

**Universidad de las Ciencias Informáticas
“Facultad 6”**



**PGST: “Módulo para la gestión de configuración del
equipamiento tecnológico en la Universidad de las
Ciencias Informáticas”**

Trabajo de Diploma para optar por el título de
Ingeniero Informático

Autor(es): Danays Rodriguez Martinez

Marilys Valiente González

Tutor(es): Ing. Orlando Cruz Rojas

Ing. Leandro Evangelio Hernández Cuello

Co-tutor: Ing. Joan Martínez Herrera

“Ciudad de la Habana, Mayo 2010”

"La inteligencia consiste no solo en el conocimiento, sino también en la destreza de aplicar los conocimientos en la práctica".

Aristóteles

DECLARACIÓN DE AUTORÍA

Declaramos ser autoras del presente trabajo de diploma y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Danays Rodriguez Martinez

Marilys Valiente González

Firma de la Autora

Firma de la Autora

Ing. Orlando Cruz Rojas

Ing. Leandro Hernández Cuello

Firma del Tutor

Firma del Tutor

DATOS DE CONTACTO

Nombre y Apellidos del Tutor: Orlando Cruz Roja
Institución: Universidad de las Ciencias Informáticas
Dirección de Gestión Tecnológica
Teléfono: 835 8347
E-mail: landy@uci.cu

Currículum: Graduado de Ingeniero en Ciencias Informáticas en el 2009. Se desempeña como Especialista de la Dirección de Gestión Tecnológica. Se le otorgó en el año 2008 la medalla de forjadores del futuro. Tiene una experiencia de 15 años trabajando en esta rama. Ha participado en fórums y eventos científicos obteniendo resultados relevantes.

Nombre y Apellidos del Tutor: Leandro Evangelio Hernández Cuello
Institución: Universidad de las Ciencias Informáticas
Departamento de Sistemas Especializados
Teléfono: 837 2228
E-mail: lehernandez@uci.cu

Currículum: Graduado de Ingeniero en Ciencias Informáticas en el 2009. Su desempeño profesional en el curso 2009-2010 estuvo vinculado al Proyecto Ensayos Clínicos desempeñando el rol de arquitecto y desarrollador.

Nombre y Apellidos del Co-tutor: Joan Martínez Herrera
Institución: Universidad de las Ciencias Informáticas
Centro de Telemática, Facultad 2
Teléfono: 835 8143
E-mail: jherrera@uci.cu

Currículum: Graduado de Ingeniero en Ciencias Informáticas en el 2008. Se desarrolla actualmente como jefe de programadores, jefe de módulo del desarrollo de mensajería WAP y analista principal dentro del proyecto Servidor de Aplicaciones WAP (SERWAP) dentro del Centro de Telemática de la Facultad 2 en la Universidad de las Ciencias Informáticas. Ha tutorado varias tesis desde su

graduación. Resumen de Tesis que ha tutorado: Cotutor de Tesis de Grado. Plataforma de Servicios de Valor Agregado: Servicio de Suscripciones Móviles. (2008-2009.)

AGRADECIMIENTOS

A mi mamá, por ser lo más grande que tengo en este mundo, por estar siempre a mi lado y darme fuerzas para seguir.

A mi abuela, por ayudar a criarme y darme siempre tan buenos consejos.

A mi novio por darme tanta seguridad, por su comprensión, su compañía y por siempre estar ahí.

A mis tutores Landy y Leandro por la tutoría de este trabajo. Para ustedes mis más sagrados agradecimientos por su calidez, sugerencias y confianza.

A Landy un agradecimiento especial por apoyarme y guiarme durante todo este largo camino.

A mi compañera de tesis por soportarme durante este tiempo y tener paciencia, ha sido duro el camino.

A mi tía por estar al tanto y ofrecerme su apoyo, por sus palabras de firmeza en los momentos que lo necesité.

A Li, sin su ayuda no hubiese podido siquiera comenzar esta carrera. Por ser más que una amiga, por ser mi hermanita de alma, por quererme y respetarme tanto.

A Joan por sus conferencias de pasillo cada vez que lo necesité, por su asesoramiento y experiencia.

A Aida porque sin su apoyo no hubiese sido posible la realización de este trabajo.

A Norbe del grupo de desarrollo de la DGT por su gran aporte y ayuda.

A todos los integrantes del equipo del CENIA que aportaron su granito de arena para que el trabajo saliera a flote. Especialmente a Ronny por las ideas y conocimientos que le brindó al trabajo. A Yanet Poza e Ingrid por la metodología.

A mis compañeros de trabajo que siempre me apoyaron, en especial a Ine, Eri, Iri, Linne.

A Julián por ser más que un profe, un amigo y por brindarme siempre su ayuda incondicional.

A Aquiles por sus buenos manuales y siempre tener un tiempo para cualquier duda.

A todos los profe que me han permitido obtener los conocimientos necesarios para poder realizar este trabajo.

A la Universidad y la Revolución por haberme dado la oportunidad de realizar mis estudios.

En fin, a todos mis amigos, familia, amistades de la UCI y del barrio que siempre me preguntaban ¿cómo va esa tesis?

A todos, Muchas Gracias

Danays.

A mi madre por ser mi apoyo incondicional en todo momento, y por estar siempre a mi lado cuando más lo necesité.

A mi hermana, la persona que más quiero en este mundo, por ser mi luz y mi guía.

A mi padre por ser mi fuente de inspiración, fue muy difícil seguir tu ejemplo.

A mi novio Alexis por comprenderme y brindarme su apoyo constante.

A mi compañera de tesis por haber sido de las dos la más audaz y segura.

A mi tía Margot por haberme ayudado a recobrar la fe y seguridad en los momentos en que se pensaba que no íbamos a salir afuera.

A mis compañeros de aula, en especial a Jany y Yusimil por su preocupación hacia nuestro trabajo.

A un trío de profesoras excepcionales: Vilmavis La Rosa, Aidacelys López, Martha Denia.

A Aidacelys por ser más que una amiga, sin su abnegada labor hubiese sido imposible poder realizar este sueño.

A nuestros tutores por ser nuestros principales consejeros y guías encaminándonos hacía el éxito.

A Joan por haber aportado todo su conocimiento para que el trabajo saliera a flote.

A Landy por ser el apoyo fiel y constante que necesitamos para poder continuar, por habernos mostrado el camino y habernos conducido siempre de la mano.

A Roque por habernos adiestrado con el tema de la investigación, mis más sinceros agradecimientos.

A la abnegada labor del claustro de profesores de la Facultad 6

A mis compañeros de trabajo por haberme soportado durante todo este tiempo.

A la Universidad y a la Revolución por habernos hecho personas mejores.

A la mamá y a la tía de Danays por habernos dado aliento en los momentos más difíciles.

A todas aquellas personas sin nombre, que durante todo este tiempo de nuestras vidas han aportado un granito de sabiduría y amor contribuyendo a que pudiéramos realizar este sueño, para todas esas personas mis más sentidos agradecimientos, y que sepan que los llevamos en el corazón.

A todos mil Gracias.

Sinceramente Marilyns

DEDICATORIA

Dedicado especialmente a mi abuela que siempre esperó ver este sueño hecho realidad y hoy le hago este regalo, y más aún a la persona más importante en mi vida, mi mejor amiga, mi compañera de vida, mi primera maestra, mi apoyo incondicional, mi amor infinito... mi mamá.

Danays.

Especialmente a mi hermana por haber sido mi tutora en la vida y haberme aportado tantas enseñanzas que han sido un ejemplo inigualable a seguir.

A mi madre, mi fuerza vital en la vida.

Marilys.

RESUMEN

El trabajo tiene como objetivo desarrollar e implementar un módulo de la Plataforma de Gestión de Servicios Tecnológicos (PGST) para la gestión de configuración del equipamiento tecnológico en la Dirección de Gestión Tecnológica (DGT) de la Universidad de las Ciencias Informáticas (UCI).

El módulo tiene como propósito automatizar la información manejada por el especialista de la DGT para confeccionar las configuraciones del equipamiento tecnológico que necesita cada local de la universidad en dependencia de las funciones que realizan, así como facilitar una mejor organización, control y distribución del equipamiento tecnológico.

Se realizó una investigación de todos aquellos requisitos que se deben tener en cuenta en vistas a ofrecer una mejor organización de las actividades que se efectúan en esta área.

Se utilizaron las tecnologías definidas en el documento de arquitectura del Centro de Informatización Universitaria (CENIA). Está concebido para funcionar en un ambiente de integración de sistemas informáticos brindando y consumiendo Servicios Web.

PALABRAS CLAVES: Configuración, equipamiento tecnológico.

TABLA DE CONTENIDOS

AGRADECIMIENTOS	V
DEDICATORIA	VII
RESUMEN	VIII
INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	5
1.1 ¿Qué es ITIL?	5
1.2 ¿Qué es Gestión de Configuración?	5
1.3 ¿Qué es Administración de Configuración?	5
1.4 Configuraciones existentes en la UCI.....	6
1.5 Sistemas de Gestión de Configuración existentes.....	6
1.5.1 Aranda Assets Management	7
1.5.2 Assyts	7
1.5.3 iTop Portal Operacional.....	7
1.5.4 BMC Atrium CMBD	7
1.6 Tendencias, tecnologías y herramientas actuales	8
1.7 Lenguaje de modelado.....	10
1.7.1 UML	10
1.8 Herramientas CASE	10
1.8.1 Visual Paradigm	11
1.9 Framework CodeIgniter.....	12
1.10 Entorno de Desarrollo	12
1.10.1 Zend Studio.....	12
1.11 Lenguaje de Programación	13
1.11.1 PHP.....	13
1.12 Servidor Web Apache	14
1.13 Sistema Gestor de Base de Datos	14
1.13.1 PostgreSQL.....	15
Conclusiones	15

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	16
2.1 Objeto de Automatización	16
2.2 Flujo Actual del Proceso.....	16
2.3 Modelo Historia de Usuario del Negocio.....	16
2.4 Actor del Negocio.....	17
2.5 Trabajadores del Negocio	17
2.6 Diagrama de Historia de Usuario del Negocio.....	18
2.7 Lista de Reserva del Producto	19
2.7.1 Funcionalidades del sistema	19
2.7.2 Requisitos no funcionales.....	21
2.8 Plan de Release (Entrega)	22
2.9 Actores del sistema.....	23
2.10 ¿Qué es un patrón?	24
2.11 Historias de Usuarios	25
Conclusiones	35
CAPÍTULO 3: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.....	36
3.1 Patrones de diseño	36
3.2 Descripción de la arquitectura	38
3.2.1 Modelo Vista Controlador	39
3.3 Diseño con metáforas	40
3.4 Diagrama de Clases Persistentes	40
3.5 Modelo de datos.....	41
Descripción de las principales tablas:	42
3.6 Diagrama de componentes	46
3.7 Diagrama de despliegue	46
3.7.1 Descripción de nodos.....	47
3.9 Casos de prueba.....	50
Conclusiones	59
CONCLUSIONES	61
RECOMENDACIONES	62

REFERENCIA BIBLIOGRÁFICAS	63
BIBLIOGRAFÍA.....	65
ANEXOS.....	67
GLOSARIO DE TÉRMINOS	69

INTRODUCCIÓN

El siglo XX es considerado como un siglo con gran impacto en cuanto a avances tecnológicos a escala mundial debido a que se desarrolló el conocimiento científico-técnico en aras de darle solución a los problemas esenciales de la sociedad, trayendo consigo crecientes avances tecnológicos a escala mundial. (1)

Hoy día estos avances se aplican en casi todas las empresas que automatizan sus procesos de negocio para obtener mejores resultados. Para ello se hace necesario lograr un aprovechamiento racional de todos los recursos tecnológicos, alcanzando de esta forma una mayor asimilación de la tecnología por parte de las empresas, que le permita conocer su funcionamiento, el uso que se le está dando, así como los elementos que la componen.

Cuba no está exenta de ello ya través del Ministerio de la Informática y las Comunicaciones (MIC) hace grandes esfuerzos por informatizar la sociedad cubana vinculando todos los sectores de la sociedad al uso de las Tecnología de Información y las Comunicaciones (TIC). El empleo de la tecnología se fomenta a través del conocimiento del hardware, software y los medios de telecomunicación, que le permita obtener una mayor especialización en todos los procesos que emergen de su uso, logrando de esta forma cumplimentar la necesidad de desarrollo.

A raíz de esta necesidad y con el surgimiento de la Batalla de Ideas, en el país comienzan a desarrollarse varios programas destinados a fortalecer material y espiritualmente las conquistas del pueblo. Un ejemplo visible de esto lo constituye el Proyecto Futuro, que surge como resultado de la larga y clara visión del Comandante en Jefe. El producto inmediato de este proyecto fue la creación de la UCI como parte del programa de informatización de la sociedad cubana.

La UCI es protagonista de gran parte de los proyectos que se realizan a nivel nacional para informatizar el país y para ello cuenta con una amplia y variada infraestructura tecnológica la cual se encuentra distribuida en la universidad en dependencia de las funciones que se realizan en cada área.

Una de las áreas existentes en la universidad es la DGT, la cual tiene como misión garantizar una alta disponibilidad de equipamiento tecnológico, que proporcione a los procesos fundamentales que se ejecutan en la universidad un escenario ideal para el alcance de sus objetivos.

La DGT gestiona de forma autodidacta las configuraciones tecnológicas y sus elementos de configuración. Este proceso se realiza de forma manual y por un grupo reducido de especialistas, trayendo consigo que dicha información solo quede reflejada en documentos y/o en la memoria de estos especialistas. Además este proceso contribuye a la gestión de cambios y compras, no siendo así a la gestión de incidentes y problemas tecnológicos, entre otros, lo que trae consigo:

- Desconocimiento de las configuraciones existentes.
- Retraso en la prestación de servicios de soporte técnico.
- Demoras de entrega de información.
- Desactualización de los cambios que constantemente se efectúan.
- Limitación de la interactividad de los datos.
- Violación de los principios de seguridad, usabilidad y disponibilidad concurrente de la gran cantidad de información que constantemente se requiere manejar.

A partir de la situación problemática planteada se define el siguiente **problema científico**: ¿Cómo mejorar el proceso de gestión de configuración del equipamiento tecnológico en la UCI?

Se establece como **objeto de estudio**: El proceso de gestión de configuración del equipamiento tecnológico en la UCI.

A partir del objeto de estudio se delimita el siguiente **campo de acción**: La administración de configuración del equipamiento tecnológico en la UCI.

Con el fin de solucionar el problema anteriormente planteado se define como **objetivo general** de la investigación: Desarrollar un módulo de la Plataforma de Gestión de Servicios Tecnológicos para la gestión de configuración del equipamiento tecnológico en la DGT de la UCI, según la metodología ITIL

Este objetivo general se desglosa en los siguientes **objetivos específicos**:

- Definir las funcionalidades que tendrá el módulo de la Plataforma de Gestión de Servicios Tecnológicos.
- Diseñar el módulo de la Plataforma de Gestión de Servicios Tecnológicos.
- Implementar y probar el módulo de la Plataforma de Gestión de Servicios Tecnológicos.

Para darle cumplimiento a los objetivos específicos se trazaron las siguientes **tareas de la investigación**:

- Investigación sobre los Sistemas de Gestión de Configuración (SGC) existentes según la metodología ITIL para vincularlo a la DGT.
- Realización de un estudio del proceso de Administración de Configuración como parte del proceso de Gestión de Configuración que propone la metodología ITIL.
- Obtención de la lista de reserva del producto.
- Realización de las historias de usuario.
- Realización del modelo de diseño según la metodología de desarrollo de software seleccionada.
- Elaboración del modelo de datos.
- Realización del diagrama de despliegue.
- Realización del diagrama de componentes.
- Implementación de los componentes del módulo de la Plataforma de Gestión de Servicios Tecnológicos.
- Elaboración y realización de los casos de prueba.

Actualidad y necesidad del trabajo

En estos momentos no existe una herramienta informática que permita gestionar la configuración del equipamiento tecnológico como fue mencionado en la situación problémica, por lo que se hace necesario el desarrollo de una herramienta que mejore el proceso de gestión de configuraciones tecnológicas. Por las pautas definidas en el documento de arquitectura del CENIA y por necesidad de la DGT se realizará un módulo para la gestión de configuración del equipamiento tecnológico en formato web debido a las ventajas que presentan estas aplicaciones como la rápida visualización, estructura clara en cuanto a diseño y actualización constante de los datos que se usan para la elaboración de las páginas.

Aportes prácticos esperados

- Brindar funcionalidades que permitan tener el control de la configuración del equipamiento tecnológico.
- Proporcionar la información necesaria con rapidez y calidad a la gestión de incidencias tecnológicas a los usuarios de la comunidad UCI.

- Proveer una información clara y precisa del equipamiento tecnológico a las líneas de soporte de gestión de incidencias tecnológicas.

Estructura de los capítulos

El contenido del trabajo se distribuirá en 3 capítulos, las conclusiones generales, recomendaciones, referencias bibliográficas, bibliografía empleada, los anexos y un glosario de términos como parte del trabajo que ayudan a tener una mayor comprensión de éste.

Capítulo I. Fundamentación Teórica: El capítulo tiene como objetivo establecer las bases teóricas generales que son fundamentales para la inicialización del trabajo. Recoge conceptos y características que se hacen necesarios dominar, así como una descripción de las tecnologías a emplear para su desarrollo. Se define el lenguaje de programación a utilizar así como el Sistema Gestor de Base de Datos (SGBD). Por último se realiza un análisis detallado del estado del arte relacionado con los sistemas de gestión de configuración del equipamiento tecnológico existentes a nivel nacional e internacional.

Capítulo II. Características del Sistema: El capítulo tiene como objetivo brindar la modelación detallada y la construcción de la estructura de la aplicación para brindar una mayor comprensión del problema existente. Se definen los actores y trabajadores del negocio, se obtiene la lista de reserva del producto así como los diagramas de historiales de usuarios correspondientes.

Capítulo III. Construcción de la Solución Propuesta: El capítulo tiene como objetivo detallar todos los elementos presentes en el diseño como: el modelo de diseño, se realiza una breve descripción de los patrones de diseño utilizados, así como el diagrama de clases persistentes y el modelo de datos. Por último se representará el modelo de implementación a través de los diagramas de componentes y de despliegue.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

El presente capítulo aborda la fundamentación teórica relacionada con la confección del sistema, se realiza un estudio del estado del arte y una descripción del objeto de estudio de la investigación. Se brinda una breve panorámica sobre algunos de los sistemas informatizados existentes vinculados al campo de acción, haciendo análisis comparativo entre la solución propuesta con respecto a las soluciones ya existentes. Además se realiza una descripción de las tendencias y tecnologías a emplear en correspondencia con lo establecido en el documento de arquitectura del CENIA.

1.1 ¿Qué es ITIL?

ITIL (Information Technologies Infrastructure Library) es una metodología que reúne una serie de buenas prácticas encaminadas a obtener calidad en la prestación de un servicio de Tecnología de Información (TI). Su base se centra en una serie de procesos (gestión de incidentes, gestión de problemas, gestión de configuración, gestión de cambio) que le permiten a la organización garantizar una estrategia operativa de toda la infraestructura de TI. (2)

Aunque existen otras metodologías en el mundo para gestionar servicios de TI, se decidió optar por ITIL debido a que brinda fácil implantación en cuanto a su forma de trabajo, además de ser adaptable a cualquier organización en la gestión de la infraestructura de TI, soportando todos los procesos que en ella se realizan. En la UCI se llevan a cabo dos procesos fundamentales (producción y docencia), el servicio de soporte técnico a la tecnología utilizada para realizar estos procesos debe estar acorde a las buenas prácticas descritas por esta metodología.

1.2 ¿Qué es Gestión de Configuración?

Es uno de los procesos más importantes de ITIL que se encarga de controlar el costo que representan los activos en una organización así como su ciclo de vida, garantizando conocer a fondo toda la infraestructura de TI para así lograr mayor provecho de la misma y rapidez en la restauración del servicio de soporte técnico. (3)

1.3 ¿Qué es Administración de Configuración?

La Administración de Configuración es un elemento estratégico en la Gestión de Configuración y se encarga de identificar los elementos de configuración tales como hardware, software, documentación, etc. Además garantiza el mantenimiento y control del equipamiento tecnológico permitiendo llevar un registro actualizado de todos los cambios efectuados.

Los especialistas de la DGT encargados de gestionar las configuraciones tecnológicas definieron la realización de un repositorio donde se almacenarán los elementos de configuración de hardware y sus datos. Haciendo énfasis en representar solo las relaciones de los elementos de configuración con los locales donde éstos estén ubicados. A partir de estos requerimientos se enfocó el trabajo a realizar.

1.3.1 Objetivos de la Administración de Configuración:

- Tener un control de todos los recursos tecnológicos del área de TI y las configuraciones asociadas a los mismos.
- Interactuar con los demás procesos de ITIL.
- Monitorizar periódicamente la configuración y hacer auditorias. (4)

1.4 Configuraciones existentes en la UCI

Teniendo en cuenta que configurar es adaptar un elemento de hardware al resto de los elementos de su entorno y a las necesidades específicas de los usuarios, se definieron según reglas establecidas 6 configuraciones de equipamiento tecnológico como se muestra en la **(Figura1)**, de acuerdo a la importancia que tiene cubrir las necesidades tecnológicas de cada local de la universidad. Estas configuraciones se determinaron en dependencia de las funciones asociadas a tipos de locales, el equipamiento tecnológico y las características por las cuales el equipamiento es identificado.

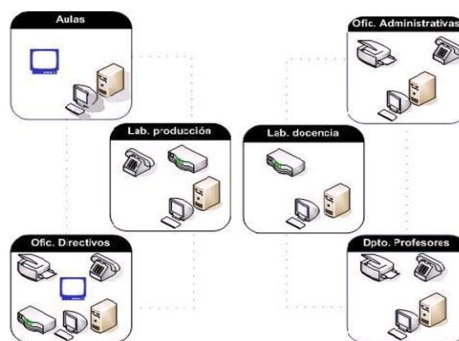


Figura 1. Configuraciones de la UCI

1.5 Sistemas de Gestión de Configuración existentes

El hombre ha ido creando herramientas cada vez mejores para facilitar su trabajo, ayudándolo a tener una buena organización, control y planificación de las actividades que realiza. Por ello ha creado sistemas que se puedan integrar a las diferentes ramas de la sociedad. Ejemplo de los sistemas que

se vinculan al campo de acción y sirven como guía para darle solución a la situación problemática podemos citar:

1.5.1 Aranda Assets Management

Este es un Software que implementa las mejores prácticas de la metodología ITIL. Le brinda al usuario un inventario actualizado del hardware, software, su estado y disponibilidad. Controla las licencias de software en cada equipo. Monitorea constantemente la configuración, localización e información histórica del uso del equipamiento. Elabora un historial de los procedimientos llevados a cabo con el equipamiento tecnológico, así como un plan de mantenimiento de éstos. Posee una Base de Datos (BD) en su repositorio central que recepciona toda la información financiera de todos los elementos de configuración inventariados. (5)

1.5.2 Assyts

Posee una poderosa BD de gestión de configuración, que ofrece un inventario preciso de los recursos de TI y las relaciones entre ellos. Administra las licencias de software. Permite definir contratos (garantía, mantenimiento, licencia de soporte y arrendamiento). Selecciona desde las opciones de consulta incluyendo análisis y ámbito o alcance del proyecto, consultas técnicas, entrenamiento, migración del producto y consultando la integración. Implementa las mejores prácticas de la metodología ITIL. (6)

1.5.3 iTop Portal Operacional

Es un portal operacional que incluye una completa BD de administración de configuración, permitiendo tener un inventario de la TI. Contiene un módulo de mesa de ayuda que opera dándoles respuestas a los usuarios. Tiene un módulo de administración de incidentes que permite dar seguimiento y comunicar todo lo acontecido con la TI. Posee un módulo de administración de cambios que informa de todos los cambios ocurridos en la infraestructura de TI. Contiene además un módulo de administración de servicios que opera con toda la documentación y con todos los contratos de servicios. (7)

1.5.4 BMC Atrium CMBD

Es un repositorio de datos inteligente que aporta una forma de trabajo que permite operar a través de toda la infraestructura de TI de una organización. Unifica todos los procesos de ITIL. Gestiona todos los tipos de elementos de configuración, así como los sistemas, aplicaciones, relaciones físicas y lógicas. Ofrece un explorador de elementos de configuración y una pantalla gráfica de relaciones entre

los elementos de configuración que permite su fácil visualización y los datos asociados a los mismos.
(8)

Luego de haber realizado el estudio de los sistemas existentes se llegó a la conclusión de que no permiten gestionar la configuración del equipamiento tecnológico de acuerdo con el objetivo que se quiere obtener y que se enfoque en el proceso y ambiente para el cual se le quiere dar uso. Además de mencionar que en su mayoría son software propietario, por lo que es necesario comprar las licencias, siendo muy costosas obtener cada versión, provocando muchos gastos a la universidad. Por tanto se hace necesario la realización de un nuevo sistema que cumpla con todas las funcionalidades requeridas.

El módulo que se propone permitirá gestionar la configuración del equipamiento tecnológico permitiendo cumplir con los requerimientos de la DGT. El módulo va a integrar funcionalidades para lograr un óptimo resultado.

1.6 Tendencias, tecnologías y herramientas actuales

Teniendo en cuenta las necesidades evidenciadas en la situación problemática y las características que presenta el entorno en el que se pondrá en práctica la solución propuesta, se realizó un estudio sobre las tendencias y tecnologías actuales a emplear definidas en los lineamientos de arquitectura del CENIA para el desarrollo de sistemas informáticos en la UCI, las cuales se describen a continuación:

- Las aplicaciones Web como estándar para la Gestión de Servicios en la UCI.
- El patrón Modelo Vista Controlador (MVC), la separación de la lógica de negocios, presentación y acceso a datos, permitiendo flexibilidad y facilidad si se tienen que hacer cambios futuros.
- PHP (Hypertext Preprocessor) como tecnología aprobada para los sitios Web y portales por ser más liviano y fácil de utilizar.
- CodeIgniter como framework para el desarrollo del sitio Web donde se hosteará esta aplicación.
- PostgreSQL como SGBD. (9)

Metodologías ágiles

En cada uno de los procesos llevados a cabo para desarrollar un software se debe tener en cuenta que para mejores resultados se tiene que aplicar el uso de una metodología, facilitando una mejor

organización y satisfacciones por parte de los desarrolladores y de los clientes que esperan su producto.

Una vez definida la necesidad del uso de una metodología es muy difícil definir cuáles de las existentes para el desarrollo de software se utilizará para el trabajo. Hay que centrarse en las necesidades y solicitudes del cliente para de esta forma analizar los artefactos que se deben producir, las notaciones que se usarán, las herramientas y especialmente el control del proceso en el que se participa. Se ha definido como metodología de desarrollo de software que guiará el proceso de informatización a SXP.

¿Qué es SXP?

Es una metodología de desarrollo de software basada en SCRUM y XP, describe la dinámica del proyecto a través de historias de usuarios (HU) permitiendo incorporar a cada una de ellas los prototipos de interfaz de usuario y otros modelos auxiliares. Está compuesta por 4 fases como se muestra a continuación:



Figura 2. Fases de SXP

¿Qué es SCRUM?

- Define un marco para la gestión de proyectos.
- Indicada para proyectos con un rápido cambio de requisitos.
- El desarrollo de software se realiza mediante sprint.
- Reuniones a lo largo del proyecto.

¿Qué es XP?

- Centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software.
- Basada en continua realimentación entre el cliente y el equipo de desarrollo.

- Simplicidad en las soluciones implementadas.
- Adecuada para proyectos con requisitos imprecisos y muy cambiantes. (10)

Se decide optar por una metodología ágil debido al poco tiempo destinado a la obtención del producto y al reducido grupo de trabajo con que se cuenta. Se valora al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas. Se desarrolla un software que funcione más que conseguir una buena documentación. El entorno del sistema es muy cambiante, se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad. Además de mencionar que el CENIA trabaja con esta metodología y está dentro de sus normativas estándares de trabajo y el módulo desarrollado para la gestión de incidencias fue documentado utilizando esta metodología.

Dentro de los objetivos que persigue se encuentran:

- La no duplicación de esfuerzos por parte de los miembros del equipo de desarrollo.
- Responder a los cambios de forma rápida y precisa.
- Evitar las documentaciones extensas y lograr mayor calidad en la obtención del producto.

1.7 Lenguaje de modelado

La comunidad del software precisa de una forma de comunicar sus modelos, no solo entre los miembros de un proyecto, sino a todas las personas involucradas en él, necesita un lenguaje para proporcionar un marco en el que desarrolladores individuales puedan pensar y analizar.

1.7.1 UML

El Lenguaje Unificado de Modelado (UML) es un lenguaje gráfico para visualizar, especificar, construir, documentar y comunicar los artefactos de un sistema de software. Es utilizado para modelar la información del sistema basado en concepto de objetos. Especifica varios diagramas que permiten crear artefactos durante el proceso de desarrollo de software. (11)

1.8 Herramientas CASE

Una herramienta CASE (Computer Aided Software Engineering) es un producto computacional enfocado a apoyar una o más técnicas dentro de un método de desarrollo de software.

De acuerdo con Kendall (12) la ingeniería de sistemas asistida por ordenador es la aplicación de tecnología informática a las actividades, las técnicas y las metodologías propias de desarrollo. Su

objetivo es acelerar el proceso para el que han sido diseñadas, para automatizar o apoyar una o más fases del ciclo de vida del desarrollo de sistemas.

Acerca de las herramientas automatizadas para la ingeniería de software se puede decir que:

- Permiten un mayor control de proyectos complejos.
- Permiten reducir costos y retrasos en la liberación de un proyecto.
- Permiten una mayor comunicación en equipos de trabajo.
- Ayudan a determinar la complejidad del proyecto y esfuerzos necesarios.

1.8.1 Visual Paradigm

Visual Paradigm es una herramienta CASE orientada a UML, soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite diseñar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

Posee entre sus principales características las siguientes:

- Generación de código e ingeniería inversa: Brinda la posibilidad de generar código a partir de los diagramas para plataformas como: .Net, Java y PHP, así como obtener diagramas a partir del código.
- Generación de documentación: Brinda la posibilidad de documentar todo el trabajo sin necesidad de utilizar herramientas externas.
- Disponibilidad en múltiples plataformas: Microsoft Windows (98, 2000, XP, o Vista), Linux, Mac OS X, Solaris o Java.
- Integración con distintos Entornos de Desarrollo Integrados (IDE): Se integra fácilmente con varios IDEs, entre ellos el de Visual Studio y el Eclipse.
- Interoperabilidad con otras aplicaciones: Brinda la posibilidad de intercambiar información mediante la importación y exportación de ficheros con aplicaciones como por ejemplo Visio y Rational Rose. Además permite importar y exportar XML (Extensible Markup Language). (13)

Se ha decidido emplear Visual Paradigm teniendo en cuenta las características anteriormente mencionadas y especialmente por su buena integración con PHP, lenguaje que se utilizará para desarrollar el módulo, y la ventaja que presenta una interfaz de fácil uso y que permite realizar los diagramas y artefactos que se generan durante el desarrollo del software.

1.9 Framework CodeIgniter 1.7.2

CodeIgniter es un entorno de trabajo para personas que construyen su aplicación web usando PHP. Su objetivo es permitir desarrollar proyectos mucho más rápido. CodeIgniter le permite creativamente enfocarse en su proyecto minimizando la cantidad de código necesaria para una tarea dada.

Algunas de sus características son:

- Usa patrón MVC.
- Extremadamente liviano.
- Rápido.
- Extensible.
- Clases de BD llenas de características con soporte para varias plataformas. (14)

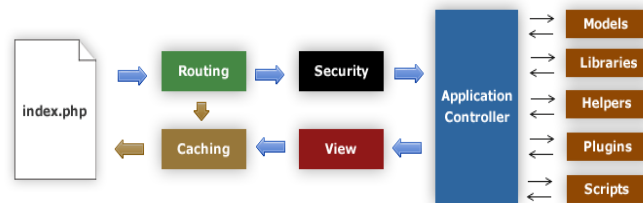


Figura 3. Flujo de datos del sistema

1.10 Entorno de Desarrollo

Un IDE (Integrated Development Environment) es un ambiente de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica de usuario. Los IDE proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación.

1.10.1 Zend Studio 6.0

Zend Studio es un IDE orientado a desarrollar aplicaciones web en lenguaje PHP. Sirve de editor de texto para páginas PHP, está escrito en Java y disponible para Microsoft Windows, Linux y MacOS. Consta de dos partes actuando Zend Studio como la parte cliente y Zend Platform como la parte servidora. Zend Studio fue diseñado para usarse con el lenguaje PHP; sin embargo ofrece soporte básico para otros lenguajes Web, como HTML, Java script y XML. (15)

Entre sus principales características se encuentran:

- No requiere la instalación previa de PHP ni del entorno de ejecución de Java.
- Soporte para PHP 4 y PHP 5.

- Resaltado de sintaxis, autocompletado de código, ayuda de código y lista de parámetros de funciones y métodos de clase.
- Detección de errores de sintaxis en tiempo real.
- Funciones de depuración: Botón de ejecución y traza, marcadores, puntos de parada, seguimiento de variables y mensajes de error del intérprete de PHP.
- Soporte para gestión de grandes proyectos de desarrollo.
- Manual de PHP integrado.
- Soporte para control de versiones usando CVS o Subversión (a elección del desarrollador).
- Cliente FTP integrado.
- Soporte para navegación en BD y ejecución de consultas SQL.

1.11 Lenguaje de Programación

Un lenguaje de programación es un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Es utilizado para controlar el comportamiento físico y lógico de una máquina.

1.11.1 PHP 5.0

PHP es un lenguaje interpretado de alto nivel embebido en páginas HTML (HyperText Markup Language) y ejecutado en el servidor. La meta de este lenguaje es permitir escribir a los creadores de páginas Web, páginas dinámicas de una manera rápida y fácil. (16)

Una de sus características más potentes es su soporte para gran cantidad de BD, entre las que se pueden mencionar InterBase, MySQL, Oracle, Informix, PostgreSQL, entre otras. Ofrece la integración con varias bibliotecas externas, que dan al desarrollador la posibilidad de realizar cualquier tarea desde generar documentos en PDF (Portable Document Format) hasta analizar código XML.

Es multiplataforma, funciona tanto para Linux (con Apache) como para Windows (con Microsoft Internet Information Server). Su sintaxis está inspirada en C, ligeramente modificada para adaptarlo al entorno en el que trabaja. Ofrece una solución simple y universal para las paginaciones dinámicas como la web de fácil programación. Su diseño elegante lo hace perceptiblemente, más fácil de mantener y ponerse al día, a diferencia con el código de otros lenguajes.

1.12 Servidor Web Apache 2.0

Apache es un servidor Web que funciona en la más amplia variedad de plataformas y entornos. Las diferentes plataformas y los diferentes entornos hacen que a menudo sean necesarias diferentes características o funcionalidades, o que en una misma característica o funcionalidad sea implementada de diferente manera para obtener una mayor eficiencia. (17)

Ofrece ventajas favorables como:

- Multiplataforma.
- Extensible.
- Potente.
- Gratuito.

1.13 Sistema Gestor de Base de Datos

Un SGBD es un conjunto de programas que permite crear y mantener una BD asegurando su integridad, confidencialidad y seguridad. Por tanto debe permitir:

- Definir una Base de Datos (BD): especificar tipos, estructuras y restricciones de datos.
- Construir la BD: guardar los datos en algún medio controlado por el mismo SGBD.
- Manipular la BD: realizar consultas, actualizarla, generar informes. (18)

Los principales objetivos que debe cumplir un SGBD son:

- Seguridad: Los SGBD deben disponer de un sistema de permisos a usuarios y grupos de usuarios, que permitan restringir los accesos no autorizados.
- Integridad: Se debe proteger la información almacenada ante fallos de hardware, datos introducidos por descuido o cualquier situación que pueda alterar la integridad de los datos almacenados.
- Tiempo de respuesta: Se debe minimizar el tiempo que el SGBD tarda en proporcionar la información solicitada y en guardar los cambios efectuados.
- Abstracción de la información: Los usuarios deben ser ajenos a detalles sobre la localización física donde han sido almacenados los datos.
- Redundancia mínima: Un diseño correcto de una BD debe evitar la duplicación o redundancia de la información.

- Independencia: Debe ser posible modificar la BD sin necesidad de realizar cambios en las aplicaciones que la accedan.

1.13.1 PostgreSQL 8.1

PostgreSQL es un SGBD orientado a objetos de software libre, publicado bajo la licencia Berkeley Software Distribution. (19)

Características fundamentales de PostgreSQL:

- DBMS Objeto-Relacional: PostgreSQL aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multiversión, soporte multiusuario, transacciones, optimización de consultas, herencia y arreglos.
- Altamente extensible: PostgreSQL soporta operadores, métodos funcionales de acceso y tipos de datos definidos por el usuario.
- Integridad referencial: PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la BD.
- Lenguajes procedurales: PostgreSQL tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL. Otra ventaja de PostgreSQL es su habilidad para usar Perl, Python, o TCL como lenguaje procedural embebido.

Conclusiones

En este capítulo se abordaron conceptos relacionados con el objeto de estudio de la investigación que se hacen necesarios dominar para lograr una mayor comprensión del tema. Se expusieron las tendencias y tecnológicas a emplear para el desarrollo del módulo definidas por el CENIA. Se realizó un estudio del estado del arte, tanto en el ámbito nacional como internacional, quedando plasmadas las características de los software estudiados. El objetivo de este capítulo ha sido vincular al lector con el tema de la investigación y dar a conocer la importancia del desarrollo de un módulo de la PGST que cumpla con los objetivos requeridos.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Es importante antes de llevar a cabo el proceso de desarrollo de un software conocer todos los aspectos y particularidades del entorno donde se desempeñará (ya sea una empresa, una organización), para así tener una clara visión de la situación problemática y poder buscar soluciones factibles y precisas, además de lograr una comunicación efectiva entre los usuarios y el equipo de proyecto, con el objetivo de llegar a un entendimiento de lo que hay que hacer, ya que esa es precisamente la clave del éxito en la producción de un software. Precisamente debido a todos estos aspectos se hace necesario un modelado del negocio.

2.1 Objeto de Automatización

Tomándose en cuenta la situación problemática anterior se decidió implementar un módulo de la PGST que permita crear configuraciones de equipamiento tecnológico para los locales de la universidad en dependencia de las funciones que éstos desempeñan, así como tener un control del equipamiento tecnológico y sus configuraciones.

2.2 Flujo Actual del Proceso

El Departamento de Inversiones elabora una oferta de compra de equipamiento tecnológico para los locales de la universidad donde se especifican los requerimientos técnicos que tendrán los mismos. Esta oferta es entregada a la DGT, quien la aprueba o rechaza desde el punto de vista tecnológico, pues es quien define y asigna la configuración del equipamiento tecnológico. En caso de que se rechace esa oferta de compra se crea una nueva oferta y se muestra hasta tanto sea aprobada. Después de comprado el equipamiento tecnológico la DGT es la encargada de ubicarlo y a partir de ese momento controla cualquier cambio que ocurra con el mismo.

2.3 Modelo Historia de Usuario del Negocio

El modelo de Historia de Usuario del Negocio (HUN) es un artefacto que se genera del juego de la planificación, luego de estar definida la concepción del sistema se hace mucho más fácil comprender el negocio.

Se definen las características específicas del negocio, así como la forma en que interactúa el sistema con los clientes y viceversa.

Los objetivos del modelado del negocio son:

- Comprender la estructura y la dinámica de la organización en la cual se va a implantar un sistema.

- Comprender los problemas actuales de la organización e identificar las mejoras potenciales.
- Asegurar que los consumidores, usuarios finales y desarrolladores tengan un entendimiento común de la organización.
- Derivar los requerimientos del sistema que va a soportar la organización. (20)

2.4 Actor del Negocio

Un actor del negocio es cualquier individuo, grupo, organización, máquina o sistema de información externo que interactúa con el negocio. El término actor significa el rol que algo o alguien juega cuando interactúa con el negocio para beneficiarse de sus resultados. De acuerdo con esta idea un actor del negocio representa un tipo particular de usuario del negocio más que un usuario físico, ya que varios usuarios físicos pueden realizar el mismo papel en relación al negocio, pueden ser instancias de un mismo actor. (21)

A continuación se muestra el actor del negocio identificado y su descripción:

Actor	Descripción
Especialista de Inversiones	El Especialista de Inversiones es el que da comienzo a los procesos de definir y asignar la configuración de equipamiento tecnológico. Se beneficia de los resultados de dichos procesos, ya que a través de ellos se puede determinar el tipo de configuración de equipamiento tecnológico que se necesita comprar para los locales.

2.5 Trabajadores del Negocio

Un trabajador del negocio es una abstracción de una persona (o un grupo de personas), una máquina o un sistema automatizado, que actúa en el negocio realizando una o varias actividades, interactuando con otros trabajadores y manipulando entidades del negocio, representando éstas generalmente, a los objetos que se toman, inspeccionan, producen o utilizan durante la realización de los casos de uso del negocio. (22)

A continuación se muestran los trabajadores del negocio identificados y sus descripciones:

Trabajadores	Descripción
Especialista DGT	Es el encargado de definir y asignar un tipo de configuración de equipamiento tecnológico para los locales. Interactúa directamente con el negocio.
Jefe de área	Es el encargado de llevar un control de los cambios que se efectúan con el equipamiento tecnológico en los locales. Interactúa directamente con el negocio.

2.6 Diagrama de Historia de Usuario del Negocio

Un diagrama de HUN representa gráficamente a los procesos del negocio y su interacción con los actores del negocio. A continuación se muestra el diagrama de HUN:

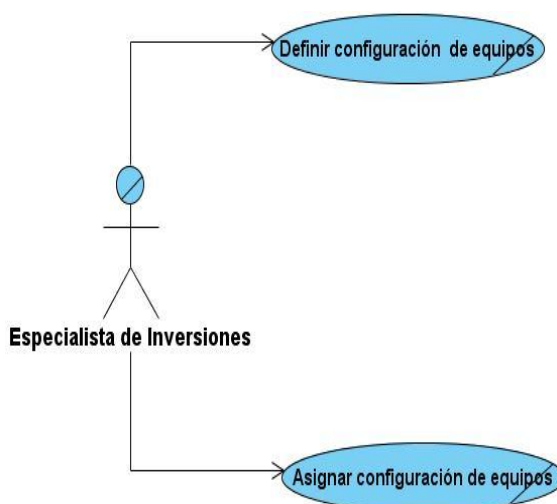


Figura4. Diagrama de HUN

2.7 Lista de Reserva del Producto

Es una lista priorizada que define el trabajo que se va a realizar en el proyecto. Cuando un proyecto comienza es muy difícil tener claro todos los requisitos sobre el producto. Sin embargo, suelen surgirlos más importantes que casi siempre son más que suficientes para un Sprint (Iteración).

Esta lista puede crecer y modificarse a medida que se obtiene más conocimiento acerca del producto y el cliente. Con la restricción de que solo puede cambiarse entre Sprint. El objetivo es asegurar que el producto definido al terminar la lista es el más correcto, útil y competitivo posible y para esto la lista debe acompañar los cambios en el entorno y el producto.

2.7.1 Funcionalidades del sistema	Prioridad
1. Autenticar usuario	Alta
2. Gestionar usuario	Alta
2.1 Insertar usuario	
2.2 Mostrar usuario	
2.3 Modificar usuario	
3. Gestionar configuración	Alta
3.1 Insertar configuración	
3.2 Modificar configuración	
3.3 Listar configuración	
4. Gestionar tipo local	Alta
4.1 Insertar local	
4.2 Mostrar local	
4.3 Modificar local	
4.4 Eliminar local	
5. Gestionar tipo equipo	Alta
5.1 Insertar tipo equipo	
5.2 Mostrar tipo equipo	
5.3 Modificar tipo equipo	
5.4 Eliminar tipo equipo	

- | | |
|--|-------|
| 6. Gestionar característica tipo equipo | Alta |
| 6.1 Insertar característica tipo equipo | |
| 6.2 Mostrar características tipo equipo | |
| 6.3 Modificar característica tipo equipo | |
| 6.4 Eliminar característica tipo equipo | |
| 7. Gestionar área | Media |
| 7.1 Insertar área | |
| 7.2 Mostrar área | |
| 7.3 Modificar área | |
| 7.4 Eliminar área | |
| 8. Gestionar edificio | Media |
| 8.1 Insertar edificio | |
| 8.2 Mostrar edificio | |
| 8.3 Modificar edificio | |
| 8.4 Eliminar edificio | |
| 9. Gestionar local | Media |
| 9.1 Insertar local | |
| 9.2 Mostrar local | |
| 9.3 Modificar local | |
| 9.4 Eliminar local | |
| 10. Gestionar control de configuración | Alta |
| 10.1 Insertar equipos | |
| 10.2 Modificar equipos | |
| 10.3 Eliminar equipos | |
| 10.4 Mostrar notificación | |
| 11. Gestionar características de equipos | Alta |
| 11.1 Insertar características de equipo | |
| 11.2 Mostrar características de equipo | |

11.3 Modificar características de equipo

11.4 Eliminar características de equipo

12. Generar informe de configuración

Baja

2.7.2 Requisitos no funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

1 Usabilidad

Cada usuario que se autentique solo tendrá acceso a la información que le corresponde.

El personal que trabaja con la aplicación debe conocer el funcionamiento del ordenador y tener conocimientos básicos sobre los entornos Web.

2 Portabilidad

El sistema será compatible con los sistemas operativos Windows 98 o superior y Linux.

3 Seguridad

El usuario para acceder al sistema pasará por la autenticación, lo cual consistirá en otorgarle un nombre de usuario único y una contraseña.

Luego del usuario autenticarse, el administrador le asignará los permisos de rol a cada usuario.

La información que se almacenará en el sistema está protegida de acceso y cambios no autorizados.

Se mantiene la integridad de todos los datos.

4 Apariencia o interfaz externa

Se diseñará una interfaz amigable y entendible para facilitar el manejo a los usuarios que trabajarán con ella.

Se tendrá una página principal desde la cual se realizarán todas las funcionalidades que una vez terminadas se retornará a ella.

No se recargarán las páginas con imágenes, gráficos y textos.

5 Software

El servidor local debe poseer un servidor Apache 2.2 con PostgreSQL8.1 y PHP 5.0.

Los clientes tendrán acceso a la aplicación a través del navegador Web Mozilla Firefox.

6 Hardware

Servidor

Microprocesador Intel Pentium III o superior, 800 MHz mínimo. Memoria RAM de 512 MB o superior. Capacidad del disco duro 80 GB o superior.

Cliente

Microprocesador Intel Pentium III o superior, 450 MHz mínimo. Memoria RAM de 128 MB mínimo.

2.8 Plan de Release (Entrega)

Una vez definidas las funcionalidades del módulo y agrupadas por historias de usuario se define el plan de release donde se muestran las iteraciones para realizar las entregas intermedias y final de cada funcionalidad del módulo. De acuerdo a la prioridad definida por funcionalidades e interés del cliente se realizó la siguiente planificación:

Release	Descripción de la iteración	Orden de la HU a implementar	Duración total
1	Implementar las funcionalidades autenticar y gestionar usuario.	1,2	1
2	Implementar las funcionalidades gestionar configuración, gestión tipo de local, gestión tipo de equipos	3,4,5,6	3

	y gestionar características		
3	Implementar las funcionalidades gestionar área, gestionar edificio y gestionar local	7,8,9	2
4	Implementar las funcionalidades gestionar control de configuración, gestionar características de equipos y generar informe de configuración	10,11,12	2

2.9 Actores del sistema

Los actores del sistema intercambian información con el sistema, pueden ser un recipiente pasivo de información, pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado. (23)

A continuación se definen los actores que interactúan con el sistema y sus descripciones:

Actores	Descripción
Especialista DGT	Es el encargado de gestionar las configuraciones de equipamiento tecnológico para los locales, gestionar los tipos de locales asociados a la configuración, gestionar los tipos de equipos así como sus características. Además de generar un informe de configuración Se beneficia de los resultados del sistema por lo que se considera un actor del sistema.

Administrador	Es en el encargado de gestionar los usuarios que trabajarán con el sistema y de asignarles permisos a esos usuarios.
Jefe de área	Es el encargado de gestionar las áreas, edificios y locales. Además de gestionar el control de configuraciones y gestionar las características de equipos.

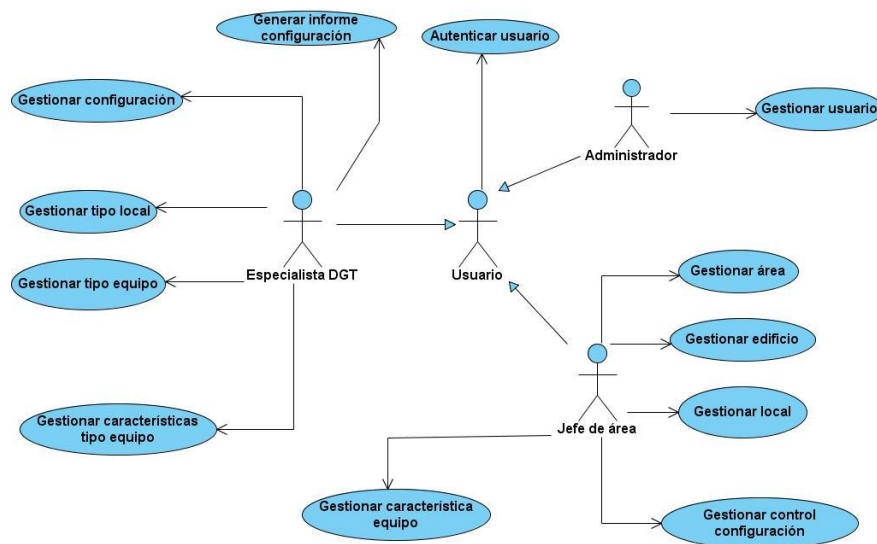


Figura5. Diagrama de HU

2.10 ¿Qué es un patrón?

Un patrón es una solución a un problema en un contexto, codifica conocimiento específico acumulado por la experiencia en un dominio. Un sistema bien estructurado está lleno de patrones. (24)

Patrón Actores múltiples: Rol común (Multiple Actors: Common Role). Dos actores interpretan el mismo papel en una determinada HU. Este rol es representado por otro actor, que contiene de forma hereditaria los actores que comparten este rol. Este patrón debe aplicarse cuando, desde el punto de vista de una HU hay una sola entidad externa interactuando con cada instancia de la HU.

Patrón CRUD (Creating, Reading, Updating and Deleting). Este patrón posee una HU llamada "Información CRUD" o "Gestionar Información", que modela todas las operaciones que se pueden

realizar sobre una parte de información de un tipo determinado; ya sea crearla, leerla, actualizarla y eliminarla. Este patrón se usa cuando todos los flujos contribuyen al mismo valor de negocio y son cortos y simples.

2.11 Historias de Usuarios

Las HU son la técnica utilizada en XP para especificar los requisitos del software, lo que equivale a los casos de uso en el proceso unificado. Las mismas son escritas por los clientes como las tareas que el sistema debe hacer y su construcción depende principalmente de la habilidad que tenga el cliente para definir las. Son escritas en lenguaje natural, no excediendo su tamaño de unas pocas líneas de texto. Las HU guían la construcción de las pruebas de aceptación, deben generarse una o más pruebas para verificar que la historia ha sido correctamente implementada y son utilizadas para estimar tiempos de desarrollo.

En esta plantilla los campos de puntos estimados y puntos reales se llenan, luego del desarrollo de la actividad de estimación de esfuerzo, donde se decide qué tiempo se le dedicará a cada HU definida con anterioridad. Las HU proporcionan ventajas pues:

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

2.12 HU: Autenticar usuario

2.12.1 Descripción de la HU Autenticar usuario

Historia de Usuario	
Número: 1	Nombre Historia de Usuario: Autenticar usuario
Modificación de Historia de Usuario Número: ninguna	
Usuario: Danays Rodriguez-Marilys Valiente	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Medio	Puntos Reales: 1
Descripción: El usuario introduce su nombre de usuario y contraseña del dominio para acceder al sistema. El sistema verifica que los datos sean correctos y otorga el acceso con	

los permisos correspondientes a dicho usuario.

Observaciones: El módulo tiene los roles de Administrador, Especialista y Técnico.

Prototipo de Interfaz:



2.13 HU: Gestionar usuario

2.13.1 Descripción de la HU Gestionar usuario

Historia de Usuario

Número:2 **Nombre Historia de Usuario:** Gestionar Usuario

Modificación de Historia de Usuario Número: ninguna

Usuario: Danays Rodriguez-Marilys Valiente **Iteración Asignada:** 1

Prioridad en Negocio: Alta **Puntos Estimados:**1

Riesgo en Desarrollo: Medio **Puntos Reales:** 1

Descripción: Permite al usuario adicionar o editar datos de un usuario. El administrador deberá introducir los atributos (nombre, roll) y el sistema se encargará de verificar los datos del usuario para realizar cada operación.

Observaciones: Los roles de los usuarios serán (Administrador, Especialista, Técnico)

Prototipo de Interfaz:



2.14 HU: Gestionar configuración

2.14.1 Descripción de la HU Gestionar configuración

Historia de Usuario

Número: 3	Nombre Historia de Usuario: Gestionar configuración
Modificación de Historia de Usuario Número: ninguna	
Usuario: Danays Rodriguez-Marilys Valiente	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 3
Riesgo en Desarrollo: Medio	Puntos Reales: 3
<p>Descripción: Permite al usuario gestionar configuraciones de equipamiento tecnológico. El sistema mostrará un Asistente de configuraciones donde se introducirá la función de la configuración, una descripción asociada a la misma y la fecha en que fue creada. El sistema verificará que la función de la configuración no sea duplicada. Cuando se crea la configuración se muestra un listado de las configuraciones y se permitirá actualizar los valores.</p>	
<p>Observaciones: Solo existirá una función por cada configuración que sea creada.</p>	
<p>Prototipo de Interfaz:</p>	

2.15 HU: Gestionar tipo local

2.15.1 Descripción de la HU Gestionar tipo local

Historia de Usuario	
Número: 4	Nombre Historia de Usuario: Gestionar tipo local
Modificación de Historia de Usuario Número: ninguna	
Usuario: Danays Rodriguez-Marilys Valiente	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 3

Riesgo en Desarrollo: Medio

Puntos Reales: 3

Descripción: Permite al usuario adicionar, editar o eliminar un tipo de local asociado a una configuración. El sistema provee un código de identificación por tipo de configuración y el usuario deberá introducir el tipo de local. En caso de que se desee editar se activarán los campos para introducir los datos y si se desea eliminar se mostrará un mensaje de confirmación para eliminar. El sistema verificará que el local no sea duplicado.

Observaciones:

Prototipo de Interfaz:



2.16 HU: Gestionar tipo equipo

2.16.1 Descripción de la HU Gestionar tipo equipo

Historia de Usuario

Número:5 **Nombre Historia de Usuario:** Gestionar tipo equipo

Modificación de Historia de Usuario Número: ninguna

Usuario: Danays Rodriguez-Marilys Valiente **Iteración Asignada:** 2

Prioridad en Negocio: Alta **Puntos Estimados:**3

Riesgo en Desarrollo: Medio

Puntos Reales: 3

Descripción: Permite al usuario adicionar, editar o eliminar los tipos de equipos para un tipo de local. El usuario deberá introducir los atributos (descripción, tipo) para adicionar el tipo de equipo y se oprimirá el botón “Siguiente” para finalizar la operación. En caso de eliminar un tipo de equipo el sistema mostrará un mensaje de confirmación para eliminar.

Observaciones: El sistema verificará que no queden campos vacíos.

Prototipo de Interfaz:



2.17 HU: Gestionar característica tipo equipo

2.17.1 Descripción de la HU Gestionar característica tipo equipo

Historia de Usuario

Número:6

Nombre Historia de Usuario: Gestionar características tipo equipo

Modificación de Historia de Usuario Número: ninguna

Usuario: Danays Rodriguez-Marilys Valiente

Iteración Asignada: 2

Prioridad en Negocio: Alta

Puntos Estimados:3

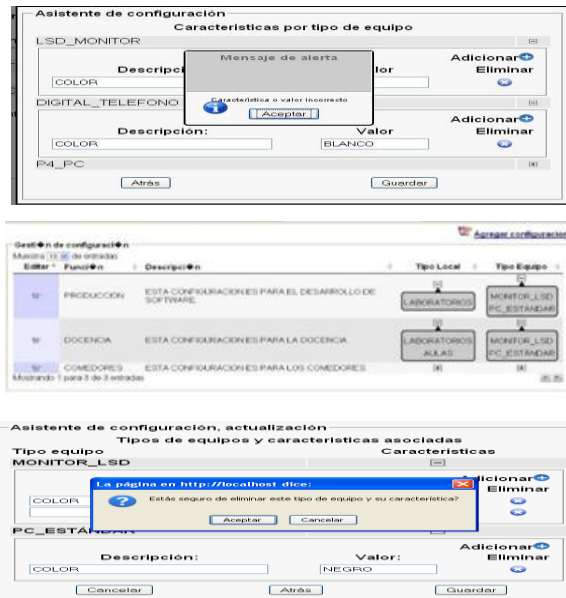
Riesgo en Desarrollo: Medio

Puntos Reales: 3

Descripción: Permite al usuario adicionar, editar o eliminar características de un tipo de equipo. El sistema muestra el tipo de equipo al que se le incorporarán sus características, se deberán introducir los atributos (descripción, valor) para incorporarlas. Se oprimirá el botón “Guardar” para almacenar los datos en el sistema. El sistema mostrará un mensaje en caso de que el valor de la característica sea incorrecto.

Observaciones: El sistema verificará que no queden campos vacíos.

Prototipo de Interfaz:



2.18 HU: Gestionar área

2.18.1 Descripción de la HU Gestionar área

Historia de Usuario

Número:7 **Nombre Historia de Usuario:** Gestionar área

Modificación de Historia de Usuario Número: ninguna

Usuario: Danays Rodriguez-Marilys Valiente **Iteración Asignada:** 3

Prioridad en Negocio: Media **Puntos Estimados:**2

Riesgo en Desarrollo: Medio **Puntos Reales:** 2

Descripción: Permite al usuario adicionar, editar o eliminar un área. El sistema permitirá adicionar el nombre del área. Se oprimirá el botón “Guardar” para almacenar los datos en el sistema. En caso que desee editar el área se pondrá el nombre de la nueva área y se oprimirá el botón “Aceptar”. Si se desea eliminar el sistema mostrará un mensaje de confirmación y se eliminarán los datos.

Observaciones:

Prototipo de Interfaz:



2.19 HU: Gestionar edificio

2.19.1 Descripción de la HU Gestionar edificio

Historia de Usuario

Número:8 **Nombre Historia de Usuario:** Gestionar edificio

Modificación de Historia de Usuario Número: ninguna

Usuario: Danays Rodriguez-Marilys Valiente **Iteración Asignada:** 3

Prioridad en Negocio: Media **Puntos Estimados:**2

Riesgo en Desarrollo: Medio **Puntos Reales:** 2

Descripción: Permite al usuario adicionar, editar o eliminar un edificio. El sistema permitirá adicionar los atributos (área, nombre). Se oprimirá el botón guardar para almacenar los datos en el sistema. En caso que se desee editar el edificio se pondrá el nuevo nombre del edificio y del área al que pertenece y se oprimirá el botón “Aceptar”. Si se desea eliminar el sistema mostrará un mensaje de confirmación y se eliminarán los datos.

Observaciones:

Prototipo de Interfaz:



2.20 HU: Gestionar local

2.20.1 Descripción de la HU Gestionar local

Historia de Usuario

Número:9 **Nombre Historia de Usuario:** Gestionar local

Modificación de Historia de Usuario Número: ninguna

Usuario: Danays Rodriguez-Marilys Valiente **Iteración Asignada:** 3

Prioridad en Negocio: Media **Puntos Estimados:**2

Riesgo en Desarrollo: Medio **Puntos Reales:** 2

Descripción: Permite al usuario adicionar, editar o eliminar un local. El sistema permitirá adicionar los atributos (edificio, nombre, tipo). Se oprimirá el botón “Guardar” para almacenar los datos en el sistema. En caso que se desee editar el local se pondrá los nuevos datos y se oprimirá el botón “Aceptar”. Si se desea eliminar el sistema mostrará un mensaje de confirmación y se eliminarán los datos.

Observaciones:

Prototipo de Interfaz:



2.21 HU: Gestionar control configuración

2.21.1 Descripción de la HU Gestionar control configuración

Historia de Usuario

Número:10 **Nombre Historia de Usuario:** Gestionar control configuración

Modificación de Historia de Usuario Número: ninguna

Usuario: Danays Rodriguez-Marilys Valiente **Iteración Asignada:** 3

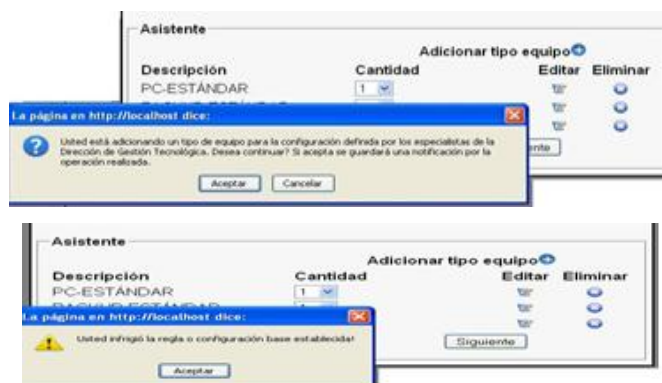
Prioridad en Negocio: Alta **Puntos Estimados:**3

Riesgo en Desarrollo: Medio **Puntos Reales:** 3

Descripción: El sistema mostrará un explorador para el control de configuraciones. Se desplegará un listado de configuraciones de equipos por locales. Se mostrará un Asistente donde se visualizarán las configuraciones de equipos asociadas a un local. En caso de que se desee adicionar, editar o eliminar equipos a una configuración se realizarán dichas operaciones. El sistema realizará un estudio de las condiciones o hipótesis existentes para esa configuración y notificará una conclusión en caso de que no se cumpla con las reglas implementadas para dicha configuración.

Observaciones: Que se mantengan las configuraciones reglamentadas.

Prototipo de Interfaz:



2.22 HU: Gestionar características equipos

2.22.1 Descripción de la HU Gestionar características equipos

Historia de Usuario

Número:11 **Nombre Historia de Usuario:** Gestionar características equipos

Modificación de Historia de Usuario Número: ninguna

Usuario: Danays Rodriguez-Marilys Valiente **Iteración Asignada:** 3

Prioridad en Negocio: Alta **Puntos Estimados:**3

Riesgo en Desarrollo: Medio	Puntos Reales: 3
<p>Descripción: El sistema mostrará un explorador para el control de configuraciones. Se desplegará un listado de configuraciones de equipos por locales. Se mostrará un Asistente donde se visualizarán las configuraciones de equipos y el usuario podrá seleccionar un equipo para introducir sus características. En caso que se desee editar la característica de un equipo se activarán los campos (descripción, valor) se actualizarán los datos y se oprimirá el botón "Aceptar". Si se desea eliminar el sistema mostrará un mensaje de confirmación y se eliminarán los datos.</p>	
<p>Observaciones: Que se mantengan las configuraciones reglamentadas.</p>	
<p>Prototipo de Interfaz:</p>	

2.23 HU: Generar informe de configuración

2.23.1 Descripción de la HU Generar informe de configuración

Historia de Usuario	
Número: 12	Nombre Historia de Usuario: Generar informe de configuración
Modificación de Historia de Usuario Número: ninguna	
Usuario: Danays Rodriguez-Marilys Valiente	Iteración Asignada: 4
Prioridad en Negocio: Baja	Puntos Estimados: 2
Riesgo en Desarrollo: Baja	Puntos Reales: 2
<p>Descripción: El sistema muestra un informe general de las configuraciones de equipamiento tecnológico. El usuario selecciona la función y el sistema despliega la configuración de la misma dada en (tipos de locales, tipos equipos, características).</p>	
<p>Observaciones: Que estén creadas las configuraciones de equipamiento tecnológico.</p>	

Prototipo de Interfaz:



Conclusiones

En este capítulo se describió el objeto de automatización y el flujo actual del proceso. Se identificaron y describieron los actores y trabajadores del negocio, se obtuvo el diagrama de Historia de Usuario del Negocio, la lista de reserva del producto, así como los requisitos no funcionales y se describieron las historias de usuario del sistema para contribuir a la futura implementación del mismo.

CAPÍTULO 3: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Tras la definición y descripción en el capítulo anterior de las funcionalidades deseadas y necesarias del sistema propuesto se hace necesario definir cómo se desarrollará.

En este capítulo quedan definido el diseño, arquitectura del sistema y la información que se desea persista a través del diseño de la BD. Se modelan los diagramas de componentes y despliegue y se pasa a la implementación del sistema propuesto. Se elaborarán además los casos de prueba de aceptación a partir de las HU descritas anteriormente.

3.1 Patrones de diseño

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces, introduciendo el trabajo con patrones de diseño para asignar responsabilidades.

SXP propone un diseño simple (Diagrama de paquetes) para una solución que pueda funcionar y ser implementada, en la cual un diseño de software correcto es aquel que supera todas las pruebas, no tiene lógica duplicada, pone de manifiesto las intenciones importantes de los programadores y tiene el mínimo número de clases y métodos.

Los patrones GRASP (General, Responsibility, Assignment Software, Patterns) incluyen al Experto, Creador, Alta Cohesión, Bajo Acoplamiento y Controlador los cuales se verán reflejados y se describen a continuación:

Experto: Consiste en asignar la responsabilidad al experto en información, la clase que cuenta con la información necesaria para cumplir dicha responsabilidad. A continuación se muestra la clase gestión_configuración, la cual tiene una relación de dependencia con el resto de las clases pues cuenta con un grupo de responsabilidades que se necesitan para interactuar con las demás clases, siendo la clase experta en el dominio de la información.


```

gestion_configuracion
+gestion_configuracion()
+index()
+index_II()
+index_III()
+listar_configuraciones()
+listar_tipo_locales()
+id_tipo_local_dado_desc()
+id_tipo_equipo_dado_desc()
+listar_diccionario()
+listar_diccionario_tipo_equipo()
+listar_diccionario_caracteristica()
+listar_diccionario_caracteristica_valor()
+listar_diccionario_tipo_tecnologia()
+listar_diccionario_local()
+listar_diccionario_update()
+tipos_equipos_por_configuracion()
+tipos_locales_por_configuracion()
+listar_caracteristicas()
+listar_caracteristicas_dado_id_tipo_equipo()
+listar_caracteristicas_distintas()
+listar_caracteristicas_distintas_dado_id_tipo_equipo()
+insertar_datos_configuracion()
+updatear_datos_configuracion()
+existe_tipo_local()
+existe_funcion()
+existe_tipo_equipo()
+datos_dado_configuracion()
+tipos_locales_aso_configuracion()
+tipo_equipo_caracteristica_aso_configuracion()
+tipo_equipo_tipo_aso_configuracion()
+desc_funcion_dado_id()
+desc_tipo_local_dado_id()
+add_tipo_local()
+id_tipo_local_dado_desc_update()
+verifica_id_tipo_local_id_funcion()
+caracteristica_x_tipo_equipo_tipo()

```

Alta cohesión: Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. A continuación se muestran las clases: gestión equipo y gestión configuración, estas dos clases tienen una estrecha relación con el objetivo de no sobrecargar de responsabilidades a una sola clase, pues no se gestiona una configuración de equipos sin gestionar los equipos.

gestion_equipo	gestion_configuracion
+gestion_equipo()	+gestion_configuracion()
+index()	+index()
+listar_tipos_equipos()	+index_II()
+mostrar_areas()	+index_III()
+mostrar_edificios()	+listar_configuraciones()
+mostrar_locales()	+listar_tipo_locales()
+mostrar_tipo_equipo_modal()	+id_tipo_local_dado_desc()
+mostrar_tipo_equipo()	+id_tipo_equipo_dado_desc()
+mostrar_equipo()	+listar_diccionario()
+listar_areas()	+listar_diccionario_tipo_equipo()
+add_area()	+listar_diccionario_caracteristica()
+datos_area_dado_id()	+listar_diccionario_caracteristica_valor()
+datos_edificio_dado_id()	+listar_diccionario_tipo_tecnologia()
+datos_local_dado_id()	+listar_diccionario_local()
+add_edificio()	+listar_diccionario_update()
+add_local()	+tipos_equipos_por_configuracion()
+editar_area()	+tipos_locales_por_configuracion()
+eliminar_area()	+listar_caracteristicas()
+eliminar_edificio()	+listar_caracteristicas_dado_id_tipo_equipo()
+eliminar_local()	+listar_caracteristicas_distintas()
+eliminar_equipo()	+listar_caracteristicas_distintas_dado_id_tipo_equipo()
	+insertar_datos_configuracion()
	+updatear_datos_configuracion()
	+existe_tipo_local()
	+existe_funcion()
	+existe_tipo_equipo()
	+datos_dado_configuracion()

Bajo acoplamiento: El acoplamiento es la medida de qué tan fuertemente está conectada una clase con las demás (es decir, cuántas clases conoce y necesita). Una clase con bajo acoplamiento no depende de muchas otras clases. A continuación se muestran las clases: gestión usuario y gestión

configuración, las cuales a pesar de que se encuentran relacionadas, tienen una débil dependencia, por lo que un cambio en las responsabilidades de una de ellas no afectaría a la otra.

gestion_usuario	gestion_configuracion
+gestion_usuario() +index() +listar_usuario() +listar_roles() +listar_roles_update() +add_usuario() +existe_usuario() +devuelve_user() +desc_user() +autenticar()	+gestion_configuracion() +index() +index_II() +index_III() +listar_configuraciones() +listar_tipo_locales() +id_tipo_local_dado_desc() +id_tipo_equipo_dado_desc() +listar_diccionario() +listar_diccionario_tipo_equipo() +listar_diccionario_caracteristica() +listar_diccionario_caracteristica_valor() +listar_diccionario_tipo_tecnologia() +listar_diccionario_local() +listar_diccionario_update() +tipos_equipos_por_configuracion() +tipos_locales_por_configuracion() +listar_caracteristicas() +listar_caracteristicas_dado_id_tipo_equipo() +listar_caracteristicas_distintas() +listar_caracteristicas_distintas_dado_id_tipo_equipo() +insertar_datos_configuracion() +updatear_datos_configuracion() +existe_tipo_local() +existe_funcion() +existe_tipo_equipo() +datos_dado_configuracion()

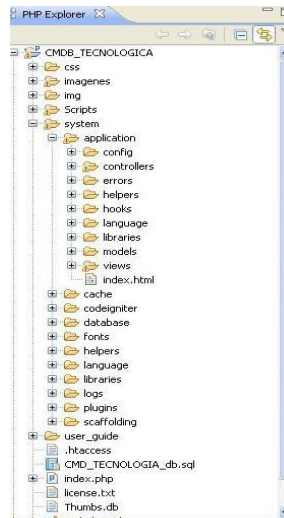
Controlador: Delega la responsabilidad de controlar el flujo de eventos del sistema en varias clases con las que tiene una alta cohesión. Por lo que no existe una sola clase controladora que realice todas las actividades garantizando de esta forma mejorar las validaciones y seguridad en el sistema. A continuación se muestran las clase controladoras: gestión usuario, gestión equipo y gestión configuración, cada una de ellas tiene las responsabilidades específicas en el dominio de la aplicación.

gestion_usuario	gestion_equipo	gestion_configuracion
+gestion_usuario() +index() +listar_usuario() +listar_roles() +listar_roles_update() +add_usuario() +existe_usuario() +devuelve_user() +desc_user() +autenticar()	+gestion_equipo() +index() +listar_tipos_equipos() +mostrar_areas() +mostrar_edificios() +mostrar_locales() +mostrar_tipo_equipo_modal() +mostrar_tipo_equipo() +mostrar_equipo() +listar_areas() +add_area() +datos_area_dado_id() +datos_edificio_dado_id() +datos_local_dado_id() +add_edificio() +add_local() +editar_area() +eliminar_area() +eliminar_edificio() +eliminar_local() +eliminar_equipo()	+gestion_configuracion() +index() +index_II() +index_III() +listar_configuraciones() +listar_tipo_locales() +id_tipo_local_dado_desc() +id_tipo_equipo_dado_desc() +listar_diccionario() +listar_diccionario_tipo_equipo() +listar_diccionario_caracteristica() +listar_diccionario_caracteristica_valor() +listar_diccionario_tipo_tecnologia() +listar_diccionario_local() +listar_diccionario_update() +tipos_equipos_por_configuracion() +tipos_locales_por_configuracion() +listar_caracteristicas() +listar_caracteristicas_dado_id_tipo_equipo() +listar_caracteristicas_distintas() +listar_caracteristicas_distintas_dado_id_tipo_equipo() +insertar_datos_configuracion() +updatear_datos_configuracion() +existe_tipo_local() +existe_funcion() +existe_tipo_equipo() +datos_dado_configuracion()

3.2 Descripción de la arquitectura

La arquitectura del sistema está basada en el patrón MVC que usa el framework CodeIgniter, permitiendo que las páginas web contengan mínima codificación ya que la presentación es separada

del código PHP. CodeIgniter también permite incorporar sus códigos existentes, o incluso desarrollar librerías de núcleo para el sistema, habilitándolo a trabajar en forma que tenga más sentido. A continuación se muestra la estructura del sistema.



3.2.1 Modelo Vista Controlador

MVC es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos:

- Modelo: representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de éstos y permite derivar nuevos datos.
- Vista: presenta el modelo en un formato adecuado para interactuar con el usuario.
- Controlador: responde a eventos o acciones del usuario e invoca cambios en el modelo y la vista.

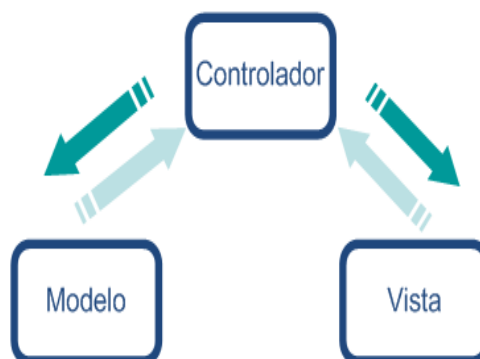


Figura 7. Patrón arquitectónico MVC

3.3 Diseño con metáforas

El modelo de diseño es un artefacto que se genera del diseño con las metáforas, donde se debe diseñar la solución más simple que pueda funcionar y ser implementada. En SXP no se enfatiza la definición temprana de una arquitectura estable para el sistema, dicha arquitectura se asume de forma evolutiva y los posibles inconvenientes que se generarían por no contar con ella en el comienzo del proyecto se solventan con la existencia de una metáfora.

El sistema es definido mediante una metáfora o un conjunto de metáforas compartidas entre el cliente y el equipo de desarrollo. Una metáfora es una historia compartida que describe cómo debería funcionar el sistema. (25)

Teniendo en cuenta las características antes expuesta, se define en esta plantilla un esbozo inicial del diseño del sistema, sin entrar en especificaciones, ni detalles, solo lo que el diseñador necesita para hacer un primer entregable del sistema. Esta plantilla proporciona ventajas como:

- Permite confeccionar un diseño inicial y sencillo del sistema.
- Es la base para la definición de una futura arquitectura.

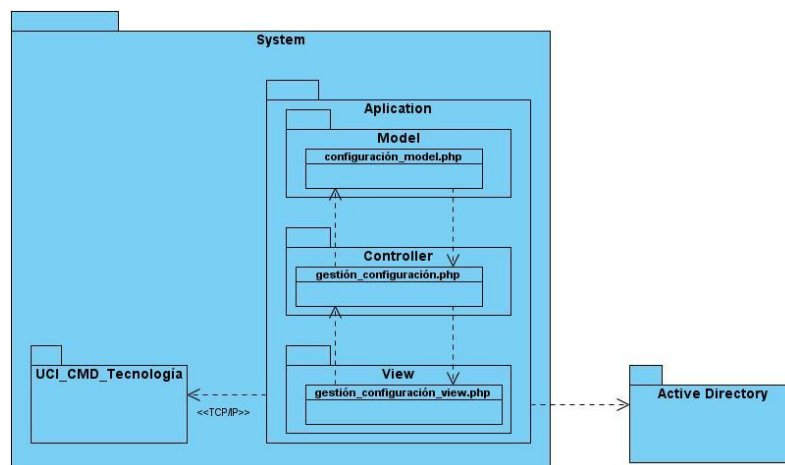


Figura8. Diagrama de paquetes

3.4 Diagrama de Clases Persistentes

La persistencia es la capacidad que tiene un objeto de perdurar en tiempo y espacio. A continuación se muestra el Diagrama de Clases Persistentes:

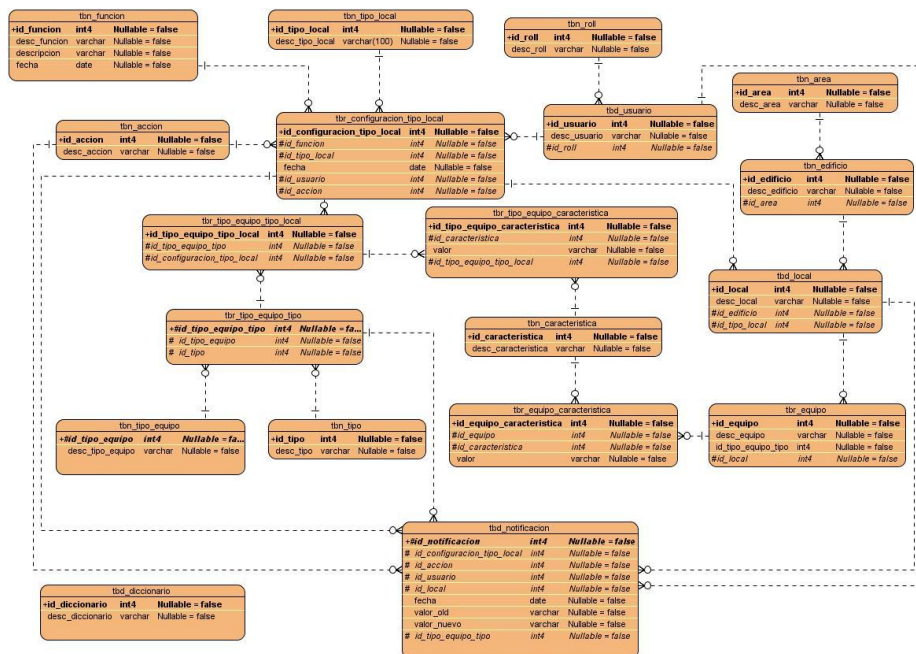


Figura10. Modelo de datos

Descripción de las principales tablas:

Tabla usuario

Nombre: usuario		
Descripción: En esta tabla se recoge toda la información referente a los usuarios que trabajarán con la aplicación.		
Atributo	Tipo	Descripción
Id_usuario	Integer	Este atributo es un identificador que representa el usuario.
desc_usuario	Varchar	En este atributo se especifica el nombre de dominio del usuario.
Id_rol	Integer	En este atributo se especifica el tipo de permisos que tendrá el usuario en la aplicación.

Tabla roll

Nombre: roll		
Descripción: En esta tabla se recoge toda la información referente a los permisos de acceso		

que tendrán los usuarios.

Atributo	Tipo	Descripción
Id_roll	Integer	Este atributo es un identificador que representa el roll que tendrá el usuario.
desc_roll	Varchar	En este atributo se especifica el tipo de roll que tendrá en usuario.

Tabla función

Nombre: función		
Descripción: En esta tabla se recoge toda la información referente a la función que realiza cada local de la universidad.		
Atributo	Tipo	Descripción
Id_función	Integer	Este atributo es un identificador que representa un código identificador de la función del local.
descripción_función	Varchar	En este atributo se especifica el nombre de la función del local.
descripción	Varchar	En este atributo se especifica una breve descripción de la función.

Tabla equipo

Nombre: equipo		
Descripción: En esta tabla se recoge toda la información referente a los tipos de equipos existentes en la universidad.		
Atributo	Tipo	Descripción
Id_equipo	Integer	Este atributo es un identificador que representa el equipo.
desc_equipo	Varchar	Este atributo representa el nombre del equipo.
Id_tipo_equipo_tipo	Integer	Este atributo es un identificador que representa el tipo de equipo al que

		pertenece el equipo.
Id_local	Integer	Este atributo representa el local al que pertenece el equipo.

Tabla tipo

Nombre: tipo		
Descripción: En esta tabla se recoge la información referente a los equipos que a su vez son de un tipo específico.		
Atributo	Tipo	Descripción
Id_tipo	Integer	Este atributo es un identificador que representa los tipos asociados a equipos.
desc_tipo	Varchar	En este atributo se especifica el nombre que se le da a ese tipo.

Tabla característica

Nombre: característica		
Descripción: Esta tabla se muestra los atributos de las características que poseen los tipos de equipos.		
Atributo	Tipo	Descripción
Id_característica	Integer	Este atributo es identificador de la característica que tiene el equipo.
desc_característica	Varchar	En este atributo se especifica el valor que posee la característica.

Tabla tipo_local

Nombre: tipo_local		
Descripción: En esta tabla se recoge toda la información referente a los tipos de locales de la universidad.		
Atributo	Tipo	Descripción
Id_local	Integer	Este atributo es un identificador que

		representa el local.
desc_local	Varchar	Este atributo describe el nombre del local.
Id_edificio	Integer	Este atributo es un identificador que representa el nombre del edificio.
Id_tipo_local	Integer	En este atributo es un identificador que representa el tipo de local al que pertenece el local.

Tabla local

Nombre: local		
Descripción: En esta tabla se recoge toda la información referente a los todos los locales de la universidad.		
Atributo	Tipo	Descripción
Id_local	Integer	Este atributo es un identificador que representa los locales de la universidad.
descripción_local	Varchar	En este atributo se especifica el nombre de los locales.
Id_edificio	Integer	Este atributo es identificador de que representa la edificación en que se encuentra ese local.

Tabla edificio

Nombre: edificio		
Descripción: En esta tabla se recoge toda la información referente a los edificios existentes en la universidad.		
Atributo	Tipo	Descripción
Id_edificio	Integer	Este atributo es un identificador que representa las edificaciones de la universidad.
Id_área	Integer	Este atributo es un identificador que

		representa las áreas existentes en la universidad.
desc_edificio	Varchar	En este atributo se describen los nombres de las edificaciones de la universidad.

3.6 Diagrama de componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos del software que conforman un sistema informático, que pueden ser: archivos, paquetes, bibliotecas cargadas dinámicamente, etc.

El sistema está descrito a través del siguiente diagrama de componentes, donde se agrupan los elementos físicos del módulo en paquetes evidenciando de esta forma el uso del patrón de arquitectura MVC y el framework CodeIgniter.

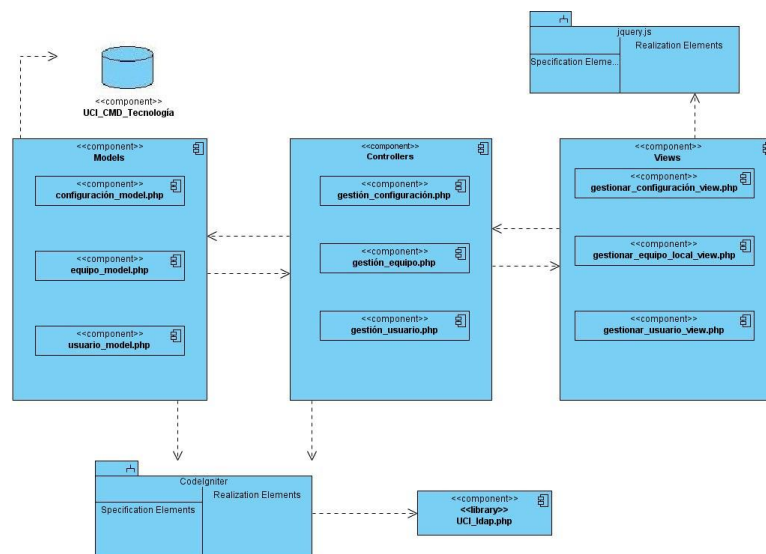


Figura 11. Diagrama de componentes

3.7 Diagrama de despliegue

La vista de despliegue define la arquitectura física del sistema por medio de nodos interconectados. Estos son elementos de hardware sobre los cuales pueden ejecutarse los elementos de software.

El diagrama de despliegue para esta aplicación consta de un Servidor Web que es donde estará instalado el sistema, éste se conecta mediante los protocolos TCP/IP al servidor PostgreSQL (Servidor BD) donde estará guardada toda la información de las solicitudes y reportes. Las estaciones de trabajo (PC Cliente) se conectan al Servidor Web también por TCP/IP que es donde el usuario interactúa con el sistema. Existirá además un Servidor de BD respaldo para hacer copias a la información.

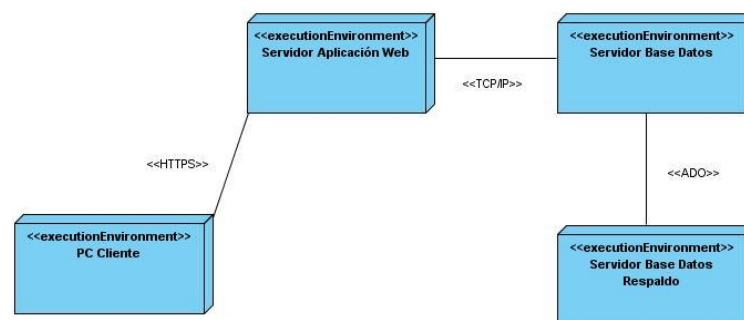


Figura 12. Diagrama de despliegue

El sistema estará distribuido de la siguiente manera: En el nodo central de la UCI se encontrarán la aplicación servidora, a la que se conectarán las PC clientes de forma directa con autorización de la dirección, el servidor de BD donde se encontrarán todos los datos del sistema y el servidor de copia de respaldo de la información manejada.

3.7.1 Descripción de nodos

Servidor de Aplicaciones Web

Requerimientos de hardware: El Servidor Web debe tener alta disponibilidad y un rendimiento adecuado, se propone: Microprocesador Intel Pentium III o superior, 800 MHz mínimo. Memoria RAM 512 MB o superior. Capacidad del disco duro: 80 GB o superior.

Servidor de BD

Requerimientos de hardware: El servidor de almacenamiento de datos debe tener buena capacidad que responda a un volumen de información grande, se propone: Microprocesador Intel Pentium III o superior, 800 MHz mínimo. Memoria RAM 512 MB o superior. Capacidad del disco duro 80 GB o superior.

Servidor de BD para copia de respaldo

Requerimientos de hardware: La información a manejar es importante para la DGT y para la universidad, por lo que debe garantizarse. Debe tener las mismas características que el servidor primario de BD, se propone: Microprocesador Intel Pentium III o superior, 800 MHz mínimo. Memoria RAM 512 MB o superior. Capacidad del disco duro: 80 GB o superior.

PC Cliente

Requerimientos de hardware: Se hace referencia a cualquier PC cliente desde la que se pueda acceder vía Web a la aplicación. Los requerimientos de hardware necesarios son mínimos debido a que el grueso del procesamiento es llevado a cabo en el servidor, se propone: Microprocesador Intel Pentium III o superior, 450 MHz mínimo. Memoria RAM 128 MB mínimo.

3.7.2 Descripción de elementos e interfaces de comunicación

<<HTTPS>>

Protocolo de comunicación que se describe como HTTP (protocolo estándar básico de la arquitectura Web) con la inclusión de SSL lo que permite que la información que se mueve a través de los diferentes canales de la red sean encriptados. Se ha utilizado para la comunicación establecida entre el Servidor de Aplicaciones y la PC cliente.

<<TCP/IP>>

El TCP/IP es la base de Internet, y sirve para enlazar computadoras que utilizan diferentes sistemas operativos, incluyendo PC, minicomputadoras y computadoras centrales sobre redes de área local (LAN) y área extensa (WAN).

3.8 Pantallas

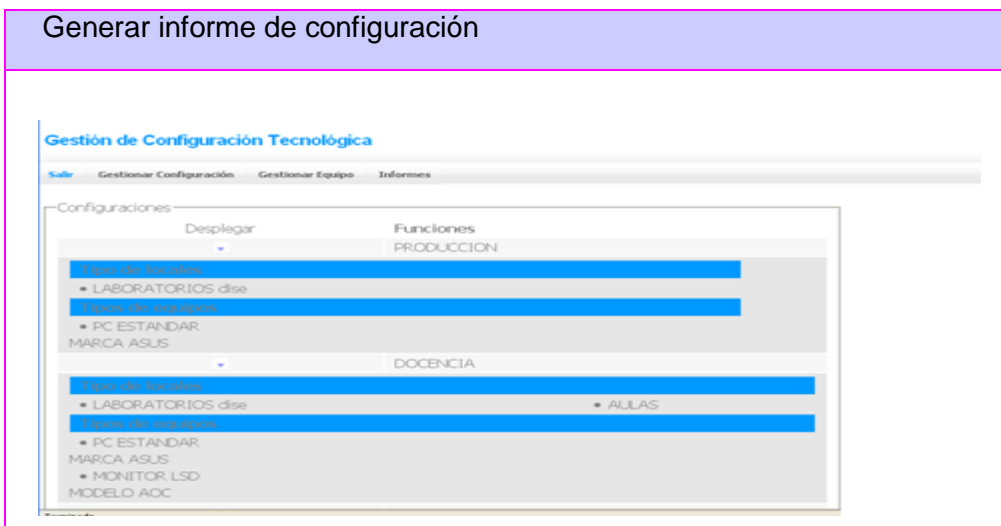
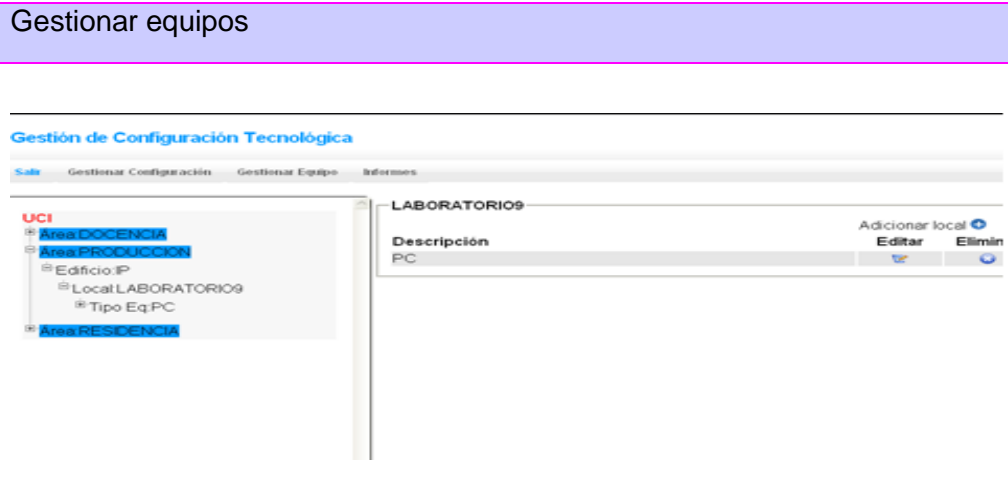
Autenticar usuario



Gestionar configuración



Editar	Fecha Creación	Función	Descripción	Tipo Local	Tipo Equipo
	2010-05-12	PRODUCCION	PARA PRODUCIR SOFTWARE		
	2010-05-12	DOCENCIA	ESTA CONFIGURACION ES PARA IMPARTIR CLASES.		
	2010-05-15	RESIDENCIA	ESTA CONFIGURACION ES PARA LOS APTOS.		



3.9 Casos de prueba

La metodología de desarrollo de software SXP define un conjunto de pruebas de funcionalidad entre cada sprint para así poder pasar a un sprint superior. Durante el desarrollo de esta aplicación se realizaron un conjunto de pruebas a las que fue sometido el módulo para validar sus funcionalidades.

Caso de Prueba de Aceptación HU: Autenticar Usuario

Caso de Prueba de Aceptación	
Código Caso de Prueba: Historia Usuario 1	Nombre Historia de Usuario: Autenticar Usuario

Nombre de la persona que realiza la prueba: Danays Rodriguez-Marilys Valiente
Descripción de la Prueba: Para entrar al sistema, en la página principal el usuario encontrará un formulario para introducir su usuario de dominio y contraseña. Si el usuario no está registrado en la aplicación no tendrá acceso a ningún servicio del sistema. Una vez introducidos los campos en el formulario el usuario tendrá acceso a los servicios del sistema de acuerdo a los permisos otorgados por el administrador (Administrador, Especialista, Técnico). Se entrarán datos incorrectos para validar que el sistema no permite la entrada de usuarios no autorizados. Se entrarán datos correctos para validar que el sistema autoriza la entrada de los usuarios que el administrador haya autorizado para trabajar con el sistema.
Condiciones de Ejecución: Que exista conexión a la red para que el sistema pueda verificar el usuario del dominio.
Entrada/Pasos de ejecución: En la ventana principal del sistema se selecciona el vínculo administración y el sistema solicita el usuario y contraseña del dominio. El usuario introduce los campos (usuario y contraseña) y oprime el botón “Aceptar”.
Resultado Esperado: Que el usuario sea autenticado.
Evaluación de la Prueba: Satisfactoria.

Caso de Prueba de Aceptación HU: Gestionar Usuario

Caso de Prueba de Aceptación	
Código Caso de Prueba: Historia Usuario 2	Nombre Historia de Usuario: Gestionar Usuario
Nombre de la persona que realiza la prueba: Danays Rodriguez-Marilys Valiente	
Descripción de la Prueba: El administrador después de haber introducido su usuario y contraseña de dominio podrá acceder al módulo de administración, el cual le permite adicionar o editar datos de un usuario. Para adicionar un usuario en el sistema, el administrador oprimirá el botón “Agregar usuario” y se activarán los campos (usuario, roll) para introducir los datos del usuario. Se oprime el botón “Aceptar” para guardar los datos. Se introducirá un usuario que exista en el sistema para verificar que el sistema valida la entrada de usuarios.	

Para editar los datos de un usuario, el administrador oprimirá el botón de “Editar” y se desplegarán los campos (usuario, roll). Se oprimir el botón “Aceptar” para guardar las actualizaciones.
Condiciones de Ejecución: Que el usuario esté autenticado en el sistema.
Entrada/Pasos de ejecución: En la ventana principal del sistema se selecciona el vínculo administración, el administrador deberá introducir su usuario y contraseña de dominio para acceder al módulo de administración.
Resultado Esperado: Que se incorporen nuevos usuarios al sistema.
Evaluación de la Prueba: Satisfactoria.

Caso de Prueba de Aceptación HU: Gestionar configuración

Caso de Prueba de Aceptación	
Código Caso de Prueba: Historia Usuario 3	Nombre Historia de Usuario: Gestionar configuración
Nombre de la persona que realiza la prueba: Danays Rodriguez-Marilys Valiente	
Descripción de la Prueba: El sistema muestra una ventana de Asistente de Configuración. El usuario introduce la función y una breve descripción asociada a la misma. Se entrará una función duplicada para validar que el sistema verifica que las funciones sean únicas. En este caso el sistema mostrará un mensaje “Esa función es duplicada”. Se entrará una función correcta para verificar que el sistema cree una configuración. Se oprime el botón “Siguiente”.	
Condiciones de Ejecución: Que el usuario esté autenticado en el sistema y tenga privilegios de administración.	
Entrada/Pasos de ejecución: En la ventana principal del sistema se selecciona el vínculo gestión de configuración y se introduce la función y una breve descripción de la misma. El sistema asocia un código de identificación para esa configuración y se oprime el botón “Siguiente”.	
Resultado Esperado: Que se cree una configuración de equipamiento tecnológico.	
Evaluación de la Prueba: Satisfactoria.	

Caso de Prueba de Aceptación HU: Gestionar tipo local

Caso de Prueba de Aceptación	
Código Caso de Prueba: Historia Usuario 4	Nombre Historia de Usuario: Gestionar tipo local
Nombre de la persona que realiza la prueba: Danays Rodriguez-Marilys Valiente	
Descripción de la Prueba: Después de creada la configuración el usuario pasa a introducir los tipos de locales asociados a la misma. Para adicionar un tipo de local el usuario deberá oprimir el botón “Adicionar” y se activarán el campo (tipo de local) para introducir los datos. Se pasarán a guardar los datos del tipo de local. Se introducirán datos incorrectos para verificar que el sistema valide que el nombre del tipo de local asociado a la configuración no es correcto. Se mostrará un mensaje: “Tipo de local incorrecto”. Se introducirán datos correctos y se oprimirá el botón “Siguiente” para verificar que el sistema crea el tipo de local. Para editar los datos de un tipo de local, el usuario deberá oprimir el botón de “Editar” y se desplegará un listado con el campo (tipo de local). Se pasará a modificar los datos del tipo de local y después el sistema guardará las actualizaciones. Para eliminar un tipo de local el usuario deberá oprimir el botón “Eliminar”, el sistema subrayará el tipo de local y el usuario oprimirá el botón “Siguiente”, se mostrará un mensaje de confirmación para eliminar.	
Condiciones de Ejecución: Que el usuario esté autenticado en el sistema y tenga privilegios de administración.	
Entrada/Pasos de ejecución: Después de haber creado una configuración de equipamiento tecnológico se muestra un Asistente de Configuración y se pasa a incorporar los tipos de locales asociados a la configuración.	
Resultado Esperado: Que se creen los tipos locales para una configuración.	
Evaluación de la Prueba: Satisfactoria.	

Caso de Prueba de Aceptación HU: Gestionar tipo equipo

Caso de Prueba de Aceptación	
Código Caso de Prueba: Historia Usuario 5	Nombre Historia de Usuario: Gestionar tipo equipo
Nombre de la persona que realiza la prueba: Danays Rodriguez-Marilys Valiente	
<p>Descripción de la Prueba: El usuario después de haber introducido el tipo de local asociado a la configuración pasará a introducir los tipos de equipos asociados a la misma. Para adicionar un tipo de equipo el usuario deberá oprimir el botón “Adicionar” y se activarán los campos (tipo equipo, tipo) para introducir los datos del tipo de equipo. Se pasarán a guardar los datos del tipo de equipo.</p> <p>Se introducirá un tipo de equipo repetido para verificar que el sistema valida la entrada de tipos de equipos existentes.</p> <p>Se introducirá un tipo de equipo que no esté repetido para verificar que el sistema incorpora los tipos de equipos.</p> <p>Para editar los datos de un tipo de equipo, el usuario deberá oprimir el botón de “Editar” y se desplegará un listado con los campos (tipo equipo, tipo) y se pasará a modificar los datos del tipo de equipo, el sistema guardará las actualizaciones.</p> <p>Se introducirá un tipo de equipo incorrecto para verificar que el sistema verifica que los tipos de equipos sean correctos.</p> <p>Para eliminar un tipo de equipo el usuario deberá oprimir el botón “Eliminar”, se muestra un mensaje de conformación y se eliminan el tipo de equipo.</p>	
Condiciones de Ejecución: Que usuario esté autenticado en el sistema y que tenga privilegios de administración.	
Entrada/Pasos de ejecución: Después de haber introducido el tipo de local en el Asistente de Configuración, el sistema pasar a incorporar los tipos de equipos de ese tipo de local.	
Resultado Esperado: Que se creen los tipos de equipos asociados a los tipos de locales.	
Evaluación de la Prueba: Satisfactoria.	

Caso de Prueba de Aceptación HU: Gestionar característica tipo equipo

Caso de Prueba de Aceptación	
Código Caso de Prueba: Historia Usuario 6	Nombre Historia de Usuario: Gestionar característica tipo equipo
Nombre de la persona que realiza la prueba: Danays Rodriguez-Marilys Valiente	
<p>Descripción de la Prueba: El usuario después de haber introducido el tipo de equipo para los locales pasará a introducir las características de esos tipos de equipos.</p> <p>Para adicionar una característica de un tipo de equipo el usuario deberá oprimir el botón “Adicionar” y se activarán los campos (descripción, valor) para introducir los datos de la característica. Se pasará a oprimir el botón “Guardar” para almacenar los datos.</p> <p>Se introducirá una característica que exista en el sistema para verificar que el sistema valida las características de equipos existentes.</p> <p>Para editar los datos de una característica el usuario deberá oprimir el botón de “Editar” y se desplegará un listado con todos campos (descripción, valor), se pasará a modificar los datos de la característica y el sistema guardará las actualizaciones.</p> <p>Para eliminar la característica el usuario deberá oprimir el botón “Eliminar” y se eliminan todos los datos de la característica del tipo de equipo.</p>	
Condiciones de Ejecución Que el usuario esté autenticado en el sistema y que tenga privilegios de administración.	
Entrada/Pasos de ejecución: Después de haber introducido el tipo de equipo en el Asistente de Configuración, el sistema pasa a incorporar las características de esos tipos de equipos.	
Resultado Esperado: Que se introduzcan las características asociadas a los tipos de equipos.	
Evaluación de la Prueba: Satisfactoria.	

Caso de Prueba de Aceptación HU: Gestionar área

Caso de Prueba de Aceptación	
Código Caso de Prueba: Historia Usuario 7	Nombre Historia de Usuario: Gestionar área

Nombre de la persona que realiza la prueba: Danays Rodriguez-Marilys-Valiente
Descripción de la Prueba: Después de que el sistema muestra un explorador para el control de configuraciones se oprime el botón “Adicionar área”. Para adicionar un área se mostrará un Asistente y el usuario introducirá el campo (nombre). Se oprimirá el botón “Guardar” para almacenar los datos. Para editar los datos de un área se oprimirá el botón “Editar” y se activarán los campos para modificar el área. Se oprimirá el botón “Aceptar” para guardar las actualizaciones. Para eliminar los datos de un área se oprimirá el botón “Eliminar”, el sistema mostrará un mensaje de confirmación para eliminar.
Condiciones de Ejecución: Que el usuario esté autenticado en el sistema.
Entrada / Pasos de ejecución: Después que el sistema muestra el explorador para el control de configuraciones, se muestra un Asistente para incorporar los datos de un área en el sistema.
Resultado Esperado: Que se incorporen áreas al sistema.
Evaluación de la Prueba: Satisfactoria.

Caso de Prueba de Aceptación HU: Gestionar edificio

Caso de Prueba de Aceptación	
Código Caso de Prueba: Historia Usuario 8	Nombre Historia de Usuario: Gestionar edificio
Nombre de la persona que realiza la prueba: Danays Rodriguez-Marilys-Valiente	
Descripción de la Prueba: Después de que el sistema muestra un explorador para el control de configuraciones se oprime el botón “Adicionar edificio”. Para adicionar un edificio se mostrará un Asistente y el usuario introducirá los campos (área, nombre). Se oprimirá el botón “Guardar” para almacenar los datos. Para editar los datos de un edificio se oprimirá el botón “Editar” y se activarán los campos para modificar el edificio. Se oprimirá el botón “Aceptar” para guardar las actualizaciones. Para eliminar los datos de un edificio se oprimirá el botón “Eliminar”, el sistema mostrará un mensaje de confirmación para eliminar.	
Condiciones de Ejecución: Que el usuario esté autenticado en el sistema.	
Entrada / Pasos de ejecución: Después que el sistema muestra el explorador para el control de configuraciones y se haya introducido el área, se muestra un Asistente para	

incorporar los datos de un edificio en el sistema.
Resultado Esperado: Que se incorporen edificios a las áreas.
Evaluación de la Prueba: Satisfactoria.

Caso de Prueba de Aceptación HU: Gestionar local

Caso de Prueba de Aceptación	
Código Caso de Prueba: Historia Usuario 9	Nombre Historia de Usuario: Gestionar local
Nombre de la persona que realiza la prueba: Danays Rodriguez-Marilys-Valiente	
Descripción de la Prueba: Después de que el sistema muestra un explorador para el control de configuraciones se oprime el botón “Adicionar local”. Para adicionar un local se mostrará un Asistente y el usuario introducirá los campos (edificio, nombre, tipo). Se oprimirá el botón “Guardar” para almacenar los datos. Para editar los datos de un local se oprimirá el botón “Editar” y se activarán los campos para modificar el local. Se oprimirá el botón “Aceptar” para guardar las actualizaciones. Para eliminar los datos de un local se oprimirá el botón “Eliminar”, el sistema mostrará un mensaje de confirmación para eliminar.	
Condiciones de Ejecución: Que el usuario esté autenticado en el sistema.	
Entrada / Pasos de ejecución: Después que el sistema muestra el explorador para el control de configuraciones y después de haber incorporados los datos del área y del edificio, se muestra un Asistente para incorporar los datos de un local en el sistema.	
Resultado Esperado: Que se incorporen locales a los edificios.	
Evaluación de la Prueba: Satisfactoria.	

Caso de Prueba de Aceptación HU: Gestionar control de configuración

Caso de Prueba de Aceptación	
Código Caso de Prueba: Historia Usuario10	Nombre Historia de Usuario: Gestionar control configuración
Nombre de la persona que realiza la prueba: Danays Rodriguez-Marilys-Valiente	
Descripción de la Prueba: Después de que el sistema muestra un explorador para el control de configuraciones se muestra un Asistente con las configuraciones por locales con los campos (descripción, cantidad).	

<p>Para adicionar equipos al control de configuraciones se oprimirá el botón “Adicionar tipo equipo”.</p> <p>Para editar los datos del control se oprimirá el botón “Editar” y se activarán los campos para modificar datos asociados. Se oprimirá el botón “Aceptar” para guardar las actualizaciones.</p> <p>Para eliminar los datos de un local se oprimirá el botón “Eliminar”, el sistema mostrará un mensaje de confirmación para eliminar.</p> <p>En cada una de las operaciones realizadas el sistema mostrará un mensaje de notificación alertando al usuario que no debe alterar las configuraciones según las reglas implementadas.</p>
<p>Condiciones de Ejecución: Que el usuario esté autenticado en el sistema.</p>
<p>Entrada / Pasos de ejecución: Después que el sistema muestra el explorador para el control de configuraciones se mostrará un Asistente que permitirá adicionar, editar o eliminar equipos de una configuración.</p>
<p>Resultado Esperado: Que se cumpla con el control de configuraciones.</p>
<p>Evaluación de la Prueba: Satisfactoria.</p>

Caso de Prueba de Aceptación HU: Gestionar características de equipos

Caso de Prueba de Aceptación	
<p>Código Caso de Prueba: Historia Usuario11</p>	<p>Nombre Historia de Usuario: Gestionar características de equipos</p>
<p>Nombre de la persona que realiza la prueba: Danays Rodriguez-Marilys-Valiente</p>	
<p>Descripción de la Prueba: Después de que el sistema muestra un explorador para el control de configuraciones se muestra un Asistente con las configuraciones por locales con los campos (descripción, cantidad). El usuario introduce los equipos y las cantidades. Se desplegará un Asistente para introducir las características de dichos equipos, se activarán los campos (descripción, valor)</p> <p>Para adicionar características de equipos se oprimirá el botón “Adicionar característica”.</p> <p>Para editar los datos de la característica de un equipo se oprimirá el botón “Editar” y se activarán los campos para modificar los datos asociados. Se oprimirá el botón “Aceptar” para guardar las actualizaciones.</p> <p>Para eliminar los datos de la característica de un equipo se oprimirá el botón “Eliminar”, el sistema mostrará un mensaje de confirmación para eliminar.</p> <p>En cada una de las operaciones realizadas el sistema mostrará un mensaje de notificación</p>	

alertando al usuario que no debe alterar las configuraciones según las reglas implementadas anteriormente.
Condiciones de Ejecución: Que el usuario esté autenticado en el sistema.
Entrada / Pasos de ejecución: Después que el sistema muestra el explorador para el control de configuraciones y después de introducido los equipos y sus cantidades se mostrará un Asistente que permitirá adicionar, editar o eliminar las características de cada uno de los equipos.
Resultado Esperado: Que se cumpla con el control de configuraciones.
Evaluación de la Prueba: Satisfactoria.

Caso de Prueba de Aceptación HU: Generar informe configuración

Caso de Prueba de Aceptación	
Código Caso de Prueba: Historia Usuario12	Nombre Historia de Usuario: Generar informe configuración
Nombre de la persona que realiza la prueba: Danays Rodriguez-Marilys-Valiente	
Descripción de la Prueba: Después de que el usuario gestiona las configuraciones de equipamiento tecnológico seleccionar el vínculo "Generar informe de configuración" y se muestra un listado con las funciones de las configuraciones creadas. El usuario seleccionará la función a visualizar la configuración y se desplegarán los tipos de locales, tipos de equipos y sus características.	
Condiciones de Ejecución: Que estén creadas las configuraciones de equipamiento tecnológico.	
Entrada / Pasos de ejecución: Después que el usuario selecciona el vínculo "Gestionar informe de configuración" se muestra un listado con los tipos de locales, tipos de equipos y sus características.	
Resultado Esperado: Que se muestre un informe con las configuraciones de equipamiento tecnológico.	
Evaluación de la Prueba: Satisfactoria	

Conclusiones

En este capítulo se han cumplido con todos los artefactos que se generan en la fase de Desarrollo de la metodología SXP como (diagrama de paquetes, diagrama de clases persistentes, modelo de datos,

diagrama de componentes y de despliegue) además de haberle realizado al sistema las pruebas de funcionalidad necesarias para validar el cumplimiento de la lista de reserva del producto, siendo este resultado satisfactorio. A partir de este momento el módulo para la gestión de configuración de equipamiento tecnológico en la UCI queda implementado.

CONCLUSIONES

- Se realizó un análisis de los principales aspectos y conceptos relacionados con el proceso de gestión de configuración en la universidad, lográndose la comprensión del entorno de negocio que sirvió de base para el desarrollo del sistema.
- Se creó un software que permite recoger toda la información referente al equipamiento tecnológico que se maneja por parte de la universidad.
- El software garantizará el control y la organización del equipamiento tecnológico en la universidad.

RECOMENDACIONES

- Agregar al módulo la implementación de la funcionalidad eliminar configuración, en caso de que lo requiera el cliente.
- Agregar al módulo la implementación de otras funcionalidades que garanticen el proceso contable que lleve un control del costo y vida útil del equipamiento tecnológico existente en la universidad.
- Asegurar sobre el sistema el cumplimiento de un proceso de mantenimiento, soporte y actualización periódica para garantizar la fiabilidad, seguridad y agilidad del sistema.
- Integrar el módulo implementado a la Plataforma de Gestión de Servicios Tecnológicos.

REFERENCIA BIBLIOGRÁFICAS

1. Divulgación y difusión científica. [En línea] [Citado el: 2 de 10 de 2009.] Disponible en: http://www.cienciateca.com/divulgacion_hugodiaz.html.
2. Fundamentos de la Gestión de TI. [En línea] [Citado el: 2 de 10 de 2009.] Disponible en: http://itil.osiatis.es/Curso_ITIL/Gestion_Servicios_TI/fundamentos_de_la_gestion_TI/que_es_ITI_L/que_es_ITIL.php.
3. Osiatis. [En línea] [Citado el: 4 de 10 de 2009.] Disponible en: http://itil.osiatis.es/Curso_ITIL/Gestion_Servicios_TI/gestion_de_configuraciones/introduccion_objetivos_gestion_de_configuraciones/introduccion_objetivos_gestion_de_configuraciones.php.
4. Osiatis. [En línea] [Citado el: 4 de 10 de 2009.] Disponible en: http://itil.osiatis.es/Curso_ITIL/Gestion_Servicios_TI/gestion_de_configuraciones/introduccion_objetivos_gestion_de_configuraciones/introduccion_objetivos_gestion_de_configuraciones.php.
5. Aranda Software Corporation. [En línea] [Citado el: 9 de 10 de 2009.] Disponible en: http://www.arandasoft.com/solucion_aam.php.
6. Axios Systems. [En línea] [Citado el: 9 de 10 de 2009.] Disponible en: <http://www.axiossystems.com>.
7. Sourceforge.net Software. [En línea] [Citado el: 3 de 5 de 2010.] Disponible en: <http://sourceforge.net/search>.
8. [En línea] [Citado el: 3 de 5 de 2010.] Disponible en: <http://sourceforge.net/search>.
9. Directorio de Servicios Web.UCI. [En línea] [Citado el: 6 de 11 de 2009.] Disponible en: http://uddi.uci.cu/lineamientos_de_arquitectura_DIN.
10. **Kniberg, Henrik.** *Scrum y XP desde las trincheras*. 2008.
11. **Booch, Grady, Rumbaugh, James y Jacobson, Ivar.** *Unified Modeling Language 2000*.
12. **Kendall, Kendall.** *Análisis y diseño de sistemas*.
13. **López Marrero, Yaneisy y Almira Morfe, Liliana.** *Aplicación Web para la confección del horario docente en la Facultad 7*. Ciudad Habana : s.n., 2008.
14. Codeigniter_Spanish_UserGuide. [En línea] [Citado el: 11 de 12 de 2009.] Disponible en: <http://www.scribd.com/doc/3965892/CodeIgniter-Spanish-UserGuide>.
15. Maestros de la Web. [En línea] [Citado el: 16 de 12 de 2009.] Disponible en: <http://www.maestrosdelweb.com/editorial/zendstudio/>.
16. Manual de PHP. Achour, Friedhelm Betz, Antony Dovgal, Nuno Lopez. [En línea] 2004. [Citado el: 22 de 12 de 2009.] Disponible en: <http://www.phpwelt.net/handbuecher/spanish/manual.htm>.

17. Linux para todos. [En línea] [Citado el: 5 de 5 de 2010.] Disponible en: <http://www.linuxparatodos.net/portal/staticpages/index.php?page=servidor-web>.
18. Buenas tareas. [En línea] [Citado el: 5 de 5 de 2010.] Disponible en: <http://www.buenastareas.com/ensayos/Bases-De-Datos/314011.html>.
19. buayacorp. [En línea] [Citado el: 25 de 11 de 2009.] Disponible en: <http://www.buayacorp.com/archivos/postgresql-82>.
20. **Romero, Peñalver, Gladys M.** *MA-GMPR-UR2. Metodología ágil para proyectos de software libre en la facultad 10 de la Universidad de las Ciencias Informáticas.* Ciudad Habana : s.n., 2008.
21. *Clases de Ingeniería del Software I/teleformacion.uci.cu/mod/resource/view.php? Clase#2 Modelamiento del Negocio.* Ciudad Habana : UCI, 2008-2009.
22. *Clases de Ingeniería de Software I/teleformacion.uci.cu/mod/resource/view.php?id=6680. Clase#2 Modelamiento del Negocio.* . Ciudad Habana : UCI, 2008-2009.
23. *Conferencia 6/Fase de Inicio/ Disciplina de Requisitos/eva.uci.cu/course/view.php?id=102.* Ciudad Habana : UCI, 2008-2009.
24. *Patrones de arquitectura/internos.uci.cu/Teleclases.asp?id_as=12.* Ciudad Habana : UCI, 2008-2009.
25. **Cruz Roja, Orlando y Puig, Claudio.** *Sistema basado en casos para la asistencia al proceso de soporte y ayuda en la gestión de incidencias tecnológicas en la Universidad de las Ciencias Informáticas.* Ciudad Habana : UCI, 2009.

BIBLIOGRAFÍA

1. Metodologías de Desarrollo de Software. [En línea] [Citado el: 12 de 3 de 2009.] Disponible en: http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html
2. Metodologías ágiles para el desarrollo de software SCRUM + XP . [En línea][Citado el: 12 de 3 de 2009.] Disponible en: http://unkasoft.googlepages.com/desarrollo_agil_UPSA.pdf.
3. **F, Z, JORGE.** Metodologías Ágiles, 2007. [En línea] [Citado el: 16 de 3 de 2009.] Disponible en: <http://libresoft.es/downloads/ferrer-20030312.pdf>.
4. **Villar, Rodríguez, Malay.** *Introducción de procedimientos ágiles en la producción de software en la Facultad 7 de la Universidad de las Ciencias Informáticas.* Disponible en: http://bibliodoc.uci.cu/TD/TD_0693_07.pdf. 2007.
5. **Romero, Peñalver, Gladys M.** *MA-GMPR-UR2 Metodología ágil para proyectos de software libre en la facultad 10 de la Universidad de las Ciencias Informáticas.* Ciudad Habana : s.n., 2008.
6. Dirección de Informatización. *Arquitectura para los Sistemas que Conforman la intranet universitaria.* [En línea] [Citado el: 7 de 10 de 2009.] Disponible en: <http://uddi.uci.cu/files/arquitectura.2007.5.9.pdf>.
7. Principios de Diseño. [En línea] [Citado el: 2 de 2 de 2009.] Disponible en: <http://htm.www.sidar./recur/desde/traduc/es/visitable/quees/dcu.htm>.
8. Desarrolloweb. [En línea] [Citado el: 22 de 4 de 2009.] Disponible en: <http://www.desarrolloweb.com/articulos/2392.php>.
9. Patrón Modelo-Vista-Controlador. [En línea] [Citado el: 5 de 2 de 2009.] Disponible en: http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ_3317.as.
10. Entorno Virtual de Aprendizaje(EVA) . [En línea] [Citado el: 5 de 10 de 2009.] Disponible en: <http://eva.uci.cu/mod/resource/view.php?id=3729>.
11. Tutorial de UML. [En línea] [Citado el: 23 de 11 de 2009.] Disponible en: <http://www.dcc.uchile.cl/~psalinas/uml/introduccion.html>.
12. Tutorial de PostgreSQL. [En línea] [Citado el: 8 de 3 de 2010.] Disponible en: <http://es.tldp.org/Postgresql-es/web/navegable/tutorial/tutorial.html>.
13. **Matos, Jorge L.** *Sistema de Gestión de Tecnología de la Información de la Universidad de las Ciencias Informáticas: Diseño e implementación del módulo Solicitud de Servicios.* Disponible en: http://bibliodoc.uci.cu/TD/TD_1734_08.pdf. 2008.
14. El CoDiGoK. [En línea] [Citado el: 21 de 4 de 2010.] Disponible en: <http://elcodigok.blogspot.com/2008/03/diagramas-de-despliegue.html>.

15. CodeIgniter. Manual disponible en Inglés, traducido al español. [En línea] [Citado el: 6 de 12 de 2009.] Disponible en: <http://www.codeigniter.com>.

ANEXOS

GUIÓN DE LA METODOLOGÍA

Planificación ↔ Definición

- Entrevista con el cliente (concepción inicial)
 - **Plantilla de concepción del sistema.**
- Juego de la planificación.
 - Plantilla Modelo de Historia de Usuario del negocio.
- Captura de requisitos:
 - Creación de la LRP.
 - Plantilla LRP (Lista de Reserva del Producto).
 - Priorización de la LRP.
 - Definir las historias de usuario.
 - Plantilla Historia de usuario.
 - Asignar las historias de usuario.
 - Plantilla Historia de usuario. (actualizar gerente.)
- Valoración del esfuerzo.
 - Plantilla Historia de usuario. (actualizar gerente.)
- Valoración de riesgos.
 - Plantilla Lista de riesgos.
- Diseño con las metáforas.
 - Plantilla Modelo de diseño.
- Refactorización.
- Reunión de revisión del diseño.

Desarrollo

- Junta de planificación.
 - Plantilla de Glosario de términos.
- Definir las historias de usuario a implementar.
- Tareas para lograr dicha implementación.
- Plantilla de Tareas de Ingeniería.
 - Plantilla Cronograma de producción.

- Plantilla de Plan de release.
- Implementación.
 - Estándar de código.
 - Código fuente.
- Junta de seguimiento
- Taller técnico.
- Junta de revisión.
- Pruebas
 - Plan de Pruebas.
 - Plantilla Caso de Prueba de aceptación.

Entrega

- Entrega de la documentación.
- Entrenamiento.
 - Capacitación
 - Manual de usuario. (Programadores)
 - Manual de Identidad (Diseñadores)
 - Manual de desarrollo. (Desarrollador, y se genera automáticamente.)
 - Instalación.
- Marketing

Mantenimiento

- Soporte.
 - Plantilla de Gestión de cambios.

Observación: Desde el diseño en la fase de Planificación ↔ Definición y toda la fase de Desarrollo es un ciclo.

GLOSARIO DE TÉRMINOS

A:

Actor: Alguien o algo, fuera del sistema o negocio que interactúa con el sistema o negocio.

Artefactos: Son los resultados tangibles del proyecto producido, modificado o usado durante el proceso de desarrollo de software hasta obtener el producto final.

Aplicación Web: Sistema informático que los usuario utilizan accediendo a un Servidor Web a través de Internet o una Intranet.

C:

Configuración: Es el proceso de adaptar un elemento de hardware o de software al resto de los elementos de su entorno y a las necesidades específicas de los usuarios.

CMDB: Base de Datos de Administración de la Configuración.

E:

Elemento de Configuración: Es un componente de la Infraestructura de TI que es necesario para brindar un servicio (hardware, software, documentación).

Entorno de Desarrollo: Que incluyen a todos los programas necesarios para realizar todas las fases de puesta a punto de un programa.

G:

Gestión de configuración: Proceso mediante el cual se obtiene un control sobre el inventario, los elementos que lo componen y las relaciones funcionales entre ellos.

I:

ITIL: (Information Technologies Infrastructure Library) Biblioteca de Infraestructura de Tecnología de Información.

Infraestructura Tecnológica: Es la forma en que se distribuyen todos los elementos de configuración (hardware) según la estructura organizacional de la entidad.

M:

Metáfora: Se conoce como metáfora a la utilización de una expresión con un sentido diferente y en un contexto distinto al habitual.

P:

Procesos de negocio: Es un conjunto de tareas relacionadas de forma lógica, llevadas a cabo para lograr un resultado de negocio definido. Cada proceso de negocio tiene entradas, salidas y funciones.

R:

Release: Entrega.

Repositorio: Es un término utilizado en el dominio de las herramientas CASE. El repositorio podría definirse como la base de datos fundamental para el diseño; no sólo guarda datos, sino también algoritmos de diseño y, en general, elementos software necesarios para el trabajo de programación.

S:

Sprint: Un Sprint es el procedimiento de adaptación de las cambiantes variables del entorno (requisitos, tiempo, recursos, conocimiento, tecnología). Son ciclos iterativos en los cuales se desarrolla o mejora una funcionalidad para producir nuevos incrementos. Durante un Sprint el producto es diseñado, codificado y probado. Y su arquitectura y diseño evolucionan durante el desarrollo.

T:

Tecnología de Información (TI): Se entiende como las herramientas y métodos empleados para recabar, retener, manipular o distribuir información. La tecnología de la información se encuentra generalmente asociada con las computadoras y las tecnologías afines aplicadas a la toma de decisiones.

