrado y el descifrado se realiza en la capa intermedia, el proceso se hace transparente desde la perspectiva de la aplicación («eCryptfs in Launchpad» 2016).

La principal desventaja de esta herramienta es que al ser un sistema de cifrado de archivos en lugar de un sistema de cifrado por bloques, los metadatos se mantienen visibles: número de ficheros cifrados, permisos de los ficheros, tamaño, fecha de modificación («eCryptfs in Launchpad» 2016).

1.3.4 Scramdisk4Linux

SD4L es un conjunto de herramientas de Linux y una interfaz gráfica de usuario (GUI) que permite la creación y el acceso a archivos contenedores cifrados de ScramDisk. Los contenedores ScramDisk4Linux constan de una cabecera y el volumen cifrado que incluye el sistema de archivos y el contenido de todos los archivos. En ambos formatos la cabecera y el volumen son indistinguibles y sin el conocimiento de la contraseña no se podrían diferenciar de un conjunto de números aleatorios por lo que en estos contenedores no se podrían separar archivos, particiones o medios de almacenamiento («SD4L - ScramDisk for Linux» 2013).

Resultados del estudio de sistemas homólogos

Con respecto a las herramientas restantes se realizó un análisis con el objetivo de conocer si cumplen con las necesidades existentes en la creación de personalizaciones para la distribución de GNU/Linux NOVA.

Criterios de Análisis: se escogen como criterios los mostrados posteriormente porque los mismos forman parte de las necesidades que se requieren para el desarrollo de la propuesta de solución.

- Tipo de licencia: especifica el tipo de licencia de uso y distribución que posee dicho software.
- Autenticación pre-arranque: determina si la herramienta proporciona un sistema de autenticación puede ser requerido antes de arrancar el ordenador, lo que permite cifrar el disco de arranque.
- Cifrado de disco completo: determina si la herramienta permite el cifrado del disco duro en su totalidad.
- Necesidad de dispositivo externo**:** determina si la herramienta requiere la utilización de un dispositivo externo para garantizar la seguridad de la información.
- Instalación automática de sistema operativo: determina si la herramienta cuenta con la opción de realizar la instalación automática de un sistema operativo.

A continuación, se muestra la Tabla 1 Resultados del análisis de las herramientas con distribuciones de GNU/Linux.

Tabla 1: Resultados del análisis de las herramientas con distribuciones de GNU/Linux

(Fuente: elaboración propia)

	Herramientas Seleccionadas			
Criterios de Análisis	**Loop-AES**	**Dm-crypt / LUKS**	**eCryptfs**	**Scramdisk4Linux**
Tipo de licencia	Libre	Libre	Libre	Libre
Autenticación Pre-Arranque	Sí	Sí	Sí	No
Cifrado de disco Completo	Sí	Sí	No	Sí

Necesidad de dispositivo externo	Sí	No	No	No
Instalación automática de sistema operativo	No	No	No	No

Una vez concluido el estudio de las aplicaciones existentes, se determina que las mismas no cumplen con las características necesarias para dar solución al problema planteado, sino que cumplen estas parcialmente. Esto se debe, a que si bien la herramienta Dm-crypt / LUKS cumple con la mayoría de los criterios, aún carece de uno importante; que permita dentro de sus opciones la instalación automática de un sistema operativo. Finalmente se decide reutilizar la herramienta Dm-crypt / LUKS al proporcionar los mejores resultados para ajustarlo a las necesidades del proyecto en cuestión

1.4 Metodología de desarrollo de software

Una metodología de desarrollo de software es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de introducción. (OBSBUSINESS, 2019) Además cuenta con factores como los costes, la planificación, la calidad y las dificultades asociadas que se manejan en el proceso que se suele seguir para diseñar una solución o un programa específico (Piñeiro, 2015).

La metodología seleccionada para el desarrollo de la propuesta de solución es una variación de la metodología “Proceso Unificado Ágil” (AUP) en unión con el modelo CMMMI-DEV fue creada por la Universidad de las Ciencias Informáticas denominada AUP-UCI, (León, 2017) la cual se adapta a las características de cada proyecto de desarrollo de dicha universidad, y se ajusta al ciclo de vida definido para la actividad productiva. Es una metodología flexible que no requiere de una gran cantidad de desarrolladores. Contiene un flujo de trabajo que cuenta de tres fases: (Rodríguez, 2015)

- **Inicio:** durante el inicio se llevan a cabo todos los planes del proyecto, recursos humanos, cronograma de tareas, plan de riesgo. Se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.

- **Ejecución:** en esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos, la arquitectura y el diseño, además de implementarse el producto.

- **Cierre:** en el cierre se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

El presente trabajo de diploma se centra en la fase de Ejecución y transitará por las siguientes disciplinas propuestas en la metodología: Modelado del negocio, Requisitos, Análisis y Diseño, Implementación, Pruebas internas y Pruebas de aceptación. Los productos generados por la investigación y por lo tanto en la propuesta de solución estarán basados en el Escenario No 4. Para esta elección se tuvo en cuenta principalmente que no se realiza modelado del negocio puesto que este no está bien definido, y que la solución a desarrollar no consta de un proyecto muy extenso lo que permite que las historias de usuario no posean demasiada información y sean lo más concretas posible.

1.5 Lenguajes y herramientas para el modelado de la propuesta de solución

En el modelado de la propuesta de solución se utilizaron los siguientes lenguajes y herramientas:

1.5.1 Lenguajes de Modelado

UML 2.0 (Lenguaje de Modelado Unificado) es el lenguaje de modelado gráfico utilizado para visualizar y documentar un sistema. Posibilita un estándar para describir los modelos, incluyendo aspectos conceptuales como procesos de negocio, funciones del sistema, expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables (UML, 2017).

Se utiliza a lo largo de todo el proceso, para establecer la serie de requerimientos, estructuras, ya artefactos necesarios para plasmar un sistema de software previo al proceso intensivo de escribir código.

1.5.2 Herramienta de modelado

***Visual Paradigm* 8.0 UML,** es una herramienta CASE (Ingeniería de Software Asistida por Computadora) multiplataforma, se define como herramienta para el modelado porque permite la representación de las etapas por las que trasciende un producto de software. Ayuda a una rápida construcción de aplicaciones de calidad y con el menor costo posible. Permite dibujar todos los tipos de diagramas de clases, generar código desde diagramas y desarrollar información (Cabriales, 2013).

Esta herramienta se selecciona por las facilidades que brinda para capturar los requisitos correctos y transformarlos en diseños precisos, además se utiliza su capacidad para diseñar cada uno de los diagramas y/o artefactos que propone UML para el análisis y diseño, implementación, pruebas y despliegue de software.

1.5.3 Herramientas y tecnologías de implementación

En el desarrollo de la propuesta de solución se utilizaron las siguientes herramientas y tecnologías.

1.5.4 Lenguaje de programación

Un lenguaje de programación es un lenguaje natural o artificial que le proporciona a una persona, la capacidad de escribir una serie de instrucciones o secuencias de órdenes en forma de algoritmos con el fin de controlar el comportamiento físico y lógico de una computadora (Rockcontent, 2019).

Bash Scripting:

Bash es un intérprete de órdenes que generalmente se ejecuta en una ventana de texto donde el usuario escribe órdenes en modo texto. Bash también puede leer y ejecutar órdenes desde un archivo, llamado guion o 'script'. Al igual que todos los intérpretes de Unix, es compatible con el agrupamiento de nombres de archivo (coincidencia de comodines), tuberías, sustitución de comandos, variables y estructuras de control para pruebas de condición e iteración. Las palabras reservadas, la sintaxis, las variables de ámbito dinámico y otras características básicas del lenguaje se copian de shell. Para realizar la implementación de la propuesta de solución se decide utilizar como lenguaje de programación Bash porque es un lenguaje de programación reutilizable, y posibilita realizar las configuraciones deseadas y así llegar a la personalización de su sistema operativo, en este caso Nova como distribución cubana de GNU/Linux.

1.5.5 Entorno de Desarrollo Integrado

Un entorno de desarrollo integrado o entorno de desarrollo interactivo (IDE), es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software. Consiste en un editor de código fuente, herramientas de construcción automáticas y un depurador (SOFTWAREPROG, 2018).

El límite entre un IDE y otras partes del entorno de desarrollo de software más amplio no está bien definido. Muchas veces, a los efectos de simplificar la construcción de la interfaz gráfica de usuario (GUI, por sus siglas en inglés) se integran un sistema controlador de versión y varias herramientas. Muchos IDE modernos también cuentan con un navegador de clases, un buscador de objetos y un diagrama de jerarquía de clases, para su uso con el desarrollo de software orientado a objetos (FERGARCIA, 2013).

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux Y macOs. Incluye soporte para la depuración, control integrado de GIT, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Es gratuito y de código abierto (SPRINGER, 2018).

Se utiliza como IDE para el desarrollo de la propuesta de solución porque ofrece como ventajas:

- Resaltado de sintaxis: capacidad para el tratamiento de textos, para diferencias elementos de textos mediante diversos colores o estilos tipográficos, dependiendo de las categorías sintácticas de sus términos.
- Depuración: identifica y corrige errores.
- *Snippet:* pequeñas partes reusables de código fuente, de máquina o texto.
- Refactorización: ofrece la posibilidad de reestructurar un código fuente, alterando su estructura interna sin cambiar su comportamiento externo.

1.5.6 Control de versiones

Un sistema de control de versiones es la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo. Una versión, revisión o edición de un producto, es el estado en que se encuentra el mismo en un momento dado de su desarrollo o modificación.

El utilizado en la propuesta de solución es el Control de Versiones de Software **GIT** el cual cuenta con características como (Paz, 2017):

- Fue diseñada por Linus Torvalds.
- No depende de un repositorio central.
- Es de software libre.
- Se puede llevar a cabo un historial completo de versiones.
- Posibilita que el usuario pueda visualizar las revisiones de código y desplazarse por el de manera ágil.
- Posee un sistema de trabajo con ramas, las cuales están destinadas a provocar proyectos divergentes de un proyecto final, brindando la posibilidad de realizar experimentos o probar nuevas funcionalidades.

1.6 Conclusiones del Capítulo

Con el estudio de los principales conceptos asociados al dominio del problema se logró obtener una mejor interpretación del mismo.

La comparación de los diferentes sistema de cifrados analizadas permitió seleccionar el idóneo que se debería utilizar para el desarrollo de una herramienta para automatizar la instalación cifrada de GNU/Linux Nova.

Para la propuesta de solución se determina el uso de Variación de AUP para la UCI como metodología de desarrollo, *Visual Paradigm* como herramienta para el modelado, Bash Scripting como lenguaje de programación, *Visual Studio Code* como entorno de desarrollo y *Git como control de versiones.*

CAPÍTULO 2: ANÁLISIS Y DISEÑO DE LA HERRAMIENTA INFORMÁTICA PARA AUTOMATIZAR LA INSTALACIÓN CIFRADA DE LA DISTRIBUCIÓN CUBANA GNU/LINUX NOVA.

Introducción

En el presente capítulo se exponen las principales características de la propuesta de solución a desarrollar. Como parte de la disciplina de Requisitos definida por la metodología de desarrollo AUP-UCI se realiza la captura de los requisitos funcionales y no funcionales, y se describen las historias de usuarios correspondientes a estos. Por último, como parte de la disciplina Análisis y Diseño se presenta la arquitectura de la solución, los patrones de diseño utilizados y el diagrama de clases del diseño.

2.1 Propuesta de solución

Se plantea como propuesta de solución una herramienta informática que de forma automática realice un cifrado total al disco duro donde se instalará el sistema operativo, empleando la herramienta de cifrado Dm-crypt / LUKS y sobre este disco duro ya cifrado se realice la instalación de la distribución cubana GNU/Linux Nova.

2.2 Requisitos

Un requisito es simplemente una declaración abstracta de alto nivel de un servicio que debe proporcionar el sistema o una restricción de éste. En el otro extremo, es una definición formal y detallada de una función del sistema (Sommerville, 2005). El objetivo principal en la disciplina Requisitos es desarrollar un modelo del sistema que se va a construir. Esta disciplina comprende la administración y gestión de los requisitos funcionales y no funcionales del producto (Sánchez, 2015).

2.2.1 Fuentes de obtención de requisitos

Las fuentes de obtención de requisitos utilizadas fueron:

- Análisis de las herramientas existentes (Ver epígrafe 1.3)
- Especialistas de CESOL.

2.2.2 Técnicas de identificación de requisitos

Las técnicas de identificación de requisitos de software permiten identificar las necesidades de negocio de los clientes y los usuarios. Son mecanismos que se utilizan para recolectar la información necesaria en la obtención de los requerimientos de una aplicación, permiten investigar aspectos generales para

posteriormente ser especificados con un mayor detalle, requieren ser adecuadamente orientadas para cubrir la información que se requiere capturar (Pressman, 2010).

Entrevista

La entrevista es una de las técnicas más utilizadas en la identificación de requisitos ya que responde a identificar los principales clientes y a mantener conversaciones técnicas para la identificación de los requisitos del sistema que va a ser desarrollado. En general, son escogidas para entrevistar a las principales personas y grupos de personas que usarán o recibirán algún impacto del sistema a desarrollar y su objetivo es determinar cuáles serán los usos que estos actores del negocio harán del sistema, el entorno que utilizarán, cuáles serán las limitaciones, cualidades y uso que harán de la herramienta para automatizar la instalación cifrada de Nova (Arias, 2014).

Observación

Por medio de esta técnica se obtiene información de primera mano sobre la forma en que se efectuarán las actividades. Este método permite observar la forma en que se llevan a cabo los procesos y, por otro, verificar que realmente se sigan los pasos especificados (Guerra, 2017).

2.2.4 Especificación de requisitos de software

La especificación de Requisitos de Software es una de las tareas más importantes en el ciclo de vida del desarrollo de software, porque en ella se determinan los planos de la nueva herramienta. Se realiza una descripción completa del comportamiento del sistema que se va a desarrollar. Y además de los requisitos funcionales y no funcionales (Vázquez, 2018).

Requisitos funcionales

Expresan las capacidades que debe poseer la solución, se identifican a partir de necesidades de negocio o necesidades de los clientes y usuarios. Estos requisitos describen beneficios que recibirán la organización y/o sus clientes con el desarrollo del sistema (Durango, 2015).

En la tabla 2 se realiza la especificación de los requisitos funcionales de la propuesta de solución, su descripción, complejidad y prioridad, determinada mediante el producto de trabajo Evaluación de Requisitos del expediente de proyecto 4.0 disponible para la actividad productiva de la universidad.

Tabla 2. Especificación de requisitos funcionales

(Fuente: elaboración propia)

Número	Requisito	Descripción	Complejidad	Prioridad

RF1	Especificar disco de instalación del sistema operativo	La herramienta debe permitir al usuario seleccionar el disco en el cual se realizará la instalación del sistema.	Baja	Media
RF2	Eliminar información existente	La herramienta debe eliminar toda la información contenida en el disco duro.	Alta	Baja
RF3	Crear nueva tabla de particiones	La herramienta debe especificar al sistema operativo, el número y el tipo de particiones que tiene nuestro sistema. Además de indicarle al sistema dónde se encuentra la partición de arranque.	Alta	Media
RF4	Cifrar disco duro	La herramienta debe cifrar el disco duro completamente exceptuando la partición de arranque.	Alta	Alta
RF5	Instalar sistema operativo	La herramienta debe instalar el sistema operativo GNU/Linux Nova	Media	Alta

Requisitos No Funcionales del sistema

Son restricciones de los servicios o funciones ofrecidas por la solución informática (tiempo, proceso o estándares del dominio de negocio), además de cualidades que se le imponen al sistema en cuanto al diseño y la implementación. Normalmente afectan al sistema en su totalidad más que una característica o servicio en particular (Durango, 2015).

Usabilidad

RNF 1: El sistema no debe presentar una interfaz gráfica, sino que utiliza una interfaz en línea de comandos.

Software

RNF 2: El sistema debe ser ejecutado en el sistema operativo GNU/Linux Nova.

RNF 3: El sistema debe ser programado en lenguaje de programación Bash como restricción de diseño.

Seguridad

RNF 4: El sistema para poder ser utilizado debe ser ejecutado como root.

2.3 Historias de usuario

Las historias de usuario (HU) constituyen una forma de administración de requisitos sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos. Las historias de usuario son cortas descripciones de una funcionalidad desde la perspectiva de la persona que la

desea, usualmente un usuario o cliente. Las mismas son escritas utilizando el lenguaje común. Son empleadas en las metodologías de desarrollo ágiles para la especificación de requisitos (COHN 2018).

En correspondencia con la selección del escenario número cuatro de la metodología empleada se procede a modelar el sistema con historias de usuario, donde se define una por cada requisito funcional, lográndose un total de 5 HU. Se muestran a continuación las HU “Especificar disco de instalación del sistema operativo”, “Cifrar disco duro” e “Instalar sistema operativo” (restantes en los anexos).

Tabla 3.RF1. Especificar disco de instalación del sistema operativo

(Fuente: elaboración propia)

Número: HU – 1	**Nombre del requisito: Especificar disco de instalación del sistema operativo.**
Programador: Alejandro Guilarte Larín	**Iteración asignada:** 1
Prioridad: media	**Tiempo estimado:** 3
Riesgo en desarrollo: medio	**Tiempo real:** 1
Descripción: luego de ejecutar la herramienta, esta permite al usuario seleccionar el disco en donde se realizará la instalación del sistema operativo. Se listan a continuación las funcionalidades requeridas: • Enumerar todos los discos disponibles. • Mostrar capacidad de todos los discos disponibles. • Mostrar marca del fabricante de todos los discos disponibles.	
Prototipo de interfaz:	

```
# determine which disk we're installing to
disks=$(lsblk | grep -P "disk *$" | awk '{print "/dev/"$1}')
while :
do
    [ $(wc -l <<< "$disks") -eq 1 ] && opt=1 && break
    echo "The following disks have been detected. To which disk would you like to install?"
    i=1
    for opt in $disks
    do
        grep -q '/dev/[sh]da' <<< "$opt" && default=$i
        printf "   [%$((1+$(wc -l <<< "$disks")/10))d] %s\n" $[i++] $opt
    done
    default=${default:-1}
    read -p "Enter the number of your selection [$default]: " opt
    opt=${opt:-$default}
    clear
    [ $opt -gt 0 ] && [ $opt -lt $i ] && break
done
disk=$(sed -n "${opt}p" <<< "$disks")

```

Tabla 4.RF4. Cifrar disco duro

(Fuente: elaboración propia)

Número: HU – 4	**Nombre del requisito: Cifrar disco duro**
Programador: Alejandro Guilarte Larín	**Iteración asignada:** 1
Prioridad: alta	**Tiempo estimado:** 3
Riesgo en desarrollo: medio	**Tiempo real:** 3
Descripción: luego de ejecutar la herramienta, esta realiza un cifrado del disco duro. Se listan a continuación las funcionalidades requeridas: • Instalar la herramienta de cifrado Dm-crypt / Luks. • Pedir al usuario clave o contraseña de cifrado. • Cifrar todas las particiones del disco duro a excepción de la partición destinada al arranque.	
Prototipo de interfaz:	

```
# setup LUKS encryption
echo "Setting up encryption:"
isEFI && luksPart=$(getDiskPartitionByNumber 3) || luksPart=$(getDiskPartitionByNumber 2)
cryptMapper="${luksPart/\/dev\/}_crypt"
echo -en "  Encrypting ${luksPart} with your passphrase ... "
echo -n "${luksPass}" | cryptsetup luksFormat -c aes-xts-plain64 -h sha512 -s 512 --iter-time 5000 --use-random -S 1 -d - ${luksPart}
echo -e "${green}done${normalText}"
if hasKeyfile; then
    echo -e "  ${boldText}We're going to need some random data for this next step. If it takes long, try  moving the mouse around or typing on the keyboar
    echo -n "  Adding key file as a decryption option for ${luksPart} ... "
    cryptsetup luksAddKey ${luksPart} "${keyfile}" <<< "${luksPass}"
    echo -e "${green}done${normalText}"
fi

# unlock LUKS partition
echo -n "  Decrypting newly created LUKS partition ... "
echo -n "$luksPass" | cryptsetup luksOpen ${luksPart} ${cryptMapper} && echo -e "${green}done${normalText}" || echo -e "${red}failed${normalText}"

```

Tabla 4.RF5. Instalar sistema operativo

(Fuente: elaboración propia)

Número: HU – 5	**Nombre del requisito: Instalar sistema operativo**
Programador: Alejandro Guilarte Larín	**Iteración asignada:** 1
Prioridad: alta	**Tiempo estimado:** 3
Riesgo en desarrollo: medio	**Tiempo real:** 2
Descripción: luego de ejecutar la herramienta esta realiza la instalación del sistema operativo GNU/Linux Nova. Se listan a continuación las funcionalidades requeridas: • Realizar la instalación del sistema operativo y todos sus componentes en la partición destinada para ello.	
Prototipo de interfaz:	

```
# mount stuff for chroot
echo -n "Mounting the installed system ... "
mount /dev/vg0/root /mnt
mount /dev/vg0/home /mnt/home
mount ${bootPart} /mnt/boot
isEFI && mount ${efiPart} /mnt/boot/efi
mount --bind /dev /mnt/dev
mount --bind /run/lvm /mnt/run/lvm
echo -e "${green}done${normalText}"
```

2.4 Análisis y Diseño

En esta disciplina, si se considera necesario, los requisitos pueden ser refinados y estructurados para conseguir una comprensión más precisa de estos y una descripción que sea fácil de mantener y ayude a la estructuración del sistema (incluyendo su arquitectura). Además en esta disciplina se modela el sistema y su forma para que soporte todos los requisitos, incluyendo los requisitos no funcionales. Los modelos desarrollados son más formales y específicos (Arias, 2014).

2.4.1 Diseño arquitectónico

En el diseño de la propuesta de solución es una arquitectura basada en flujo de datos y se utiliza como patrón arquitectónico Tuberías y Filtros donde los componentes o procesos se denominan filtros, ya que actúan como traductores que toman la entrada, la transforman según el algoritmo o funcionalidad que se desarrolle en el proceso, y generan la salida hacia un conducto de comunicaciones. Los conductos de entrada y salida se denominan tubos (Molinero Parra, 2018).

```
totalRAM=$(cat /proc/meminfo | head -n1 | grep -oP "\d+.*" | tr -d ' B' | tr 'a-z' 'A-Z' | numfmt --from iec --to iec --format "%.f")
read -p "Size for /boot [2G]: " boot
isEFI && read -p "Size for /boot/efi [100M]: " efi
read -p "Size for LVM [remaining disk space]: " lvm
read -p "Size for swap in LVM [$totalRAM]: " swap
read -p "Size for / (root) in LVM [32G]: " root
read -p "Percent of remaining LVM space to use for /home [100%]: " home
```

Figura 1: Tuberías y Filtros

(Fuente: Elaboración propia)

Estilo Arquitectónico

Para la implementación de la herramienta se selecciona el tipo Tuberías y filtros ya que este enfoque pone énfasis en la transformación gradual de los datos por componentes sucesivos. El flujo de datos es conducido por datos y todo el sistema se descompone en componentes de origen de datos, filtros, tuberías, y los sumideros de datos. Las conexiones entre los módulos son de flujo de que puede ser flujo de bytes, caracteres o cualquier otro tipo de este tipo. La característica principal de esta arquitectura es su ejecución concurrente (Ramos, 2017).

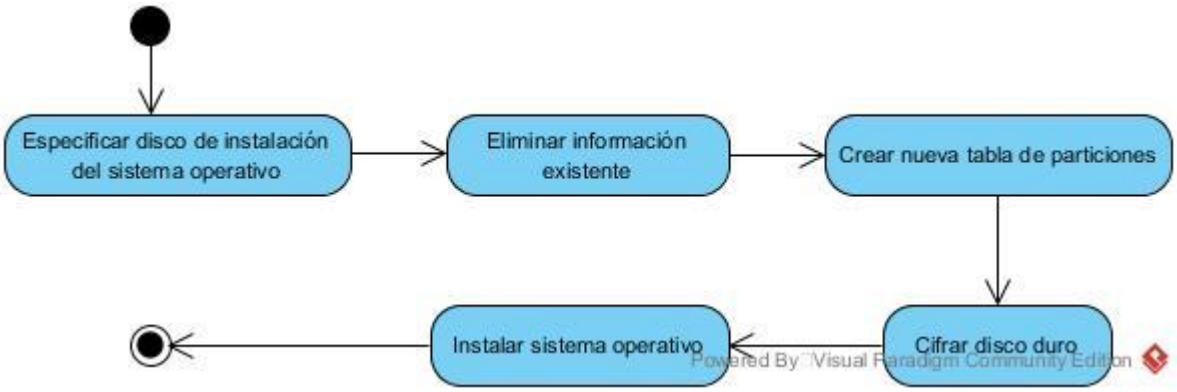

Figura 2: Tuberías y Filtros

(Fuente: Elaboración propia)

2.5 Conclusiones del Capítulo

El análisis y diseño de la propuesta de solución permitió realizar una descripción detallada de las características de la Herramienta Informática para automatizar la instalación cifrada de la distribución cubana GNU/Linux Nova, lo que permitió un mejor entendimiento para la fase de implementación al tener los principales artefactos para su desarrollo. Se definieron 5 requisitos funcionales y 4 no funcionales los cuales proporcionan una guía de desarrollo de las funcionalidades de la herramienta.

Se seleccionó como arquitectura Flujo de Datos ya que esta responde a las necesidades de la herramienta Informática.

La definición del estilo arquitectónico permitió un mejor entendimiento del funcionamiento del sistema.

CAPÍTULO 3: IMPLEMENTACIÓN, PRUEBAS Y EVALUACIÓN DE LA HERRAMIENTA INFORMÁTICA PARA AUTOMATIZAR LA INSTALACIÓN CIFRADA DE LA DISTRIBUCIÓN CUBANA GNU/LINUX NOVA

3.1 Introducción

El presente capítulo se enfoca en la construcción del sistema a partir de los resultados de la disciplina de Análisis y Diseño. Se elaboran los diagramas de componentes y de despliegue y se definen los estándares de codificación a utilizar en la implementación de la solución. Además, se realizan pruebas de software con el objetivo de descubrir y corregir errores y se evalúa la propuesta de solución.

3.1.1 Implementación

La etapa de implementación del software es el proceso de convertir una especificación del sistema en un sistema ejecutable (Sommerville, 2011). Esta fase comprende la materialización, en forma de código, de todos los artefactos, descripciones y arquitectura propuestos en la etapa de análisis y diseño; con el objetivo de conformar el producto final requerido por el cliente (Larman, 2004).

Una vez desarrollado el software, el mismo debe ser sometido a una serie de pruebas que muestren que el sistema se ajusta a su especificación y que cumple con las expectativas del cliente. Según (Sommerville, 2011), esta etapa es conocida como la validación e implica una serie de procesos de comprobación, como las inspecciones y revisiones.

3.1.2 Estándares de codificación

Los estilos de programación, también llamado estándares de código, es un término que describe convenciones para escribir código fuente en un cierto lenguaje de programación. El estilo de programación es dependiente del lenguaje de programación que se haya elegido. Ellos garantizan el mejoramiento de la comunicación en los equipos de desarrollo de software, reducen los errores de programación y mejoran la calidad del software. Todo esto repercute en la competitividad de las empresas de software y en la productividad de sus trabajadores porque se mejora su facilidad de mantenimiento teniendo un gran impacto en la reducción de los costos de mantenimiento (PEP8, 2015).

Para la codificación de la propuesta de solución se utiliza el estándar de codificación para Bash Scripting utilizado en el Universidad de las Ciencias Informáticas. En este documento se listan distintas convenciones utilizadas en el código Bash Scripting comprendido en la librería estándar de la distribución principal de Bash Scripting (PEP8, 2015).

Indentación (Sangría):

Se utilizaron 4 espacios por cada nivel de indentación.

Tabuladores o espacios:

No se mezclaron tabuladores y espacios en la codificación. Las formas más populares de indentar en Bash es utilizando sólo espacios o sólo tabuladores, el código indentado con una mezcla de tabuladores y espacios se reformateó y se usaron espacios exclusivamente.

Tamaño máximo de línea:

Todas las líneas están limitadas a un máximo de 82 caracteres.

Estructura

- Los nombres de las variables responderán al estilo de capitalización lowerCamelCase. Ver figura 3.
- Se inicializarán las variables locales donde se declaran. La única razón para no hacerlo será si su valor inicial depende de cálculos posteriores.
- No se emplearán caracteres especiales (@, #, $, %, ^, &, * u otros) para la nomenclatura.
- El idioma inglés se empleará para la codificación de la solución.

```
67  # function to convert things like 2G or 1M into bytes
68  bytes() {
69      num=${1:-0}
70      numfmt --from=iec $num 2> /dev/null || return 1
71  }
72
73  # get upper and lower bounds given the start and size
74  bounds() {
75      start=$(bytes $1)
76      size=$2
77      stop=$(($start + $(bytes $size) - 1))
78      echo $start $stop
79  }
80
```

Figura 3: Aplicación de los Estándares de Codificación

(Fuente: elaboración propia)

3.1.4 Diagrama de Despliegue

El diagrama de despliegue consiste en una representación estructural de la arquitectura del sistema desde el punto de vista de la distribución de los artefactos del software en los destinos de despliegue; definiendo a los artefactos como representaciones de elementos concretos en el mundo físico que son el resultado de un proceso de desarrollo (UCC, 2017).

A continuación, se muestra en la Figura 4 el Diagrama de Despliegue de la propuesta de solución.

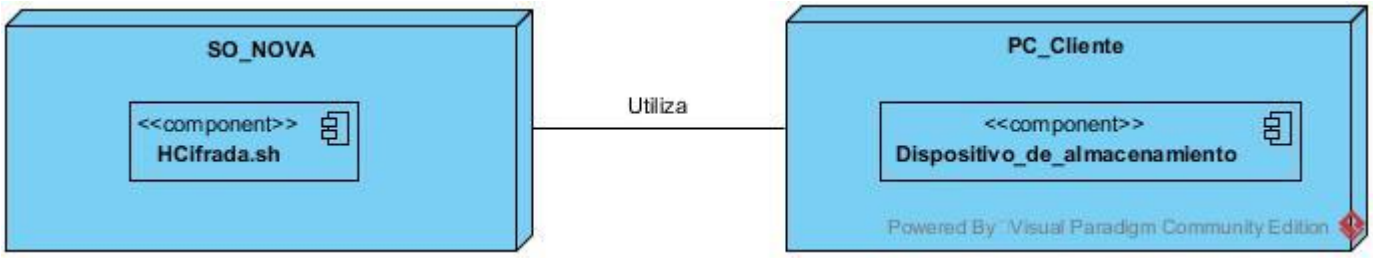

Figura 4: Diagrama de Despliegue

(Fuente: elaboración propia)

Descripción de los nodos:

- **SO_NOVA:** es el sistema operativo a instalar, que contiene la herramienta para automatizar la instalación cifrada (HCifrada.sh), dicha herramienta se puede ejecutar una vez se inicie el sistema operativo mediante su versión live (*en vivo*).
- **PC Cliente:** contiene los dispositivos o volúmenes de almacenamiento, sobre los cuales se realizara el proceso de instalación cifrada del sistema operativo Nova.

3.2 Pruebas de software

Luego del desarrollo de la propuesta de solución, se hace necesario verificar y validar el sistema implementado a través de una estrategia de pruebas que permitan comprobar el cumplimiento de las especificaciones del diseño y de la codificación e identificar los posibles errores cometidos.

Las pruebas de software son un conjunto de herramientas, técnicas y métodos que evalúan el desempeño y la calidad de un software, así como evaluar los resultados. Las técnicas son variadas y se realizan desde el personal de prueba hasta herramientas automatizadas que facilitan el trabajo y el rango de tiempo en que se realizan estas (Hidalgo, 2017).

3.2.1 Pruebas a realizar

Pruebas Funcionales

Las pruebas funcionales se concentran en las acciones visibles para el usuario y en la salida del sistema que este puede reconocer. La validación se alcanza cuando el software funciona de tal manera que satisface las experiencias razonables del cliente (Pressman, 2010).

Pruebas unitarias

Las pruebas unitarias se concentran en el esfuerzo de verificación de la unidad más pequeña del software, el componente o módulo de software. Tomando como guía de partida la descripción del diseño a nivel de componentes, se prueban importantes caminos de control para descubrir errores dentro de los límites del código. Centran su actividad en verificar la funcionalidad y la estructura (lógica interna) de cada elemento individualmente, una vez que ha sido codificado (Pressman, 2010).

3.2.2 Métodos de prueba

Caja blanca permite realizar verificaciones y validaciones directamente con el código fuente, para ello las **pruebas unitarias** son realizadas por el programador cada vez agrega una funcionalidad o método para asegurar la calidad del código. Esta técnica se basa en el diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivarlos.

Permite al ingeniero de software obtener casos de prueba que:

1. Garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método.
2. Ejerciten todas las decisiones lógicas en las vertientes verdaderas y falsas.
3. Ejecuten todos los bucles en sus límites operacionales.
4. Ejerciten las estructuras internas de datos para asegurar su validez.

Se considera a la prueba de **caja blanca** como uno de los tipos de pruebas más importantes que se le aplican al software, logrando como resultado que disminuya en un gran porciento el número de errores existentes en los sistemas y por ende una mayor calidad y confiabilidad (Pressman, 2010) .

3.2.3 Técnicas de prueba

Partición de equivalencia (posibles valores divididos en clases, valores de entrada y valores de salida). Se agrupan todos los valores para los cuales se espera que el programa tenga un comportamiento común (rango de valores) y esa es una clase de equivalencia, existen clases de equivalencia válidas y clases de equivalencia inválidas (FING, 2018)

Camino Básico permite obtener una medida de la complejidad lógica de un diseño procedimental y que use esta medida como guía para definir un conjunto básico de rutas de ejecución. Los casos de pruebas derivados para ejercitar el conjunto básico deben garantizar que se ejecuta cada instrucción por lo menos una vez durante la prueba (Pressman, 2010).

3.3 Aplicación de las pruebas de software

3.3.1 Pruebas internas
Pruebas Unitarias

Las pruebas unitarias se desarrollaron utilizando la técnica del camino básico del método de prueba caja blanca. En esta técnica se utilizó la métrica del software complejidad ciclomática que proporciona una medida cuantitativa de la complejidad lógica de un programa o procedimiento. Cuando se utiliza en el contexto de prueba el valor calculado mediante la complejidad ciclomática define el número de caminos independientes en el conjunto básico de un programa o procedimiento y proporciona un límite superior para el número de pruebas que se deben realizar para asegurar que se ejecuta cada sentencia al menos una vez (Pressman, 2010).En la figura 5 se presenta el procedimiento “encrypt” de la funcionalidad dentro del RF4. Cifrar disco duro.

Figura 5: Procedimiento encrypt del RF Cifrar disco duro

(Fuente: elaboración propia)

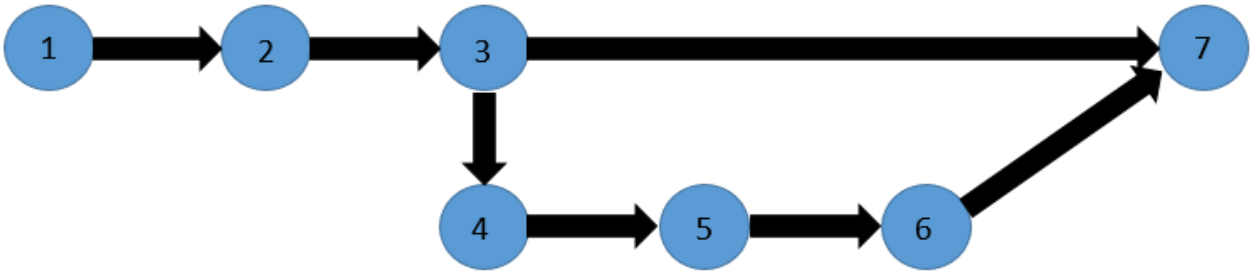

Figura 6: Grafo de flujo

(Fuente: elaboración propia)

La complejidad ciclomática es una medición de software que proporciona una evaluación cuantitativa de la complejidad lógica de un programa. Cuando se usa en el contexto del método de prueba de la ruta básica o camino básico, el valor calculado por la complejidad ciclomática define el número de rutas independientes del conjunto básico de un programa. Brinda una cota superior para el número de pruebas que debe realizar, a fin de asegurar que todos los enunciados se ejecutaron al menos una vez. La complejidad ciclomática tiene fundamentos en la teoría de grafos y proporciona una medición de software extremadamente útil. La complejidad se calcula de tres formas:

1. El número de regiones del grafo de flujo corresponde a la complejidad ciclomática.
2. La complejidad ciclomática **V(G)** para un grafo de flujo **G** se define como:

   V (G) = E - N + 2 donde **E** es el número de aristas del grafo de flujo y **N** el número de nodos del gráfico de flujo.

   V(G) = P + 1 donde **P** es el número de nodos predicado[1] contenidos en el gráfico de flujo **G** (Pressman, 2010).

En el grafo de flujo anterior, la complejidad ciclomática es calculada usando cada uno de los algoritmos indicados:

1. El grafo de flujo tiene dos regiones.
2. V (G) = 7 aristas - 7 nodos + 2 = 2.

1 *Un nodo predicado es el que representa una condicional if o case, es decir, que de él salen varios caminos.*

V (G) = 1 nodos predicado + 1 = 2.

La complejidad ciclomática del grafo de flujo asociado al método encrypt () tiene valor 2. V (G) proporciona la cota superior sobre el número de rutas linealmente independientes a través de la estructura de control del programa. Por tanto, las rutas o caminos posibles son las siguientes:

Tabla 5: Rutas básicas del grafo de flujo

(Fuente: elaboración propia)

Número de ruta	Rutas Básicas
1	1-2-3-7
2	1-2-3-4-5-6-7

Una vez determinadas las rutas o caminos básicos de flujo, se pasa a ejecutar los casos de pruebas para cada camino resultante. Los datos deben elegirse de modo que las condiciones en los nodos predicados se establezcan de manera adecuada conforme se prueba cada ruta. Cada caso de prueba se ejecuta y compara con los resultados esperados. Una vez completados todos los casos de prueba, el examinador puede estar seguro de que todos los enunciados del programa se ejecutaron al menos una vez.

Tabla 6: Caso de prueba Ruta Básica 1

(Fuente: elaboración propia)

Ruta Básica 1: 1-2-3-4-5-6-7	
Descripción	Proceso para correr el método encrypt.
Condición de ejecución	La herramienta haya particionado exitosamente el disco duro previamente.
Resultado esperado	Todas las particiones del disco exceptuando la de arranque, quedan cifradas.
Resultado	Satisfactorio.

Después de aplicado el método del camino básico, en el cual se realizaron 2 iteraciones de acuerdo a la cantidad de rutas básicas obtenidas, se concluye que, de los 5 casos de pruebas realizados (los restantes se encuentran en los Anexos), todos tuvieron resultados satisfactorios. Las no conformidades detectadas fueron resueltas, por tanto, la aplicación de la prueba fue satisfactoria en su totalidad.

Pruebas Funcionales

Para validar que la herramienta informática cumple con el funcionamiento esperado, se realizaron pruebas funcionales. El diseño de casos de prueba para la partición equivalente se basa en la evaluación de las

clases de equivalencia para una condición de equivalencia para una condición de entrada. A continuación se muestra el Caso de Prueba: Especificar disco de instalación del sistema operativo que contienen clases de equivalencia válida y no válida, para cada campo de entrada del sistema.

Tabla 7 Diseño de Caso de Prueba: Especificar disco de instalación del sistema operativo

(*Fuente: elaboración propia*)

Escenario	Descripción	Respuesta de la aplicación	Flujo central
EC1 Especificar disco de instalación del sistema operativo.	La aplicación al ser ejecutada debe permitir seleccionar en disco en que se desea instalar el sistema operativo.	Marca el disco seleccionado y automáticamente la herramienta lo toma como disco principal.	El usuario debe confirmar que entiende que puede perder información contenida en el disco a seleccionar y presionar enter El sistema muestra un listado con todos los discos y las propiedades de estos que posee el ordenador. El usuario selecciona el disco deseado. El sistema le pide una confirmación al usuario. El usuario presiona enter para continuar El sistema el sistema marca el disco seleccionado como disco principal para trabajar.

3.4 Evaluación del objetivo de la investigación

Para la validación del resultado, se utilizó la Técnica de Iadov y el método criterio de expertos en su variante Delphi:

Técnica de Iadov: Aplicada a una muestra de especialistas del proyecto NOVA para medir mediante el Índice de Satisfacción Grupal (ISG) el nivel de aceptación de la herramienta informática desarrollada, por parte de los usuarios finales.

3.4.1 Resultados de la técnica de Iadov

Cuando se realiza una propuesta, es recomendable retroalimentarse con la opinión de los usuarios potenciales. Esta información es útil para conocer las debilidades de la propuesta y profundizar en sus fortalezas. En ese sentido, **la técnica de Iadov** es un instrumento que ayuda a conocer el grado de satisfacción de los potenciales usuarios (Silega, 2017).

Mediante esta técnica se puede determinar, de forma indirecta, Índice de Satisfacción Grupal (ISG) de los individuos involucrados en el proceso que está siendo objeto de análisis. La técnica de Iadov se basa en un cuestionario conformado por cinco preguntas, de las cuales tres son cerradas y dos abiertas. Las preguntas cerradas guardan una relación entre sí, que previamente no es de conocimiento por parte del sujeto al que se le aplica la técnica (Ramírez, 2018).

Estas tres preguntas se relacionan a través del "Cuadro Lógico de Iadov" el cual permite ubicar a cada encuestado, según el cuadro lógico en una escala de satisfacción, para luego calcular el ISG. Las respuestas a cada una de estas preguntas permiten determinar la posición de cada sujeto en la escala de satisfacción que toma valores desde 1 hasta 6, distribuido de la siguiente manera: 1-Clara satisfacción, 2-Más satisfecho que insatisfecho, 3-No definida, 4-Más insatisfecho que satisfecho, 5-Clara insatisfacción y 6-Contradictoria (Ramírez, 2018). El cuadro lógico utilizado en la investigación se muestra en la siguiente tabla.

¿Qué tan satisfecho se siente con los resultados de la herramienta informática para automatizar la instalación cifrada de Nova?	¿Estima provechosa la herramienta informática desarrollada para el trabajo en el proyecto Nova?								
	Sí			No sé			No		
	¿Utilizaría la herramienta informática para la instalación cifrada de Nova?								
	Sí	No sé	No	Sí	No sé	No	Sí	No sé	No
Me satisface	1	2	6	2	2	6	6	6	6
Me resulta más satisfactorio que insatisfactorio	2	2	3	2	3	3	6	3	6
Me son indiferentes	3	3	3	3	3	3	3	3	3
Me resulta más insatisfactorio que satisfactorio	6	3	6	3	4	4	3	3	4
No me satisfacen en lo absoluto	6	6	6	6	4	4	6	6	5
No sé decir	2	3	6	3	3	3	6	6	4

Tabla 8 Cuadro lógico de Iadov utilizado

(Fuente: Elaboración propia)

Para medir el grado de satisfacción de los usuarios respecto al sistema desarrollado, se tomó como muestra a siete especialistas vinculados del proyecto Nova. La selección se realizó teniendo en cuenta la experiencia como analista y desarrolladores en el proceso de creación de la distribución cubana GNU/Linux Nova. Los resultados obtenidos para la satisfacción de forma individual se exponen a continuación.

A partir de la cantidad de respuestas por categoría es posible calcular el Índice de Satisfacción Grupal (ISG) siguiendo la siguiente fórmula:

$$ISG = \frac{\mathrm{A}(+1) + \mathrm{B}(+0.5) + \mathrm{C}(0) + \mathrm{D}(-0.5) + \mathrm{E}(-1)}{N}$$

Las variables representan las cantidades de participantes agrupados por las escalas del índice de satisfacción individual. La cantidad de participantes que expresaron tener una clara satisfacción son representados por A, la cantidad que se sienten más satisfechos que insatisfechos se expresan mediante B, no definido y contradicción se evidencia mediante C, los que se sienten más insatisfechos que satisfechos mediante D, E es la cantidad de participantes que expresan una clara insatisfacción. El valor de N representa el total de participantes.

El valor del ISG permite identificar las siguientes categorías grupales:

- Máxima insatisfacción: -1
- Más insatisfecho que satisfecho: -0.5
- No definido y contradictorio: 0
- Más satisfecho que insatisfecho: 0.5
- Máximo de satisfacción: +1

Esta técnica permite determinar el índice de satisfacción grupal (ISG), que representa los niveles de satisfacción en una escala numérica que abarca el intervalo desde -1 hasta 1. Los valores que se encuentran comprendidos entre -1 y - 0,5 indican insatisfacción; los comprendidos entre - 0,49 y + 0,49 evidencian contradicción y los que están entre 0,5 y 1 indican que existe satisfacción.

Resultados obtenidos

1. Los resultados obtenidos de la aplicación de la encuesta se presentan a continuación:

Categorías grupales de satisfacción	Escala	Participantes en la escala
Clara satisfacción	A	4
Más satisfecho que insatisfecho	B	3
No definido	C	0
Más insatisfecho que satisfecho	D	0
Clara insatisfacción	E	0
Contradictorio	C	0

Tabla 9 Escala del índice de satisfacción individual

Fuente: Elaboración propia

2. Cálculo del Índice de Satisfacción Grupal

ISG= A (+1)+B (+0.5)/N

ISG= (4(+1)+3(+0.5))/7 = 0.79

3. Interpretación del resultado del ISG

El proceso de validación mediante la técnica de Iadov de la consulta a los usuarios donde se ha implementado la herramienta informática para la herramienta informática para automatizar la instalación cifrada de GNU/Linux Nova propuesta, confirmó su factibilidad de uso, expresado cuantitativamente en el alto Índice de Satisfacción Grupal (ISG= 0.79) y cualitativamente en los criterios emitidos donde evidencian su satisfacción por la contribución del sistema, lo que refleja aceptación de la propuesta y un reconocimiento a su utilidad. Las respuestas a las preguntas abiertas brindadas por los encuestados reafirman los beneficios que traerá la utilización del sistema propuesto. Por lo anteriormente planteado se puede afirmar que se cumplió el objetivo de la investigación.

El formulario presentado a los participantes incluye dos preguntas abiertas, mostradas a continuación:

1. ¿Qué importancia le concede a la herramienta informática para la automatizar la instalación cifrada de GNU/Linux Nova?

2. ¿Qué aspectos a su juicio potencian o limitan el uso de la herramienta informática para automatizar la instalación cifrada de GNU/Linux Nova?

Sobre la primera pregunta, los participantes manifestaron que resulta de gran importancia porque permite de manera automatizada la instalación cifrada de la distribución cubana GNU/Linux Nova, ahorrando tiempo y evitando la necesidad de ser experto en la temática para lograrlo. Permitiendo de esta manera un alto nivel de seguridad, especialmente en entidades o individuos que manejan información sensible.

Sobre la segunda pregunta los participantes expresaron que la creciente necesidad de mejorar el proceso de automatización para la instalación cifrada la distribución cubana GNU/Linux Nova y como podría esta tener una interfaz gráfica que interactúe con el usuario.

3.5 Conclusiones del Capítulo

En el capítulo concluido se realizó la elaboración del diagrama de componentes, facilitó la comprensión de la estructura general del sistema a través de sus componentes.

Se estableció el estándar de codificación a tener en cuenta para la implementación de la herramienta informática para la automatización para la instalación cifrada de la distribución cubana GNU/Linux Nova, permitiendo una mejor legibilidad y comprensión del código, facilitando su mantenimiento.

La implementación del sistema permitió la obtención de una aplicación funcional y completamente operativa.

La ejecución de la estrategia de pruebas especificada, permitió detectar y corregir deficiencias presentes en la solución y ofrecer una aplicación con mayor calidad, seguridad y usabilidad.

Se evidenció la satisfacción de la propuesta de solución a partir de la técnica de Iadov, la cual propició la evaluación satisfactoria de la herramienta para la automatización instalación cifrada la distribución cubana GNU/Linux Nova.

CONCLUSIONES GENERALES

La investigación realizada cumple con los objetivos planteados mediante el desarrollo de la Herramienta Informática para automatizar la instalación cifrada de la Distribución Cubana GNU/Linux Nova y se arriba a las siguientes conclusiones:

- El análisis de los referentes teóricos y de los sistemas informáticos estudiados evidenció la necesidad de desarrollar una herramienta informática para automatizar el proceso de instalación cifrada de la Distribución Cubana GNU/Linux Nova.
- Se obtuvo una herramienta informática que permite automatizar el proceso de instalación cifrada de la Distribución Cubana GNU/Linux Nova.
- La evaluación de la investigación mediante las pruebas realizadas garantizan el correcto funcionamiento de la aplicación y demostraron la satisfacción del cliente hacia la herramienta desarrollada.

REFERENCIAS

AROCHE, EUGENIO DURÁN., 2014. *Trabajo de diploma para optar por ingeniero en Ciencias Informáticas. Sistema de Construcción de Personalizaciones de Nova*. La Habana, Cuba: Universidad de las Ciencias Informáticas.

AVIZTAR, 2017. Curso práctico de Modelado de Negocios con BPMN y UML. [en línea]. [Consulta: 3 diciembre 2019]. Disponible en: http://www.milestone.com.mx/CursoModeladoNegociosBPMN.htm.

CANEPA, G., 2018. Qué es Bash: shell, intérprete, y más. *Blog Carrera Linux* [en línea]. [Consulta: 4 diciembre 2019]. Disponible en: https://blog.carreralinux.com.ar/2018/04/que-es-bash-shell-interprete-y-mas/.

CHIROLDES, A.T., 2014. *Herramienta para gestionar los repositorios locales de software en Nova.* La Habana, Cuba: Universidad de las Ciencias Informáticas.

Cifrado de datos. Qué es y cómo hacerlo. *Grupo Ático34* [en línea], 2018. [Consulta: 22 febrero 2020]. Disponible en: https://protecciondatos-lopd.com/empresas/cifrado-datos/.

DEBIAN, 2016. ¿Qué es GNU/Linux? [en línea]. [Consulta: 3 diciembre 2019]. Disponible en: https://www.debian.org/releases/stable/mips/ch01s02.html.es.

CAPRIL, 2017. Definición de instalación — Definicion.de. *Definición.de* [en línea]. [Consulta: 22 febrero 2020]. Disponible en: https://definicion.de/instalacion/.

Disk encryption (Español) - ArchWiki. [en línea], 2019. [Consulta: 3 diciembre 2019]. Disponible en: https://wiki.archlinux.org/index.php/Disk_encryption_(Espa%C3%B1ol).

eCryptfs in Launchpad. [en línea], 2016. [Consulta: 3 diciembre 2019]. Disponible en: https://launchpad.net/ecryptfs.

Encrypting an entire system. [en línea], 2015. [Consulta: 3 diciembre 2019]. Disponible en: https://wiki.archlinux.org/index.php/Dm-crypt_(Espa%C3%B1ol)/Encrypting_an_entire_system_(Espa%C3%B1ol).

ERNESTO RODRIGUEZ ORTIZ & GEORGE SÁNCHEZ AMAYA, 2009. *Rubik, contenedor cifrado de datos personales*. La Habana, Cuba: Universidad de las Ciencias Informáticas.

How to install Arch Linux with Full Disk Encryption. *HowtoForge* [en línea], 2015. [Consulta: 3 diciembre 2019]. Disponible en: https://www.howtoforge.com/tutorial/how-to-install-arch-linux-with-full-disk-encryption/.

Instalinux. *LinuxCOE* [en línea], 2014. [Consulta: 3 diciembre 2019]. Disponible en: http://www.instalinux.com/howto.php.

JAVIER PIÑEIRO CÁRDENAS, 2017. *Herramienta web para la creación de personalizaciones de Nova Servidores*. La Habana, Cuba: Universidad de las Ciencias Informáticas.

KARSPESKY, 2019. ¿Qué es la ciber seguridad en internet? | Kaspersky. [en línea]. [Consulta: 20 febrero 2020]. Disponible en: https://latam.kaspersky.com/resource-center/definitions/what-is-cyber-security.

LINUX-ES, 2015. El rincón de Linux. [en línea]. [Consulta: 3 diciembre 2019]. Disponible en: http://www.linux-es.org/sobre_linux.

Loop-aes vs DM-crypt. [en línea], 2017. [Consulta: 3 diciembre 2019]. Disponible en: https://www.linuxquestions.org/questions/linux-security-4/loop-aes-vs-dm-crypt-385324/.

PEÑALVER, GLADYS MARSI ROMERO, 2017. *Ingenierias Ágiles*. La Habana, Cuba: Universidad de las Ciencias Informáticas.

POUSA, A., 2011. *Algoritmo de cifrado simétrico AES* [en línea]. Tesis. S.l.: Universidad Nacional de La Plata. [Consulta: 3 diciembre 2019]. Disponible en: http://sedici.unlp.edu.ar/handle/10915/4210.

RODRÍGUEZ, TAMARA SÁNCHEZ., 2015. *Metodología de desarrollo para la actividad productiva de la UCI*. La Habana, Cuba: Universidad de las Ciencias Informáticas.

SD4L - ScramDisk for Linux. [en línea], 2013. [Consulta: 3 diciembre 2019]. Disponible en: http://sd4l.sourceforge.net/.

TARGETWARE, 2016. Software.com.ar. Visual Paradigm para UML. [en línea]. [Consulta: 3 diciembre 2019]. Disponible en: http://www.software.com.ar/p/visual-paradigm-para-uml.

UBUNTU-ES. 2016., 2016. Sobre Ubuntu. [en línea]. [Consulta: 3 diciembre 2019]. Disponible en: http://www.ubuntu-es.org/sobre_ubuntu.

Martinez. 2015. PostgreSQL . [En línea] 2 de octubre de 2015. [Citado el: 25 de octubre de 2018.] http://www.postgresql.org.

Abran, Alain and Moore, James W. 2004. *Swebok. Guide to the Software Engineering Body of Knowledge.* Estados Unidos de América : IEEE Computer Society Professional Practices Committee, 2004. 204.

Abreu, Ramon. 2016. 2016.

Acunetix Vulnerability Scanner. 2016. Audit Your Web Security with Acunetix Vulnerability Scanner . [En línea] 2016. [Citado el: 22 de 03 de 2019.] https://www.acunetix.com/vulnerability-scanner/.

AFT. 2012. Administracion. [En línea] 11 de junio de 2012. [Citado el: 2016 de 10 de 10.] http://www.administracionmoderna.com/2012/06/activo-fijo-tangible.html.

Alegre Ramos, María del Pilar. 2019. Sistemas Operativos Monopuesto. *[en linea].* [En línea] 11 de mayo de 2019. [Citado el: 2 de octubre de 2019.] https://books.google.com.cu/books?id=qt-ZDwAAQBAJ&printsec=frontcover&dq=que+es+un+sistema+operativo&hl=es&sa=X&ved=0ahUKEwjH35S_h-

rlAhXExFkKHYgdBEAQ6AEINTAC#v=onepage&q=que%20es%20un%20sistema%20operativo&f=false. ISBN 978-84-283-4139-4.

Almeira, Adriana. 2007. *Arquitectura de Software: Estilos y Patrones.* Argentina : Universidad Nacional De La Patagonia San Juan Bosco, 2007.

Álvarez, Miguel Angel. 2017. DesarrolloWeb. [En línea] 14 de septiembre de 2017. [Citado el: 07 de noviembre de 2018.] https://DesarrolloWeb.com.

Amparo López Gaona. 2012. El modelo Entidad-Relación. [En línea] Universidad Nacional Autónoma de México. Facultad de Ciencias, 2012. [Citado el: 14 de 04 de 2017.] http://hp.fciencias.unam.mx/~alg/bd/.

Arias, Angel. 2014. *Aprende sobre la Ingeniería del Software.* [https://books.google.com.cu/books?id=0E3mCgAAQBAJ&pg=PT85&lpg=PT85&dq=entrevista+como+t%C3%A9cnica+de+identificaci%C3%B3n+de+requisitos+del+software&source=bl&ots=BiZALVmCtd&sig=ACfU3U3PyqFqdI9UokZnVWtkf-GO6yrk2A&hl=es&sa=X&ved=2ahUKEwiJz922uqznAhVyuVkKHZ] s.l. : IT Campus Academy, 2014. 978-1497417656.

Aroche, Eugenio Durán. 2012. *Sistema de Construcción de Personalizaciones de Nova.* Universidad de las Ciencias Informáticas. La Habana : s.n., 2012. Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas.

—. 2014. Trabajo de diploma para optar por ingeniero en Cencias Informáticas. *Sistema de Construcción de Personalizaciones de Nova.* La Habana : Universidad de las Ciencias Informáticas, 2014.

Arrete, Juan Pablo. 2016. Una introducción a MongoDB. [En línea] 09 de marzo de 2016. [Citado el: 14 de enero de 2018.] https://www.blogs-informaticos.com/MongoDB.

Bartak, Pedro. 2013. [En línea] 9 de 12 de 2013. [Citado el: 5 de 4 de 2017.] http://www.grandespymes.com.ar/2013/12/29/definicion-y-componentes-del-sistema-de-control-de-gestion-en-las-empresas/.

Bejerano, Yunier Pñerez. 2019. *Sistema de Gestión de Licencias del Personal Aeronáutico del Instituto de la Aeronática Civil de Cuba versión 2.0.* Universidad de las Ciencias Informáticas. La Habana : s.n., 2019. Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas.

Berberat, Esteve. 2012. *Visual Paradigm.* Universidad de Madrid. Madrid : s.n., 2012. Tesis de Maestria.

Boltton, Vladimir. 2017. UML_Secuencia. [En línea] 15 de 02 de 2017. [Citado el: 13 de 02 de 2019.] https://uml.org/uml-secuencia.

Bresnahan, Christine y Blum, Richard. 2015. LPIC-1: Linux Professional Institute Certification Study Guide. [En línea] 2015. [Citado el: 22 de septiembre de 2019.] https://books.google.com.cu/books?id=jf3zBgAAQBAJ&pg=PA140&dq=fdisk&hl=es&sa=X&ved=0ahUKEwiS3cuM5JjmAhUSvVkKHU6qAHMQ6AEIUTAE#v=onepage&q=fdisk&f=false. ISBN 978-1-119-02118-6.

Brito, Eugenio González. 2015. Pruebas de rendimiento de software. [En línea] 18 de 07 de 2015. [Citado el: 21 de 03 de 2019.]

https://cgrw01.cgr.go.cr/rup/RUP.es/LargeProjects/core.base_rup/guidances/concepts/performance_testing_37A31809.html.

Cabriales, Andy Michel Moreno. 2013. *Manejador de Copias para NOVA 4.0.* Universidad de las Ciencias Informáticas. La Habana : s.n., 2013. Título para optar por el título como Ingeniero en Ciencias Informáticas.

Canales, Pedro Ramos. 2017. Visual Paradigm. *Cómo se realizan los diagramas en Visual Paradigm.* [En línea] 22 de octubre de 2017. [Citado el: 20 de enero de 2019.] https://www.adictosaltrabajo.com/tutoriales/vparadigm/.

CELERY. 2017. Celery: Distributed Task Queue. [En línea] 20 de octubre de 2017. [Citado el: 4 de mayo de 2019.] https://celery/distrbuted_task_queue. ISBN: 974-48-448-1.

Chakray. 2017. ¿Qué es el BPMN y para que sirve? [En línea] 20 de septiembre de 2017. [Citado el: 10 de noviembre de 2019.] https://www.chakray.com/es/que-es-el-bpmn-y-para-que-sirve/.

Condori, Jose Luis. 2016. *Phython - DjangoFramework de desarrollo web para perfeccionistasBasado en el Modelo MTV.* 2016.

Creed, Manuel. 2017. Marco de Desarrollo de la Junta de Andalucía. [En línea] 2017.

Cruz, Manuel Enrique Peiso. 2018. *Procedimiento de construcciónde imágenes de la personalizacion de GNU/Linux Nova Escritorio.* Universidad de las Ciencias Informáticas. La Habana : s.n., 2018. Trabajo presentado en opción al título de Máster een Informática Aplicada.

CSS, Learn to style HTML using. 2017. Mozilla Developer Network. [En línea] 23 de 11 de 2017. [Citado el: 25 de 05 de 2018.] https://developer.mozilla.org/en-US/docs/Learn/CSS..

Durango, Alicia. 2015. *Diseño de software: 2 Edición.* [https://books.google.com.cu/books?id=1ShpCwAAQBAJ&pg=PA15&dq=que+es+un+requisito+funcional&hl=es&sa=X&ved=0ahUKEwj958Grz6znAhXSt1kKHZXBDzEQ6AEIRDAE#v=onepage&q=que%20es%20un%20requisito%20funcional&f=false] s.l. : IT Campus Academy, 2015. ISBN:978-1519620736.

Echevarría, Dayma. 2010. *Proceso de perfeccionamiento empresarial.* La Habana : Universidad de la Habana, 2010.

EPSILONMAG. 2017. epsilon. *Por que linux es considerado el mejor sistema operativo.* [En línea] 2017. [Citado el: 24 de septiembre de 2019.] https://epsilonmag.com/pc/por-que-linux-es-considerado-el-mejor-sistema-operativo/.

Estela Raffino, María. 2018. Concepto.de. *Software Libre.* [En línea] 23 de noviembre de 2018. [Citado el: 14 de noviembre de 2019.] https://concepto.de/software-libre/.

FERGARCIA. 2013. Entorno de desarrollo integrado. [En línea] 25 de enero de 2013. [Citado el: 1 de noviembre de 2019.] https://fergarciac.wordpress.com/2013/01/25/entorno-de-desarrollo-integrado-ide/.

Fernández, Carlos Martínez. 2014. *Metodología de desarrollo para la actividad productiva de la UCI.* Universidad de las Ciencias Informáticas. La Habana : UCI, 2014. Tesis de maestria.

Ferrer, Laurato Cantínez. 2017. *Propuesta de conceptualización de requisitos para proyectos software basados en formalismo de ingeniería de conocimiento.* Madrid : Evens Books, 2017. ISBN: 978-48-751-2.

Ferrer, Vilma Rodriguez. 2016. SQlite, sistema gestor de base de datos. [En línea] 29 de julio de 2016. [Citado el: 01 de 11 de 2018.] https://blogs-enginering/Sytms-of-Date-Base.

FING. 2018. Find Edu. *Testing basado en el cubrimiento.* [En línea] 2018. [Citado el: 22 de marzo de 2019.] https://www.fing.edu.ut/inco/cursos/ingsoft/.

Fuente Rodríguez, Juan Manuel. 2018. Tesis de Maestría en Informática Avanzada. *Procedimiento de gestión de los Repositorios de la Distribución Cubana GNU/Linux Nova.* La Habana, Cuba : Universidad de las Ciencias Informática, 2018.

Fuentes, Juan Manuel Rodríguez. 2018. *Procedimiento de Gestión de los Repositorios de la Distribución Cubana de GNU/Linux Nova.* Universidad de Las Ciencias Informáticas. La Habana : s.n., 2018. Tesis de Maestría de Informática Avanzada.

Gasterol, Yoandri. 2015. [En línea] 17 de 09 de 2015.

Gauchat, Juan Diego. 2013. *Gran libro de html5, css3 y javascript.* España : CEDRO, 2013.

GIT. 2016. Git- Control de versiones. [En línea] 26 de marzo de 2016. [Citado el: 25 de marzo de 2019.] https://git-scm.com/book/es/v1/Empezando-Acerca-del-control-de-versiones..

Gómez, Maria del Carmen. 2015. *Notas del curso Base de Datos. MÉXICO, D.F : Casa abierta al tiempo.* 2015.

González, Dra. Anaisa Hernández. 2013. *Diagramas de Casos de Uso del Negocio y del Sistema.* Habana, Cuba : Instituto Superior Politécnico Jose Antonio Hechavarría (CUJAE), 2013.

González, Jesus Ponce, Dominguez, Fransisco Jose, Rodriguez, Javier Gutierrez and Escalona, Maria Jose. 2014. *Pruebas de aceptación orientadas al usuario: contexto ágil para un proyecto de gestión documental.* España : Universidad de Sevilla, Universidad de Sevilla, 2014. ISSN 1888-0967.

GPARED. 2017. gpared. *Manual de GParted.* [En línea] Septiembre de 2017. [Citado el: 1 de noviembre de 2019.] https://gparted.org/display-doc.php?name=help-manual&lang=es.

Granda, Ailec. 2016. Modelo didáctico para el uso de comunidades virtuales en el proceso de enseñanza aprendizaje de la Disciplina Ingeniería y Gestión de Software en la Universidad de las Ciencias Informáticas. Tesis para obtener el grado de doctora en tecnología educativa. 2016.

Guerra, Arturo César. 2017. Obtención de Requerimientos. Técnicas y Estrategia. [En línea] 2017. [Citado el: 12 de diciembre de 2019.] https://sg.com.mx/revista/17/obtencion-requerimientos-tecnicas-y-estrategia.

Gutiérrez, Javier J. 2015. framework Django. [En línea] 2015.

Helmke, Matthew, Hudson, Andrew y Hudson, Paul. 2015. *Ubuntu Unleashed 2015 Edition: Covering 14.10 and 15.04.* [https://books.google.com.cu/books?id=U1RwBQAAQBAJ&pg=PA161&dq=fdisk&hl=es&sa=X&ved=0ahUKEwjxreiz5pjmAhVkrlkKHTYCBBs4ChDoAQhMMAQ#v=onepage&q=fdisk&f=false] Indianapolis : Pearson Education. Inc, 2015. 978-0-672-33837-3.

Hertzog, Raphael. 2015. ¿Why a GNU/Linix Distribution? *The Debian Administrator's Handbook.* [En línea] 2015. [Citado el: 10 de octubre de 2019.] https://debian-handbook.info/browse/stable/sect.why-gnu-linux.html. ISBN: 979-10-91414-04-3.

Hertzog, Raphael y Mas, Roland. 2015. *The Debian Administrator's Handbook.* New York : Debian Free Software Guidelines, 2015. ISBN 979-10-91414-05-0.

Heurtel, Olivier. 2011. *Desarrollar un sitio Web dinámico e interactivo.* s.l. : ENI Ediciones, 2011. 497.

Hidalgo, Asney Palmero. 2017. *Herramienta para la gestión de perfiles de AppArmor para GNU/Linux Nova servidores.* Universidad de las Ciencias Informáticas. La Habana : s.n., 2017. Título para optar como ingeniero en Ciencias Informáticas.

Hipertextual. 2015. *Iniciativas a favor del software libre en América Latina.* [En línea] 2015. [Citado el: 2 de septiembre de 2019.] https://hipertextual.com/2015/04/politicas-de-software-libre-en-latinoamerica.

Hugo, Jorge. 2014. Analisis de perfiles de usuarios. [En línea] 28 de noviembre de 2014. [Citado el: 17 de octubre de 2018.] https://www.googlebook.com/Análisis-de-perfiles-de-usuarios.

IEEE. 2014. IEEE Recommended Practice for the Adoption of Computer-Aided Software Engineering (CASE) Tools. *Software Engineering.* 2da, 2014, Vol. 40, 24.

Jacobson, Ivar, Boocch, Grady and Rumbaugh, James. 2000. [aut. libro] Ivar Jacobson. *El proceso unificado de desarrollo de software.* Madrid, España : Pearsons Educacion S.A, 2000.

Jahongir Rahmonov. 2019. PyCharm for Productive Python Development (Guide). [En línea] 28 de agosto de 2019. [Citado el: 23 de diciembre de 2019.] https://realpython.com/pycharm-guide/.

Jiménez Toro, J.A. 2015. *UF1876- Atención a usuarios e instalación de aplicaciones cliente.* [https://books.google.com.cu/books?id=m7lWDwAAQBAJ&pg=PA72&dq=los+sistemas+operativos+gen%C3%A9ricos&hl=es&sa=X&ved=0ahUKEwjL77fsxuDnAhXSVt8KHZL0BAQQ6AEIJzAA#v=onepage&q=los%20sistemas%20operativos%20gen%C3%A9ricos&f=false] Madrid : ELEARNING, 2015. ISBN 978-84-15199-14-3.

Jiménez, J.A Toro. 2015. *Uf-1876-Atención a usuarios e instalación de aplicaciones cliente.* [En línea] 22 de octubre de 2015. [Citado el: 1 de septiembre de 2019.] https://books.google.com.cu/books?id=m7lWDwAAQBAJ&pg=PA76&dq=sistema+operativo+espec%C3%ADfico+jimenex+toro&hl=es&sa=X&ved=0ahUKEwjN8eeooZ_oAhWEmuAKHQWoD80Q6AEIKTAA#v=onepage&q=sistema%20operativo%20espec%C3%ADfico%20jimenex%20toro&f=false. ISBN 978-84-15199-14-3.

Jiménez, Jose. 2018. *Unidad 1. Principios del modelado del negocio.* Ciudad de México : Universidad Abierta y a Distancia de México, 2018.

Kaplan-Moss, Adrian Holovaty y Jacob. 2016. *The Definitive Guide to django.* 2016.

Kerlinger, Fred. 2016. perfiles de usuario en redes sociales. *perfiles de usuario en redes sociales.* México D.F : s.n., 2016.

Kili, Aaron. 2018. TecMint.com. *Top 6 Partition Managers for Linux.* [En línea] RHCSA and RHCE Certification Exam Study Book, 9 de febrero de 2018. [Citado el: 24 de octubre de 2019.] https://www.tecmint.com/linux-partition-managers/.

Larman, C. 2004. *UML y Patrones: Introducción al análisis y diseño orientado a objetos. La Habana: Félix Varela.* 2004.

—. 2004. *UML y Patrones: Introducción al análisis y diseño orientado a objetos. La Habana: Félix Varela.* 2004.

Larman, Craig. 2003. *UML y Patrones.* s.l. : Prentice Hall 2da Edición, 2003.

Lenguajes de programación. **Kuzko, Richard. 2016.** 2016.

León García, Aldy. 2017. Trabajo de diploma para optar por el título Ingeniero en Ciencias Informáticas. *Control de acceso a recursos compartidos en el Nautilus.* La Habana, Cuba : Universidad de las Ciencias Informáticas, 2017.

León, Aldy García. 2017. *Control de acceso a recursos compartidos en el Nautilus.* Universidad de las Ciencias Informáticas. La Habana : s.n., 2017. Trabajo de Diploma para optar por el título de ingeniero en ciencias informáticas.

Leontis, Neocles. 2012. RNA 3D Structure Analysis and Prediction. [En línea] 2012. [Citado el: 28 de abril de 2019.] https://books.google.com.cu/books?id=4nVaim7HXmEC&pg=PA113&dq=Novo+Builder&hl=es&sa=X&ved=0ahUKEwim1ZbKmpvmAhUBmlkKHcNkATEQ6AEIJzAA#v=onepage&q=Novo%20Builder&f=false. ISBN 978-3-642-25739-1.

Lewis, William E. 2005. *Software testing and continuous quality improvement.* Estados Unidos : CRC Press LLC, 2005. ISBN: 0-8493-2524-2.

LinuxAdictos. 2017. *Como se compone la estructura del sistema de archivos de linux.* [En línea] 5 de octubre de 2017. [Citado el: 2 de noviembre de 2018.] https://www.linuxadictos.com/como-se-compone-la-estructura-del-sistema-de-archivos-de-linux-parte-1.html.

LLonch. 2016. Symfony. [En línea] 2016. [Citado el: 2017 de febrero de 2017.] http://symfony.com.

Loisel, Jérôme. 2016. JMeter Tutorial for Beginners. [En línea] 2016. [Citado el: 18 de 03 de 2019.]

Lopes-Martinez, Igor. 2012. 2012.

López, Dailyn Sosa. 2016. Que es un software Libre. [En línea] 22 de octubre de 2016. [Citado el: 2 de noviembre de 2018.] https://google_book/software_libre/. ISBN: 975-84-58-71-1.

López, David. 2014. Genbeta. [En línea] 9 de enero de 2014. https://www.genbetadev.com/herramientas/netbeans-1.

López, MSc. Dailyn Sosa. 2016. [En línea] 2016.

MAKETECHEASIER. 2010. maketecheasier. *Build Your Own Ubuntu-based Distro With Novo Builder.* [En línea] 2 de Julio de 2010. [Citado el: 23 de octubre de 2019.] https://www.maketecheasier.com/build-your-own-ubuntu-based-distro-with-novo-builder/.

MarketShareReport. 2019. *Linux Market Share.* [En línea] 24 de enero de 2019. [Citado el: 19 de octubre de 2019.] https://netmarketshare.com/linux-market-share.

Martinez, Alberto. 2015. Las redes sociales en Internet. [En línea] 18 de 12 de 2015. [Citado el: 10 de 11 de 2018.]

Mateo, Sergi. 2015. Manual práctico de benchmarking. [En línea] 2015. [Citado el: 29 de 10 de 2018.]

Molinero Parra, Jose Manuel. 2018. *UF2218-Desarrollo de un CMS.* [https://books.google.com.cu/books?id=cF5WDwAAQBAJ&pg=PA60&dq=tuber%C3%ADas+y+filtros+es+un+patr%C3%B3n+arquitect%C3%B3nico&hl=es&sa=X&ved=0ahUKEwi31ZirwITOAhVDc98KHQjsDXMQ6AEIOjAC#v=onepage&q=tuber%C3%ADas%20y%20filtros%20es%20un%20patr%C3%B3n%20arquitect] 2018. 978-447-58-444-741-1.

Montero, Richard. 2014. Universidad de Antioquia. Escuela Interamericana de Bibliotecología. Seminario de Estudios de Usuarios. Unidad 2. [En línea] 25 de 06 de 2014. [Citado el: 08 de 12 de 2018.]

Naramore, Elizabeth. 2016. [En línea] 2016.

Negus, Christopher. 2014. *Live Linux CDs: Building and Customizing Bootables.* Boston : Safari Book online, 2014. ISBN 0-13-243274-9.

Nikkel, Bruce. 2016. *Practical Forensic Imaging: Securing Digital Evidence with Linux Tools.* [ed.] Alinson Law. San Francisco : Library of Congress Catalogy- in- Publication Data, 2016. ISBN 1-59327-793-8.

Noguera, Bulmaro. 2016. Culturación. Ventajas de utilizar HTML 5. [En línea] 16 de 08 de 2016. [Citado el: 2018 de 10 de 25.]

Nolasco, Jorge Santiago Valenzuela. 2018. *Pyton Aplicaciones prácticas.* Madrid : RA-MA Editorial, 2018. 978-849964-758-6.

NOVA. 2019. *Soluciones a la medida.* [En línea] Universidad de las Ciencias Informáticas, 2019. [Citado el: 1 de octubre de 2019.] https://www.nova.cu/.

Nuñez, Inter Hnos. 2014. importancia de las redes sociales en la sociedad actual. [En línea] 24 de junio de 2014. [Citado el: 05 de octubre de 2018.]

OBSBUSINESS. 2019. *¿Qué son las metodologías de desarrollo de software?* [En línea] Universidad de Barcelona, 26 de octubre de 2019. [Citado el: 22 de noviembre de 2019.] https://obsbusiness.school/int/blog-project-management/metodologia-agile/que-son-las-metodologias-de-desarrollo-de-software.

Pacheco, Juan Manuel Fernandez. 2017. Real Academia Española. [En línea] 2017. [Citado el: 07 de 10 de 2018.]

Palmero, Asney Hidalgo. 2017. Trabajo de diploma para optar por ingeniero en Ciencias Informáticas. *Herramienta para la gestión de perfiles de AppArmor para GNU/Linux Nova servidores.* La Habana : Universidad de las Ciencias Informáticas, 2017.

Pantaleo, Guillermo. 2016. *Ingeniería de Software.* [https://books.google.com.cu/books?id=a8j2DQAAQBAJ&pg=PT557&dq=modelo+conceptual+en+ingenieria+de+software&hl=es&sa=X&ved=0ahUKEwiv3sOioavnAhXmwVkKHVg6C9AQ6AEIJzAA#v=onepage&q=modelo%20conceptual%20en%20ingenieria%20de%20software&f=false] s.l. : Editorial- Ink, 2016. 978-987-1609-78-9.

Parrilla, Castillo y Antonio, José. 2018. *Bienes digitales. Una necesidad europea.* [https://books.google.com.cu/books?id=ywSCDwAAQBAJ&pg=PA42&dq=importancia+de+los+ordenadores&hl=es&sa=X&ved=0ahUKEwiRx_bKyuDnAhWoc98KHe9HDH0Q6AEIMDAB#v=onepage&q=importancia%20de%20los%20ordenadores&f=false] Madrid : DYKINSON,SL, 2018. ISBN: 978-84-9148-987-0.

Patrones de Caso de Uso. **Cuesta, Saúl.** s.l. : SG.

Paz, Arturo Arias. 2017. Control de versiones de Software con GIT. [En línea] 2017. [Citado el: 5 de noviembre de 2019.] https://books.google.com.cu/books?id=uvw0DgAAQBAJ&printsec=frontcover&dq=que+es+un+control+de+versiones&hl=es&sa=X&ved=0ahUKEwjy3t_pkKvnAhWSmlkKHa3TCgkQ6AEIJzAA#v=onepage&q=que%20es%20un%20control%20de%20versiones&f=false. 978-1544105536.

Peiso, Manuel Enrique Cruz. 2018. *Procedimiento de construcción de imágenes de la personalización de GNU/Linux.* Universidad de las Ciencias Informáticas. La Habana : s.n., 2018. Trabajo final presentado en opción al título de Máster en Informática Avanzada.

Peñalver, Gladys Marsi Romero. 2017. *Ingenierias Ágiles.* Universidad de las Ciencias Informáticas. La Habana : s.n., 2017. Conferencia.

PEP8. 2015. PEP 8-- Style Guide for Python Code. [En línea] 1 de agosto de 2015. [Citado el: 26 de marzo de 2019.] https://www.python.org/dev/peps/pep-0008/.

Pérez, Damián Alfonso. 2015. *Técnica para el diagnóstico de variantes de procesos de negocio.* Universidad de las Ciencias Informáticas. La Habana : s.n., 2015. Tesis presentada en opción al título de Máster en Informática Aplicada.

Pérez, Juan Fernández. 2015. *La oportunidad del software libre : capacidades, derechos e innovación.* [https://books.google.com.cu/books?id=2R-M68dD-

foC&pg=PA11&dq=software+libre&hl=es&sa=X&ved=0ahUKEwjT3421i5zmAhWm1FkKHdAjAI4Q6AEINDAC#v=onepage&q=software%20libre&f=false] Valencia : Escuela de negocio, 2015. ISBN: 849-87-41-1.

Pilar, María Soler. 2011. programación web. [En línea] 21 de mayo de 2011. [Citado el: 22 de febrero de 2017.] http://progpagweb.blogspot.com/2011/05/definicion-y-caracteristicas-de-php-con.html.

Piñeiro, Javier Cárdenas. 2015. *Herramienta web para la creación de personalizaciones de Nova Servidores.* Universidad de las Ciencias Informáticas. La Habana : s.n., 2015. Tesis de fin de curso para optar como ingeniero en Ciencias Informáticas.

Pivotal. 2017. RabbitMQ is the most widely deployed open source message broker. [En línea] 23 de mayo de 2017. [Citado el: 27 de marzo de 2019.] https:// rabbitmq/what-this/.

Polanco, Manuel Ponce. 2012. *Sistema de control de gestión.* Lima, Perú : s.n., 2012.

Pons, Nicolas. 2018. *Linux: principios b'asicos de uso del sistema.* Barcelona : Ediciones ENI, 2018. ISBN 978-2-409-00547-3.

Potencier, Fabien, Weaver, Ryan and Eguiluz, Javier. 2014. *Buenas prácticas oficiales de Symfony.* 2014.

Pressman, Roger S. 2010. *INGENIERÍA DEL SOFTWARE. UN ENFOQUE PRÁCTICO.* s.l. : McGRAW-HILL, 2010. págs. 383 - 404. ISBN:978-607-15-0314-5.

—. 2010. *Ingeniería del software. Un enfoque práctico.* s.l. : Editores S.A, 2010. 978-0-07-337597-7.

Pressman, RogerS. 2010. *Ingeniería de Software. Un enfoque práctico.* 2010.

Producción, Dirección Técnica de la Producción Dirección Técnica de la. 2010. *Programa de mejora. Estandar de codificación para Python. Universidad de las Ciencias Informaticas.* 2010.

PYTHONES. 2019. Qué es Python- Definición, características y ventajas. [En línea] 20 de julio de 2019. [Citado el: 1 de octubre de 2019.] https://pythones.net/que-es-python-y-sus-caracteristicas/#.

Queue., Celery: Distributed Task Queue. 2011. Homepage | Celery: Distributed Task. 2017. [En línea] 2017.

Quimera, Felipe. 2015. Patrones de diseño: qué son y por qué debes usarlos. [En línea] 2015. http://www.genbetadev.com/metodologias-de-programacion/patrones-de-diseno-que-son-y-por-que-debes-usarlos..

Rafael Martinez. 2015. PostgreSQL . [En línea] 2 de octubre de 2015. [Citado el: 25 de octubre de 2018.] www.postgresql.org.

Ramírez, Jose Gabriel Espinosa. 2018. *Sistema para la detección de roles sobre colecciones de tuits.* Universidad de las Ciencias Informáticas. 2018. Título para optar por el título de Ingeniero en Ciencias Informáticas.

Ramos, César Vinuesa. 2015. *El auge y la importancia de las redes sociales en el mundo actual.* 2015.

Ramos, Frank Ernesto Sendiña. 2017. *Herramienta para consumir paquetes de repositorios comprimidos en la distribución cubanadeGNU/Linux Nova.* Universidad de las Ciencias Informáticas. La Habana : s.n., 2017. Título para optar como Ingeniero en Ciencias Informáticas.

Raya Cabrera, José Luis, Raya González, Laura y Zurdo, Javier S. 2014. *Sistemas Informáticos (GRADO SUPERIOR).* Madrid : Ra-Ma, 2014. ISBN 978-84-9964-349-6.

Richarte, Javier. 2018. Servicio Técnico 05: El BIOS: Curso visual y práctico: PCS • NOTEBOOKS ... [En línea] 2018. [Citado el: 3 de octubre de 2019.] https://books.google.com.cu/books?id=8jxLDwAAQBAJ&pg=PA14&dq=ventajas+de+utilizar+UEFI&hl=es&sa=X&ved=0ahUKEwjWy4SZ1JjmAhWhtVkKHbO7CicQ6AEIJzAA#v=onepage&q=ventajas%20de%20utilizar%20UEFI&f=false. ISBN 949-4-571-852-1.

Rockcontent. 2019. *¿Qué es un lenguaje de programación y qué tipos existen? .* [En línea] 25 de julio de 2019. [Citado el: 12 de octubre de 2019.] https://rockcontent.com/es/blog/que-es-un-lenguaje-de-programacion/.

Rodríguez, Tamara Sánchez. 2015. *Metodología UCI.* Universidad de las Ciencias Informáticas. La Habana : s.n., 2015. Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas.

Rohaut, Sébastien. 2017. Linux: dominar la administración del sistema. [En línea] 3, 2017. [Citado el: 5 de diciembre de 2019.] https://books.google.com.cu/books?id=1P2BgIuvLFsC&pg=PA284&dq=UEFI/GPT&hl=es&sa=X&ved=0ahUKEwi8sJ7J1pjmAhXIxVkKHXDwAkcQ6AEIKjAA#v=onepage&q=UEFI%2FGPT&f=false. ISBN 978-2409-01222-8.

Rojas, Pablo Ramírez. 2012. Metodologías Tradicionales vs. Metodologías Ágiles. [En línea] 6 de febrero de 2012. [Citado el: 26 de noviembre de 2016.] http://www.mygnet.net/manuales/software/metodologias_tradicionales_vs_dot_metodologias_agiles.

Rose, Heirald. 2014. [En línea] 2014.

Rothman, Michael y Zimmer, Vincent. 2017. *Harnessing the UEFI Shell: Moving the Platform Beyond DOS, Second Edition.* Berlin : CPI books GmbH, Leck, 2017. ISBN 978-1-5015-1480-7.

Salazar, Patricia. 2016. *El perfil de usuario de informacion.* 2016.

Sánchez, Manuel Alejandro del Campo. 2015. *PERSONALIZACIÓN DE GNU/LINUX NOVA PARA ELSISTEMA DE EDUCACIÓN PRIMARIA EN CUBA.* Universidad de las Ciencias Informáticas. La habana : s.n., 2015. Trabajo de Diploma para Optar por el Título de Ingeniero en Ciencias Informáticas.

Sánchez, Manuel Alejandro. 2015. *Personalización de GNU/Linux Nova para el Sistema de Educación Primaria de Cuba.* Universidad de las Ciencias Informáticas. La Habana : s.n., 2015. Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas.

Santamaría, Pedro. 2015. Balsamiq mockup, una muy buen herramienta para esbozar tus futuras apps . *Applesfera.* [En línea] 25 de noviembre de 2015. [Citado el: 1 de noviembre de 2019.]

https://www.applesfera.com/aplicaciones-os-x-1/balsamiq-mockup-una-muy-buen-herramienta-para-esbozar-tus-futuras-apps.

Santana, Alberto. 2017. Sitios de redes sociales en Internet. [En línea] 20 de 07 de 2017. [Citado el: 05 de 11 de 2018.]

Sarmah, Harshajit. 2019. analyticsindiamag. *5 Tools that will help you create your on Linux Distro.* [En línea] 2019. [Citado el: 3 de diciembre de 2019.] https://analyticsindiamag.com/5-tools-that-will-help-you-create-your-own-linux-distro/.

Sarmiento, Johana. 2016. Vision General de los diagramas de despliegue. [En línea] 22 de 03 de 2016. [Citado el: 14 de 01 de 2019.] http://umldiagramadespliegue.blogspot.com/..

Schmuller, Joseph. 2015. 2015.

SIAC. 2018. Sistema Contable Administrativo. [En línea] 21 de octubre de 2018. [Citado el: 22 de octubre de 2019.] http://sistemacontableec.com/sistema-contable-administrativo/.

Silega, Nemury. 2017. *Método para la trasformación automatizada del modelo de procesos de negocio a modelo de componentes para sistemas de gestión empresarial. Tesis presentada en opción al grado científico de Doctor en Ciencias Técnicas.* Universidad de las Ciencias Informáticas. 2017. Tesis de Maestria.

Smith, Roderick W. 2011. CompTIA Linux+ Complete Study Guide Authorized Courseware: Exams LX0-101 and ... [En línea] 2011. [Citado el: 4 de diciembre de 2018.] https://books.google.com.cu/books?id=xa2-b3bsRMcC&pg=PA140&dq=GNU+pARTED&hl=es&sa=X&ved=0ahUKEwiI0Mfe55nmAhVLnlkKHQZHCJEQ6AEIRDAD#v=onepage&q=GNU%20pARTED&f=false. ISBN 978-0-470-88845-2.

Sobell, Mark G. 2014. A Practical Guide to Ubuntu Linux: A Practical Gui Ubun Linu_p4. [En línea] 2014. [Citado el: 24 de marzo de 2019.] https://books.google.com.cu/books?id=BOzlBQAAQBAJ&q=GNOME+DISK&dq=GNOME+DISK&hl=es&sa=X&ved=0ahUKEwi914u615nmAhWorFkKHXXXC_gQ6AEIRDAD. ISBN 945-479-4-103-4.

SOFTWAREPROG. 2018. Entorno de Desarrollo Integrado. [En línea] 2018. [Citado el: 2 de septiembre de 2019.] https://sites.google.com/site/softwaredeprogramacion2/entorno-de-desarrollo-integrado.

Sommerville, Ian. 2011. *Ingeniería de Software.* Madrid : Pearson Addison Wesley, 2011. ISBN: 84-7829-074-5.

—. 2005. *Ingeniería del Software.* Madrid : Pearson Addison Wesley, 2005. 84-7829-074-5.

SPRINGER. 2018. Visual Studio Code Destilled: Evolved Code Editing for Windows, masOs, and Linux. *Introducing Visual Studio Code.* [En línea] 2018. [Citado el: 10 de noviembre de 2019.] http://link.springer.com/10.1007/978-1-4842-4224-7. DOI 10.1007/978-1-4842-4224-7.

Thornton, J. 2013. *Bootstrap3 Manual Official, 2013 [Citado el: 28 de noviembre de 2018.] Disponible en: http://conocimientoabierto.es.* 2013.

TICARTE. 2014. *Herramientas de compresión.* [En línea] 12 de septiembre de 2014. [Citado el: 28 de octubre de 2019.] http://www.ticarte.com/contenido/herramientas-de-compresion.

Torrecilla, Pablo. 2012. 2012.

TributosNet. 2016. Inventarios. [En línea] 11 de julio de 2016. [Citado el: 2019 de octubre de 15.] http://www.tributos.net/definicion-de-gestion-de-inventarios-1013/.

UBUNLOG. 2017. *Distroshare: un script que te ayuda a crea tu propia imagen de Ubuntu.* [En línea] 14 de noviembre de 2017. [Citado el: 5 de septiembre de 2019.] https://ubunlog.com/author/darkcrizt/.

UCC. 2017. Visión general del Diagrama de despliegue. [En línea] 24 de septiembre de 2017. [Citado el: 1 de octubre de 2019.] http://umldiagramadespliegue.blogspot.com/..

UML. 2017. What is UML. [En línea] 2017. [Citado el: 24 de octubre de 2019.] https://www.uml.org/what-is-uml.htm.

Universidad de Mexico. 2015. Gestión de Software. [En línea] 1 de marzo de 2015. [Citado el: 2016 de 10 de 15.] http://mejoratugestion.com/mejora-tu-gestion/que-es-un-sistema-de-gestion/.

Urrutia, Angélica. 2015. Análisis de Información de Redes Sociales (Twitter). [En línea] 17 de 09 de 2015. [Citado el: 07 de 12 de 2018.]

Vázquez, Eduardo Carlo. 2018. *Ingeniería de Requisitos: Software Orientado al Negocio.* [https://books.google.com.cu/books?id=kDjMvwEACAAJ&dq=especificaci%C3%B3n+de+requisitos+del+software&hl=es&sa=X&ved=0ahUKEwjslPDAwKznAhUro1kKHZvTDEEQ6AEIPDAD] 2018. ISMN:1729136680.

Velazquez, Luis Romero. 2015. *Analisis y diseño orientado a objetos.* Madrid : Lenguage Books, 2015. ISBN: 479-587-41-1.

Warsaw, Barry y Coghlan, Nick. 2013. Style Guide for Python Code. [En línea] 5 de julio de 2013. [Citado el: 22 de septiembre de 2019.] https://www.python.org/dev/peps/pep-0008/.

WOLF, Gunnar, y otros. 2015. *Fundamentos de sistemas operativos.* Mexico D.F : Círculo Mario de la Cueva, 2015. ISBN 978-607-02-5544-0.

ZAZO RODRÍGUEZ, Ángel F., ALONSO BERROCAL, José Luis y FIGUEROLA, Carlos G. 2014. *Herramientas de software libre para el trabajo científico colaborativo: EN ..* [https://books.google.com.cu/books?id=78mMAwAAQBAJ&pg=PA149&dq=software+libre&hl=es&sa=X&ved=0ahUKEwjT3421i5zmAhWm1FkKHdAjAI4Q6AEITzAG#v=onepage&q=software%20libre&f=false] Salamanca : Universidad de Salamanca, 2014. 947-58-7-548-624-08225.

ANEXOS

Anexo 1: Entrevista realizada a especialistas del proyecto Nova

Tipo de entrevista: Estructurada porque se tiene un orden lógico de lo que se quiere preguntar, es decir se conoce el objetivo y finalidad de la entrevista.

Cantidad de personas entrevistadas: 4

Nombre de los entrevistados: Juan Manuel Fuentes Rodríguez, Aldy León García, Javier Piñeiro Cárdenas, Luis Daniel Sierra Corredera.

Función de los entrevistados en el proyecto: presentan más de 5 años de experiencia, han trabajado directamente con el proyecto.

Objetivo de la entrevista: conocer cómo se realiza el proceso de instalación cifrada en la distribución cubana GNU/Linux NOVA.

1. ¿Cree usted que es importante cifrar la información en Nova? ¿Por qué?
2. ¿Cuáles son los aspectos fundamentales a tener en cuenta para realizar una instalación cifrada de Nova?
3. ¿Cuáles son los pasos a seguir para realizar una instalación cifrada de Nova?
4. ¿Cree necesario la creación de una herramienta que permita realizar este tipo de instalación de manera automática? ¿Por qué?
5. ¿Cuáles serían las características que tendría esta herramienta?
6. ¿Qué bibliografía y lenguaje de programación me recomendaría para ampliar mis conocimientos acerca de este tema?

Resultado de la entrevista: se evidencia que en la actualidad el proceso de instalación cifrada de Nova se realiza mediante un procedimiento manual, carecen de una herramienta informática que automatice el proceso.

Anexo 2: Guía de observación para el proceso de instalación cifrada de la distribución cubana GNU/Linux Nova.

Observador: Alejandro Guilarte Larin

Lugar: Laboratorio del proyecto Nova

Objetivo: Identificar los requisitos fundamentales para la realización del proceso de instalación cifrada de la distribución cubana GNU/Linux Nova.

1. Datos de identificación del proceso
    - Nombre del proceso
    - Especialista que ejecuta el proceso
2. Características del espacio donde se desarrolló el proceso
    - ¿Dónde se debe realizar?
    - ¿Bajo qué condiciones se realiza?
    - ¿Qué se necesita para iniciar el proceso?
3. Carácterísticas del proceso
    - ¿Cómo es el proceso?
    - ¿Qué herramientas se emplean?
    - ¿Cuáles son los pasos a seguir?
    - ¿Cuál es el resultado?

Anexo 3: Descripción de las historias de usuarios

Tabla 10.RF2. Eliminar información existente

(Fuente: elaboración propia)

Número: HU – 2	**Nombre del requisito: Eliminar información existente.**
Programador: Alejandro Guilarte Larín	**Iteración asignada:** 1

Prioridad: baja	**Tiempo estimado:** 2
Riesgo en desarrollo: medio	**Tiempo real:** 1

Descripción: luego de ejecutar la herramienta, esta debe eliminar toda la información el disco en donde se realizará la instalación del sistema operativo.

Se listan a continuación las funcionalidades requeridas:

- Pedir confirmación al usuario de la eliminación de la información contenida.
- Enumerar todos los discos disponibles.
- Mostrar capacidad de todos los discos disponibles.
- Mostrar marca del fabricante de todos los discos disponibles.

Prototipo de interfaz:

```
125  parts="efi=100M boot=2G lvm=-1MB swap=${totalRAM} root=32G home=100%"
126  for part in $parts
127  do
128      name=$(cut -f1 -d= <<< $part)
129      [ "$name" == "efi" ] && ! isEFI && continue
130      [ ${!name} ] || eval "${part}"
131  done
132  grep -q "%" <<< ${home} || home="${home}%"
133
```

Tabla 11.RF3. Crear nueva tabla de particiones

(Fuente: elaboración propia)

Número: HU – 3	**Nombre del requisito: Crear nueva tabla de particiones.**
Programador: Alejandro Guilarte Larín	**Iteración asignada:** 1
Prioridad: media	**Tiempo estimado:** 4
Riesgo en desarrollo: medio	**Tiempo real:** 2

Descripción: La herramienta debe especificar al sistema operativo, el número y el tipo de particiones que tiene nuestro sistema. Además de indicarle al sistema dónde se encuentra la partición de arranque.

Se listan a continuación las funcionalidades requeridas:

- Enumerar todos los discos disponibles.
- Mostrar capacidad de todos los discos disponibles.

- Mostrar marca del fabricante de todos los discos disponibles.
- Mostrar tipo de tabla de particiones.
- Mostrar la partición reservada para el Sistema Operativo
- Mostrar la partición reservada para el arranque del sistema.
- Mostrar la partición reservada para la Swap.
- Mostrar la partición reservada para los datos del usuario.

Prototipo de interfaz:

```
dd if=/dev/zero of=$disk bs=1M count=10 2> /dev/null
if isEFI; then
    tableType='gpt'
else
    tableType='msdos'
fi
parted $disk mktable $tableType > /dev/null 2>&1

# get information about desired sizes
totalRAM=$(cat /proc/meminfo | head -n1 | grep -oP "\d+.*" | tr -d ' B' | tr 'a-z' 'A-Z' | numfmt --from iec --to iec --format "%.f")
read -p "Size for /boot [2G]: " boot
isEFI && read -p "Size for /boot/efi [100M]: " efi
read -p "Size for LVM [remaining disk space]: " lvm
read -p "Size for swap in LVM [$totalRAM]: " swap
read -p "Size for / (root) in LVM [32G]: " root
read -p "Percent of remaining LVM space to use for /home [100%]: " home
echo
while :
do
    echo "Nothing will be displayed as you type passphrases!"
    read -sp "Encryption passphrase: " luksPass && echo
    [ "$luksPass" == "" ] && echo "Oops, looks like you forgot to provide a passphrase. Try again." && continue
    read -sp "Confirm encryption passphrase: " confirm
    clear
    [ "$luksPass" == "$confirm" ] && break
echo "passphrases didn't match or passphrase was blank! Try again"
done
echo  -e 'In addition to the passphrase you provided, a keyfile can be generated that can \nalso be used for decryption. It is STRONGLY RECOMMENDED that y
read -p "Key file size in bytes, or 'none' to prevent key file creation [512]: " keyfileSize
keyfileSize=${keyfileSize:-512}
keyfile=/tmp/LUKS.key
hasKeyfile && dd if=/dev/urandom of="${keyfile}" bs=${keyfileSize} count=1 2> /dev/null

```

Anexo 4: Diseños de Casos de Pruebas

Tabla 12 Diseño de Caso de Prueba: Eliminar información existente

(*Fuente: elaboración propia*)

Escenario	Descripción	Respuesta de la aplicación	Flujo central
EC2 Eliminar información existente	La aplicación al ser ejecutada debe eliminar la información contenida en el disco en que se desea instalar el sistema operativo.	El disco seleccionado sufre un proceso de eliminación de toda la información contenida.	El usuario debe confirmar que entiende que puede perder informacion contenida en el disco a seleccionar y presionar enter El sistema elimina toda la información contenida en el disco duro. El sistema muestra el estado resultante de la operación. El usuario presiona enter para continuar.

Tabla 13 Diseño de Caso de Prueba: Crear nueva tabla de particiones

(*Fuente: elaboración propia*)

Escenario	Descripción	Respuesta de la aplicación	Flujo central
EC3 Crear nueva tabla de particiones	La aplicación al ser ejecutada debe especificar al sistema operativo, el número y el tipo de particiones que tiene nuestro sistema. Además de indicarle al sistema dónde se encuentra la partición de arranque.	En el disco duro se generan 4 particiones. La partición de arranque, la partición de datos del sistema operativo, la partición de la swap y la partición de datos del usuario.	El sistema crea una partición destinada al arranque. El sistema crea una partición destinada al sistema operativo. El sistema crea una partición destinada a la Swap del sistema operativo. El sistema crea una partición destinada a los datos del usuario. El sistema muestra muestra como queda conformada la nueva tabla de particiones El usuario presiona enter para continuar.

Tabla 14 Diseño de Caso de Prueba: Cifrar disco duro

(*Fuente: elaboración propia*)

Escenario	Descripción	Respuesta de la aplicación	Flujo central
EC4 Cifrar disco duro	La aplicación al ser ejecutada debe cifrar el disco duro completamente exceptuando la partición de arranque.	Cifra las particiones destinadas al sistema operativo,a la swap del sistema y a los datos del usuario.	El sistema le da formato ext4 a todas las particiones. El sistema instala la herramienta de cifrado Dm-crypt / Luks. El usuario ingresa la clave de cifrado. El sistema cifra todas las particiones del disco, exceptuando la partición de arranque. El usuario presiona enter para continuar.

Tabla 15 Diseño de Caso de Prueba: Instalar sistema operativo

(*Fuente: elaboración propia*)

Escenario	Descripción	Respuesta de la aplicación	Flujo central
EC5 Instalar sistema operativo	La aplicación al ser ejecutada debe instalar el sistema operativo GNU/Linux Nova	Inicia la instalación del sistema operativo Nova.	El sistema inicia la instalación del sistema operativo Nova en la partición reservada para ello.