



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

Título: Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas.

Autor: Guillermo Armada González

Tutores: Ing. Yamila Mateu Romero

Ing. Laritza González Marrero

Cotutor: MSc. Graciela González Pérez

La Habana, Junio 2019

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas



“... toda aquella gente que venga con buena voluntad, olvidar los rencores, los rencores olvidables, y seguir avanzando.”

Israel Rojas

(Dúo Buena Fe)

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

Declaración de autoría

Declaro por este medio que yo: Guillermo Armada González, con carné de identidad 95090528346, soy el autor principal del trabajo final de tesis de pregrado que se titula: “Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas”. El cual ha sido desarrollado como parte del trabajo en el centro DIN de la Vicerrectoría de Tecnología. Autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Y para que así conste, se firma la presente declaración jurada de autoría en La Habana, a los _____ días del mes de _____ del año 2019.

Autor

Guillermo Armada González

Tutor

Ing. Yamila Mateu Romero

Tutor

Ing. Laritza González Marrero

Tutor

MSc. Graciela González Pérez

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

RESUMEN

La presente investigación recoge un estudio que incluye el desarrollo e implementación de un módulo, integrado al Sistema de Gestión Universitaria de la Universidad de las Ciencias Informáticas, para el registro de medios tecnológicos personales. Parte de la necesidad de controlar el ecosistema de medios que circula y consume servicios de red en la Universidad. Para su acometido consta de tres capítulos donde se recogen toda la documentación generada durante el desarrollo de la investigación. En un primer capítulo de fundamentación teórica, se presenta el análisis de sistemas de inventario de dispositivos de hardware y sistema de gestión de solicitudes, incidiendo positivamente en la claridad de las funcionalidades deseadas que debe tener un módulo de este tipo. Posteriormente en el capítulo dos de análisis y diseño, se definen las herramientas y el entorno de desarrollo. Además se documenta el diseño del módulo, presentando todos los artefactos generados del uso de la metodología Scrum y algunos elementos de la metodología XP. En un tercer capítulo de implementación y validación, se definen los estándares de codificación empleados. Además quedan plasmados los resultados de las pruebas realizadas a la propuesta solución, destacando que cumple con los requerimientos funcionales exigidos por el cliente. Finalmente, aplicando el criterio de expertos, se constata que el resultado de esta investigación satisface la necesidad actual del proceso de registro de medios tecnológicos personales.

Palabras clave: registro, medios tecnológicos, módulo, desarrollo, resultados.

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

Índice de contenidos

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	7
1.1 <i>Inventario de activos tecnológicos</i>	7
1.1.1 Sistemas de inventario tecnológico a nivel internacional.....	8
1.1.2 Sistemas de inventario tecnológico a nivel nacional.....	10
1.2 <i>Entorno de desarrollo</i>	12
1.2.1 Lenguaje de modelado.....	12
1.2.2 Herramienta de modelado.....	13
1.2.3 Herramienta para el control de versiones	13
1.2.4 Lenguajes de programación.....	13
1.2.5 Biblioteca.....	14
1.2.6 Marco de trabajo.....	14
1.2.7 Formato de intercambio de datos.....	15
1.2.8 Servidor de aplicaciones web.....	15
1.2.9 Gestor de bases de datos	16
1.2.10 Entorno de Desarrollo Integrado.....	16
1.2.11 Metodología de desarrollo de software.....	17
1.3 <i>Conclusiones del capítulo.....</i>	21
CAPÍTULO 2: ANÁLISIS Y DISEÑO	22
2.1 <i>Objeto de automatización.....</i>	22
2.1.1 Descripción del proceso	22
2.2 <i>Descripción de la propuesta solución.....</i>	24
2.3 <i>Planeación.....</i>	25
2.3.1 Requisitos Funcionales	25
2.3.2 Requisitos no Funcionales	27
2.3.3 Definición de roles	28
2.3.4 Historias de Usuario	29
2.3.5 Estimación del esfuerzo por grupo de Historias de Usuario	35
2.4 <i>Plan de publicaciones.....</i>	36
2.4.1 Plan de iteraciones.....	36
2.4.2 Plan de duración de iteraciones	37
2.4.3 Plan de Entregas	38
2.5 <i>Patrones</i>	38
2.5.1 Patrón arquitectónico.....	39
2.5.2 Patrones de diseño.....	40

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

2.5.3 Patrones GoF	44
2.6 Modelo de datos	47
2.7 Conclusiones del capítulo.....	49
CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN DE LA PROPUESTA SOLUCIÓN.....	50
3.1 Estándares de codificación.....	50
3.1.1 Indentación, llaves de apertura y cierre, y tamaño de líneas	50
3.1.2 Convención de nomenclatura	51
3.1.3 Estructura de control	53
3.1.4 Documentación.....	53
3.1.5 Espacios en blanco	53
3.1.6 Buenas prácticas	53
3.2 Validación de requisitos.....	54
3.2.1 Técnicas de validación de requisitos.....	55
3.2.2 Resultados de aplicar las técnicas de revisión:.....	55
3.3 Estrategia de pruebas	55
3.3.1 Pruebas de unidad	56
3.3.2 Pruebas de integración.....	59
3.3.3 Pruebas de validación	62
3.3.4 Pruebas de sistema	66
3.4 Validación de hipótesis.....	68
3.5 Conclusiones del capítulo.....	70
CONCLUSIONES GENERALES	71
RECOMENDACIONES	72
BIBLIOGRAFÍA.....	73

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

INTRODUCCIÓN

El desarrollo de nuevas tecnologías y el abaratamiento de sus costos han facilitado el acceso de un mayor número de personas a las plataformas de hardware. Con una inclinación a la adquisición de medios tecnológicos portátiles como laptops y celulares, muy por encima a la preferencia por computadoras de escritorio (ver anexo 1). Esto ha propiciado que sea común contar con al menos uno de estos medios como parte de la vida cotidiana. A la vez, ha sido cómplice en el surgimiento de la tendencia Bring Your Own Device (BYOD), cuya traducción sería “trae tu propio dispositivo”. Esta tendencia permite al personal vinculado a un centro educativo y/o productivo, llevar y utilizar sus propios medios tecnológicos para acceder a los recursos de su entorno. Actualmente se observa la generalización de este conjunto en las diferentes esferas de la sociedad, como lo es en apoyo a la docencia. [1][2]

Cuba, a pesar de ser un país subdesarrollado, implementa un programa de educación para todos, que busca vincular las tecnologías con el ámbito educativo. Uno de sus principales exponentes es la Universidad de las Ciencias Informáticas (UCI), entidad comprometida con la informatización del país mediante el desarrollo de soluciones propias. La UCI cuenta con una plantilla de más de ocho mil personas entre estudiantes y trabajadores. Ante la complejidad que sobrepone administrar una institución de tal envergadura, ha optado por la informatización de sus procesos. Entre sus soluciones destaca una herramienta de apoyo personalizada denominada Sistema de Gestión Universitaria, que constituye un sistema integrador de los procesos sustantivos de la Universidad. [3]

En la UCI existe un ecosistema de medios tecnológicos en manos de los usuarios. La circulación de estos medios es restringida, por lo que debe ser controlada por la Dirección de Seguridad y Protección, y la Dirección de Seguridad Informática. La Dirección de Seguridad y Protección controla el acceso y salida de personal y activos del campus. La Dirección de Seguridad Informática norma las políticas y conductas referentes al uso de los medios tecnológicos, que deben seguir los estudiantes, trabajadores y el personal que visita la entidad, atendiendo al consumo de los diferentes servicios de red que se brindan. Según el Reglamento de Seguridad Informática, un medio tecnológico no registrado, no está autorizado a consumir servicios de red en la Universidad.

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

Los medios tecnológicos cuentan como activos básicos de la Universidad y pueden ser objeto de hurto, pérdida y delitos de seguridad informática. Para enfrentar esta situación se aplica un procedimiento básico de registro. Cada propietario debe solicitar el registro y aprobación de su medio tecnológico auxiliándose de una planilla impresa, lo cual es poco productivo, ineficaz y engorroso, dada la magnitud de usuarios que disponen de estos medios.

Cada solicitud es archivada, en ella se registran atributos del medio y datos del propietario. Del medio se registra el tipo de medio, marca, modelo, dirección física cableada, dirección física por conexión inalámbrica y número de serie. Del propietario se registra nombre completo, carné de identidad, número de solapín, área a la que pertenece dentro de la Universidad, dirección particular, teléfono, rol universitario y firma. Además se registra nombre, cargo, firma y cuño de los directivos de las diferentes áreas por la que debe circular la solicitud para ser aprobada.

La planilla no permite que el proceso sea flexible a la incorporación de un nuevo tipo de medio tecnológico o a un cambio en la estructura de aprobación sin la necesidad de diseño e impresión de un nuevo modelo. Al ser completada de forma manual, es propensa a errores en la introducción de los datos. Algunas de las incidencias más comunes comprenden, faltas ortográficas, no estandarización y/o desactualización de campos como dirección particular y teléfono, valores de marca y modelo que no existen, número de serie incorrecto y dirección física ficticia. Al ser impresa en papel, se deteriora con el tiempo producto a la manipulación y el descuido del usuario. Su formato da un amplio margen a la falsificación.

La información recopilada de las solicitudes es de alto valor para la Dirección de Seguridad Informática, se utiliza en auditorías, estudios analíticos y la identificación de dispositivos ante incidencias de hurto y delitos de seguridad informática. El tiempo empleado por los especialistas para realizar los análisis correspondientes es cada vez mayor.

Las áreas de aprobación por las que debe circular cada solicitud se desglosan en cuatro niveles consecutivos. Dichos niveles son:

1. **Área:** Área de la Universidad a la que pertenece el solicitante, puede ser una dirección, un centro de desarrollo o una facultad. El Directivo del área en cuestión es el responsable de autorizar o no la solicitud de uso de medio tecnológico en el primer nivel.
2. **Consejo Universitario:** Es el segundo nivel de aprobación. Sus miembros representan a las diferentes áreas de la Universidad en el Consejo Universitario. El miembro del Consejo

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

Universitario del área en cuestión es el responsable de autorizar o no la solicitud de uso de medio tecnológico en el segundo nivel.

3. **Departamento de Seguridad y Protección:** Área que se encarga de controlar el acceso a la Universidad, además de velar que no ocurra la extracción no autorizada de un activo físico. El Director de Seguridad y Protección es el responsable de autorizar o no la solicitud de uso de medio tecnológico en el tercer nivel.
4. **Dirección de Seguridad Informática:** Área encargada de velar por la seguridad informática en la Universidad. El Director de Seguridad Informática es el responsable de autorizar o no la solicitud de uso de un medio tecnológico en el cuarto nivel.

Existe un cúmulo de solicitudes pendientes a aprobación en cada área. Además de que ningún directivo conoce cuando ha firmado una solicitud para un mismo medio con anterioridad o cuando un propietario tiene a su nombre más de dos solicitudes activas. La distancia a recorrer entre los diferentes niveles de aprobación influye considerablemente en la demora del proceso.

Por otra parte, los usuarios no pueden informarse en tiempo real en qué estado se encuentra su solicitud. Además, no existe un mecanismo para que el propietario pueda establecer que su solicitud no es vigente o que cuando una persona cause baja de la Universidad su solicitud sea desactivada.

Ante esta situación se plantea como **problema de investigación:** ¿Cómo mejorar la eficiencia en el proceso de registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas? Definiéndose eficiencia como la fiabilidad e integridad de los datos registrados y el tiempo empleado en identificar un medio en específico (por cualquiera de sus atributos o del propietario).

Para dar solución a este problema se tomará como **objeto de estudio** los sistemas de inventario de dispositivos de hardware y sistemas de gestión de solicitudes. Derivándose como **campo de acción** el registro de medios tecnológicos personales en el Sistema de Gestión Universitaria en la Universidad.

Teniendo en cuenta lo antes planteado, el **objetivo general** de esta investigación es: Desarrollar un módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas.

Para cumplir con el objetivo general expuesto, se han trazado varios **objetivos específicos:**

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

- 1) Fundamentar la selección de la metodología, herramientas y tecnologías a utilizar en el desarrollo del módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas.
- 2) Realizar el análisis, diseño y desarrollo del módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas.
- 3) Realizar pruebas al módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas.

La definición de los objetivos y el análisis del problema, guiaron la obtención de la siguiente **hipótesis**: el desarrollo de un módulo para el registro de medios tecnológicos, mejora la eficiencia en el proceso de registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas.

Identificándose como **variable independiente**: módulo para el registro de medios tecnológicos, y como **variable dependiente**: eficiencia en el proceso de registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas.

Tabla 1: Operacionalización de variables. (Elaboración propia)

Variable conceptual	Dimensión	Indicadores	Sub-indicadores	Unidad de medida
Módulo para el registro de medios tecnológicos	Universidad de las Ciencias Informáticas	Funcionalidad	Funcionalidad	(%)
		Usabilidad	Instructibilidad	(%)
			Operatividad	(%)
Eficiencia en el proceso de registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas	Usuarios que utilizan el módulo	Fiabilidad e integridad de los datos registrados	Alta	95-100 (%)
			Media	85-95 (%)
			Baja	0-85 (%)
		Tiempo necesario para identificar un medio	Alta	0-10 (min)
			Media	10-20 (min)
			Baja	Más de 20 (min)

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

Con el propósito de cumplir con los objetivos específicos planteados se establecen las siguientes **tareas de la investigación**:

- 1) Delimitación del problema y caracterización del estado del arte de sistemas de registro similares.
- 2) Definición de los métodos y técnicas de investigación a emplear.
- 3) Descripción del entorno y la metodología de desarrollo a emplear.
- 4) Definición de los requisitos funcionales y no funcionales según el proceso de desarrollo aplicado.
- 5) Validación de los requisitos obtenidos aplicando técnicas.
- 6) Desarrollo del módulo.
- 7) Elaboración, ejecución y análisis de pruebas aplicables al módulo.

Los **métodos de investigación** utilizados para dar solución a la presente investigación son:

Teóricos:

- **Histórico-Lógico:** Posibilitó realizar un estudio sobre la existencia de sistemas similares (registro de medios tecnológicos), así como establecer conceptos relacionados al negocio desde su surgimiento hasta las definiciones más actuales.
- **Modelación:** Facilitó la representación, mediante el uso de diagramas, de la base de datos relacional y la interacción de los diferentes procesos que intervienen en el negocio, propiciando un mejor entendimiento del caso de estudio.
- **Análisis-Síntesis:** Mediante la lectura y comprensión de la documentación estudiada permitió contraer conocimientos para aplicar durante el desarrollo del módulo.

Empíricos:

- **Entrevista no estructurada:** Se concretaron encuentros con el cliente mediante los cuales se pudo definir la necesidad de la creación del módulo, establecer requisitos funcionales, observar el avance durante las fases de desarrollo y validación.
- **Análisis documental:** Permitted recopilar y valorar todos los conocimientos ya establecidos para conformar el marco teórico.

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

La presente investigación se encuentra estructurada en 3 capítulos de la siguiente forma:

Capítulo 1: Fundamentación teórica.

En este capítulo se realiza un estudio del estado del arte referente al problema de investigación. Se describen herramientas, metodologías y tecnologías empleadas para el desarrollo del módulo.

Capítulo 2: Análisis y diseño del módulo.

En este capítulo se hace una descripción del módulo. Se presentan los artefactos obtenidos una vez aplicada la fase de planeación de la iteración que propone la metodología Scrum. Se establece el entendimiento entre el cliente y el equipo de trabajo en relación con los objetivos a lograr, realizando un correcto análisis y diseño del módulo.

Capítulo 3: Implementación y validación del módulo.

En este capítulo se definen los estándares de codificación que debe cumplir el equipo de desarrollo, así como los métodos y técnicas para la realización de pruebas al módulo. Finalmente se describe la validación de hipótesis y los resultados alcanzados.

Posibles resultados:

Con la presente investigación se pretende obtener un módulo para el registro de los medios tecnológicos personales en la plataforma de Gestión Universitaria de la Universidad de las Ciencias Informáticas.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se realiza un estudio del estado del arte referente al problema de investigación. Como parte de dicho estudio se analizan diferentes sistemas que permiten el inventario de medios tecnológicos y la gestión de solicitudes. Además se describen herramientas, metodologías y tecnologías empleadas para el desarrollo del módulo.

1.1 Inventario de activos tecnológicos

Desde los primeros años del siglo XXI se hace evidente, para las organizaciones o entidades con cualquier tipo de razón social, la necesidad de manejar su información mediante la gestión de procesos que faciliten su competitividad. Dicha necesidad demanda el uso de herramientas tecnológicas que le permita a tales entidades estar a la vanguardia. La continua evolución tecnológica exige a las organizaciones, alinear sus procesos administrativos con la gestión tecnológica, facilitando muchas de las tareas inherentes a los diferentes procesos y funciones, lo cual representa optimización en costos y tiempo.

A medida que las empresas se desarrollan, su directorio activo de dominio crece exponencialmente, según se contratan nuevos empleados y se abren nuevos puestos. Los empleados a menudo requieren múltiples dispositivos para realizar sus tareas y se deben implementar nuevas terminales para proporcionar a los mismos un entorno de trabajo óptimo. Con una gran cantidad de dispositivos y usuarios, puede ser todo un reto mantener una visión general de todos los activos y usuarios en una empresa, y cómo están vinculados entre sí.

Lo antes expuesto es aún más relevante en esta época de la computación en la nube, los dispositivos móviles y el Internet de las cosas. Las empresas necesitan tener un inventario claro, preciso y constantemente actualizado de sus activos de hardware, para saber cuáles son sus recursos existentes y cómo se usan. Gracias al desarrollo científico y tecnológico existen herramientas que facilitan, que cualquier empresa pueda tener el control de sus activos hardware a lo largo de todo su ciclo de vida, hasta la retirada y la eliminación. Este proceso puede verse más claramente como inventario de activos tecnológicos. [4]

Para este caso de estudio el inventario es un término que puede causar confusión, debido que el mismo comúnmente se asocia con factores de costo, estado de un medio, componentes, entre otros. [5] En

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

cambio se empleará el término registro, que es más específico al objetivo de esta investigación. El módulo solo necesita procesar una pequeña parte de la información que puede brindar cada una de estas herramientas, por lo cual es factible su estudio al cubrir el área de conocimiento buscada. El análisis de dichas herramientas permite conocer las tendencias actuales en el registro de medios tecnológicos.

1.1.1 Sistemas de inventario tecnológico a nivel internacional

Network Inventory Advisor

Es una aplicación de inventario de activos en la red desarrollada y con soporte de la compañía ClearApps LLC, compañía que surge en 1999 en Elk Grove, California. La primera versión registrada con el mismo nombre data del 9 de diciembre de 2013.

Es una aplicación de escritorio para sistema operativo Windows, el cual brinda una amplia gama de servicios. Busca en remoto los datos sobre la CPU, memoria, sistema, audio y video, los dispositivos periféricos y otros detalles del dispositivo. Con él se puede planificar fácilmente actualizaciones masivas, solucionar problemas del dispositivo, conocer la marca y el modelo del equipo, realizar auditorías de red automatizadas en cuestión de minutos y escanear cientos de computadoras de forma rápida, segura y fiable.

El inventario de aplicaciones desde la red junto con la auditoría y cumplimiento de las licencias son otras características fundamentales, puede hacer fácilmente un seguimiento de las instalaciones, versiones de aplicaciones, licencias y servicios en todos los ordenadores. [6]

Este sistema cubre parte de las funcionalidades deseadas del módulo a desarrollar y agrega otras que no son necesarias. No es capaz de administrar las áreas de aprobación que es un concepto propio de la investigación. Es una herramienta privativa cuya adquisición constituiría un costo de 1.188,00 USD a la institución. El uso de esta herramienta entra en contradicción con el lineamiento 108 del Partido Comunista de Cuba referente al desarrollo de soluciones propias para garantizar la soberanía tecnológica. [7]

Sistema Para la Administración de Recursos Tecnológicos de la FUNLAM (SPARTEC)

SPARTEC es una aplicación desarrollada en la Fundación Universitaria Luis Amigó de Medellín (FUNLAM) Colombia en el año 2014.

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

Implementada con el objetivo de controlar los recursos tecnológicos de la institución mediante la técnica de inventario. Está diseñada para su uso interno en FUNLAM, con las definiciones y casos de usos específicos. La construcción de la herramienta informática documentada, en este caso, se fundamentó en el uso de la plataforma PHP, que es un lenguaje de programación que permite técnicas de orientación a objetos y está enfocado a la plataforma web. [4]

Cubre parte de las funcionalidades deseadas del módulo a desarrollar y agrega otras que no son necesarias. No es capaz de administrar las áreas de aprobación que es un concepto propio de la investigación. A pesar de estar correctamente documentada no se cuenta con el código base de esta aplicación web. El uso de esta herramienta entra en contradicción con el lineamiento 108 del Partido Comunista de Cuba referente al desarrollo de soluciones propias para garantizar la soberanía tecnológica. [7]

Sistema de Gestión Integrado. Análisis del procedimiento “Gestión de Cambios Ambiente Productivo”. (SGI-GCAP)

Documento elaborado y aprobado en 2017 por Superintendencia de Sociedades, organismo técnico, adscrito al Ministerio de Comercio, Industria y Turismo de Colombia. Su objetivo es preservar la disponibilidad de los servicios gubernamentales soportados por la infraestructura tecnológica mediante el empleo de esta normativa. Este procedimiento define la planeación, coordinación, monitoreo y comunicación de los cambios que afectan a los recursos tecnológicos y Sistemas de Información, para minimizar el impacto en el ambiente de producción, en los compromisos de niveles de servicio y orientar a la persona que va a ingresar a la sala de cómputo a manipular algún elemento.

Permite gestionar los procesos de cambio que impliquen: dispositivos, aplicaciones y equipos de comunicación, sistemas de aplicaciones, aplicaciones “en producción”, toda la documentación y procedimientos asociados con la ejecución, soporte y mantenimiento de los sistemas en producción. Permite configurar las diferentes áreas existentes en la empresa, realizar reportes e intercambiar información entre áreas y procesos. Establece diferentes roles dentro del proceso, entre ellos el Gestor de Cambios y el Responsable del Cambio. El Gestor de Cambios recibirá las Peticiones de Cambio (vía correo electrónico), comprobará su completitud y corrección, y las analizará brevemente, filtrando aquellas que considere totalmente impracticables. En función de la prioridad y la categoría asignadas, canalizará la petición a un determinado tratamiento para su evaluación y aprobación. El Responsable del Cambio es el

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

encargado de aprobar o no el cambio dentro de su área, o promoverlo a evaluación de un comité de cambios en función de la prioridad y la categoría asignadas. [8]

A pesar de no ser una aplicación este documento describe detalladamente la gestión de solicitudes, que es una funcionalidad deseada que debe implementar el módulo a desarrollar. Da una visión clara de los roles que interactúan en el proceso con su descripción. Explica e ilustra el flujo del proceso. Sirve como base para el entendimiento de las características que debe implementar un sistema de registro y gestión de solicitudes.

1.1.2 Sistemas de inventario tecnológico a nivel nacional

Sistema Automatizado para el Control de Medios Informáticos (SACMI)

Este sistema fue desarrollado en Santa Clara en el año 2014 como resultado de la tesis de grado de Darién Rodríguez Prieto, tutorado por el Dr. Carlos E. García González. Es un sistema encargado de inventariar y llevar la caracterización de los medios de tecnológicos pertenecientes a la Universidad Central Marta Abreu de las Villas (UCLV). En la misma se recogen cualidades técnicas del dispositivo y sus aplicaciones, así como su ubicación y el responsable del dispositivo. Permite generar una planilla impresa con todos los datos recogidos para cada equipo. Es una aplicación de escritorio desarrollada en el lenguaje de programación Java, apoyada en la herramienta AIDA64. [9]

Cubre parte de las funcionalidades deseadas del módulo a desarrollar y agrega otras que no son necesarias. No es capaz de administrar las áreas de aprobación que es un concepto propio de la investigación, ni recorrer el flujo de aprobación.

Plataforma de Gestión de Servicios Telemáticos en GNU/Linux. Sistema de Inventario de Hardware y Software. Módulo obtención de información. (GRHS)

Este sistema fue desarrollado en la UCI en el año 2010 como resultado de la tesis de grado de Camilo Denis González y Alié Castillo Ruiz, tutorados por Ing. Ramón Alexander Anglada Martínez.

El sistema propuesto se divide lógicamente en dos partes fundamentales, un servidor y un cliente o agente. El agente está destinado a la recolección de información del dispositivo como lo son las características físicas, las aplicaciones instaladas y la configuración de la PC agente. Una PC agente es aquella en la que está instalado el agente o cliente del Sistema de Inventario de Hardware y Software. El sistema servidor recibe los inventarios enviados por los clientes, almacena la información recibida en una

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

base de datos, analiza si el inventario es una incidencia o no, de ser incidencia verifica si existe periodo de cambio. Si no se ha definido periodo de cambio, ejecuta alertas según la configuración establecida por el administrador de red responsable del control sobre los recursos informáticos.

Para la presente investigación se estudia específicamente el sistema servidor, que se encarga de almacenar y analizar los datos brindados por la aplicación agente. Este sistema recoge cualidades técnicas de hardware y software de cada dispositivo, así como permite generar alertas ante incidencias. Es un sistema de tecnología de escritorio desarrollado en el lenguaje de programación Python. [10]

Cubre parte de las funcionalidades deseadas del módulo a desarrollar y agrega otras que no son necesarias. No es capaz de administrar las áreas de aprobación que es un concepto propio de la investigación, ni recorrer el flujo de aprobación.

El análisis realizado a las herramientas de inventario de medios tecnológicos permitió elaborar un resumen en forma de tabla teniendo en cuenta los siguientes parámetros:

- **Procedencia:** Referente a la procedencia del producto. Puede tomar valores Nacional y Extranjero.
- **Gestión de solicitudes:** Referente a si la herramienta puede gestionar solicitudes de uso de medios tecnológicos realizadas por los usuarios.
- **Registro de medios:** Referente a si la herramienta logra el registro de medios tecnológicos.
- **Web:** Referente a si la herramienta es una aplicación web o no.

Tabla 2: Resumen de Sistemas Homólogos. (Elaboración propia)

Herramienta	Procedencia	Gestión de solicitudes	Registro de medios	Web
Network Inventory Advisor	Extranjero	No	Si	No
SPARTEC	Extranjero	No	Si	Si
SIG-GCAP	Extranjero	Si	Si	No
SACMI	Nacional	No	Si	Si
GRHS	Nacional	No	Si	No

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

A partir del estudio del estado del arte realizado, el autor arriba las siguientes impresiones:

- Network Inventory Advisor es una herramienta de corte privativo que no cubre totalmente las funcionalidades deseadas para el caso de estudio, además de no estar desarrollado en una tecnología web.
- Se cuenta con la documentación pertinente para aplicar re-ingeniería a los sistemas SPARTEC, SACMI, pero los mismos no cubren totalmente las funcionalidades deseadas.
- La tecnología empleada en los sistemas SPARTEC, SACMI no les permiten ser adheridos directamente al Sistema de Gestión Universitaria. Aunque pudiera aplicárseles un proceso de migración, no se cuenta con el código base de los mismos.
- GRHS es una aplicación de escritorio propiedad de la UCI. La misma no permite configurar ni ejecutar el flujo de aprobación.
- SGI-GCAP brinda información clara y valiosa de cómo funciona el proceso de gestión de solicitudes además de lograr el registro de medios. El procedimiento descrito se asemeja en gran medida a la situación actual de la Universidad.

Por todo lo antes expuesto, se puede resumir que las herramientas señaladas no cumplen con el objetivo de la presente investigación. Ninguna reúne las funcionalidades necesarias y suficientes para ser utilizada directamente como propuesta de solución. Sin embargo, cada una contribuyó a una mejor comprensión de las principales características que debe cumplir una aplicación de este tipo.

1.2 Entorno de desarrollo

A continuación se hace una descripción de las herramientas, lenguajes, metodologías y tecnologías específicas empleadas para guiar el desarrollo del módulo. Las mismas fueron definidas por la Dirección de Informatización de la UCI para ser empleadas en su marco de trabajo.

1.2.1 Lenguaje de modelado

Lenguaje Unificado de Modelado (UML por sus siglas en inglés) es un lenguaje de modelado estandarizado que permite a los desarrolladores especificar, visualizar, construir y documentar artefactos de un sistema de software. Por lo tanto, UML hace que estos artefactos sean escalables, seguros y robustos en ejecución. Utiliza la notación gráfica para crear modelos visuales de sistemas de software. Para la investigación se emplea Lenguaje Unificado de Modelado en su versión 2.5. [11]

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

1.2.2 Herramienta de modelado

Visual Paradigm 15.0 es una herramienta de Ingeniería de Software Asistida por Computación (CASE por sus siglas en inglés). La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Es una potente herramienta multiplataforma de diseño y gestión fácil de usar para sistemas de Tecnologías de la Información. Facilita una excelente interoperabilidad con otras herramientas CASE y la mayoría de los Entornos de Desarrollo Integrado (IDE por sus siglas en inglés). [12]

1.2.3 Herramienta para el control de versiones

Para la presente investigación se emplea el **repositorio GitLab 10.7.1**, con el **cliente Git en su versión 2.17.1**. El mismo es un Sistema de Control de Versiones Distribuido el cual permite el trabajo colaborativo y previene que se interrumpa el servicio. Es capaz de gestionar grandes proyectos eficientemente destacándose la velocidad de trabajo. Posee un gran soporte para el desarrollo no lineal y presenta un diseño sencillo. [13]

1.2.4 Lenguajes de programación

PHP acrónimo recursivo en inglés de PHP Hypertext Preprocessor (preprocesador de hipertexto), es un lenguaje de programación de propósito general de código del lado del servidor. Originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en un documento HTML. Para el desarrollo del módulo se emplea la versión PHP 7.2. [14]

JavaScript consiste en un lenguaje de programación interpretado, que habitualmente se utiliza en sitios web para ejecutar acciones del lado del cliente, estando embebido en el código fuente de la página web. Técnicamente, constituye un dialecto del estándar ECMAScript, propuesto por la entidad internacional de estándares de información y comunicación ECMA International y diseñado inicialmente por Netscape y, posteriormente, por la Fundación Mozilla. También constituye un estándar ISO. Permite crear contenido nuevo y dinámico, controlar archivos de multimedia, crear imágenes animadas, entre muchas otras funcionalidades. Para el desarrollo del módulo se utiliza específicamente la biblioteca multiplataforma de JavaScript conocida como jQuery 1.9.2, que será definida posteriormente. [15]

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

HTML acrónimo recursivo en inglés HTML HyperText Markup Language (lenguaje de marcas de hipertexto) es un lenguaje de marcado que establece una estructura para dar un sentido al contenido web, por ejemplo define párrafos, cabeceras, tablas, imágenes y videos en la página. Para el desarrollo del módulo se utiliza HTML 4.01. [16]

CSS acrónimo recursivo en inglés de CSS Cascading Style Sheets (hojas de estilo en cascada) es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas. Permite colocar colores de fondo, fuentes y marginando al contenido en múltiples columnas, entre otras funcionalidades relacionadas a la apariencia visual. Para el desarrollo del módulo es utilizado CSS3. [17]

1.2.5 Biblioteca

Una biblioteca es un conjunto de recursos (algoritmos) prefabricados, que pueden ser utilizados por el programador para realizar determinadas operaciones. Las declaraciones de las funciones utilizadas en estas bibliotecas, junto con algunas macros y constantes predefinidas que facilitan su utilización, se agrupan en ficheros de nombres conocidos que suelen encontrarse con extensiones .lib, .js, .dll, entre otras.

jQuery 1.9.2 es una biblioteca de JavaScript que permite simplificar el proceso de desarrollo de una aplicación web. La misma será ampliada en el marco de trabajo de la presente investigación. [18]

1.2.6 Marco de trabajo

El Departamento de Desarrollo de la Dirección de Informatización (DIN) define como marco de trabajo **GUUD2.0**, que permite la creación de aplicaciones web escritas en PHP. Constituye un híbrido entre el marco de trabajo de PHP **CodeIgniter** en su versión 1.7.2 y la librería de JavaScript, **jQuery** en su versión 1.9.2.

Para la selección de **CodeIgniter** se tuvo en cuenta que es un producto de software libre, por lo que su uso es ilimitado. También se tuvo en cuenta algunas de sus principales características como: versatilidad, compatibilidad con la mayoría de los Sistemas Operativos, facilidad de instalación, flexibilidad, ligereza y documentación amplia y específica.

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

Por otra parte se une la librería **jQuery 1.9.2** que ofrece un conjunto de funcionalidades, para el fácil manejo de JavaScript, pues de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio. Algunas de sus principales características son:

- Permite la manipulación de elementos DOM ¹
- Realiza la manipulación de hojas de estilos CSS
- Se pueden lograr efectos y animaciones.
- Funcionalidades para el uso de AJAX ²

1.2.7 Formato de intercambio de datos

JSON acrónimo de JavaScript Object Notation (Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript aunque hoy se considera un formato de lenguaje independiente. JSON se emplea habitualmente en entornos donde el tamaño del flujo de datos entre cliente y servidor es de vital importancia. [19] Por lo antes expuesto, se emplea el formato JSON para el desarrollo del módulo.

1.2.8 Servidor de aplicaciones web

El servidor **nginx 1.14** es un servidor web ligero de código abierto. Es multiplataforma, por lo que corre en sistemas tipo Unix, GNU/Linux, BSD, Solaris, MacOS, Windows, entre otros. Conocido por su alto rendimiento, estabilidad, configuración simple y bajo consumo de recursos. A diferencia de los servidores tradicionales, nginx no se basa en subprocesos para manejar las solicitudes. En su lugar, utiliza una arquitectura más escalable basada en eventos (asíncrona). Esta arquitectura utiliza cantidades pequeñas, pero más importantes, predecibles de memoria bajo carga.

¹ **DOM** (Document Object Model) es un estándar que permite acceder y manipular los elementos de una página web mediante algún lenguaje de programación web, como JavaScript o su librería jQuery.

² **AJAX** (Asynchronous Javascript and XML) es una técnica de desarrollo web que, al combinar una serie de tecnologías independientes, permite intercambiar información entre el servidor y el cliente (un navegador web) de forma asíncrona.

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

Nginx fue inicialmente desarrollado con el fin explícito de superar el rendimiento ofrecido por el servidor web Apache.¹⁴ Como servidor de archivos estáticos, nginx usa menos memoria que Apache, y puede manejar más solicitudes por segundo. [20]

1.2.9 Gestor de bases de datos

PostgreSQL es un Sistema Gestor de Bases de Datos Objeto-Relacional (de sus siglas en inglés ORDBMS). Se distribuye bajo la licencia PostgreSQL, una licencia de código abierto liberal. PostgreSQL incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. Es una herramienta de diseño y gestión de bases de datos. En el desarrollo del módulo se emplea Postgres10. [21]

PgAdmin III es la plataforma de desarrollo y administración de código abierto más popular y rica en funciones para PostgreSQL. Puede ejecutarse como una aplicación web o de escritorio. Es una herramienta multiplataforma capaz de ejecutarse en sistemas con Microsoft Windows, Linux y MacOS. Está diseñado para múltiples versiones de PostgreSQL y sus derivados. Presenta una extensiva documentación y una comunidad virtual. [22] Dentro de sus herramientas destacan:

- Herramienta de consulta con resaltado de sintaxis en color
- Visualización gráfica del plan de consulta.
- Depurador de lenguaje de procedimiento

1.2.10 Entorno de Desarrollo Integrado

NetBeans IDE 8.2 es un Entorno de Desarrollo Integrado (IDE por sus siglas en inglés), modular, de base estándar (normalizado), escrito en el lenguaje de programación Java. El proyecto NetBeans consiste en un IDE de código abierto y una plataforma de aplicación, las cuales pueden ser usadas como una estructura de soporte general (framework) para compilar cualquier tipo de aplicación. La programación mediante NetBeans se realiza a través de componentes de software modulares, también llamados módulos. Constantemente valida la calidad del código y ofrece soluciones inmediatas para los problemas encontrados en todos los niveles, desde la instrucción individual para arquitectura global, utilizando las inspecciones de código avanzado y análisis de matriz de dependencia. [23]

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

1.2.11 Metodología de desarrollo de software

Una metodología de desarrollo de software tiene como principal objetivo aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo. No existe una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable. Las metodologías de desarrollo se pueden dividir en dos grupos de acuerdo con sus características y los objetivos que persiguen: ágiles y tradicionales (o robustas).

- **Metodología ágil:** Se basa en una fuerte y constante interacción, donde clientes y desarrolladores trabajan juntos, estableciéndose así una estrecha comunicación. Estas metodologías están orientadas al resultado del producto y no a la documentación. Exige que el proceso sea adaptable, permitiendo realizar cambios de último momento.
- **Metodología tradicional:** Están guiadas por una fuerte planificación. Centran su atención en llevar una documentación exhaustiva de todo el proceso de desarrollo y en cumplir con un plan de proyecto, definido en la fase inicial del mismo.

Para la implementación del módulo se decide utilizar una metodología ágil. Éstas permiten entregar productos de calidad con los costes y tiempos pactados, y las metodologías tradicionales ya no bastan para este cometido, no se adaptan a las nuevas expectativas de los usuarios y a las exigencias del mercado. Las metodologías ágiles permiten centrarse en la correcta obtención del módulo sin importar cuán documentado esté, enfatizando en la rapidez y la calidad con que se desarrolle el mismo. [24]

Metodología Scrum

Es una metodología ágil y flexible que permite gestionar el desarrollo de software y cuyo objetivo es maximizar el retorno de la inversión para quien la emplea. Se fundamenta básicamente en construir la funcionalidad de mayor valor para el cliente y en los principios de supervisión continua, auto-gestión, adaptación e innovación.

Propone un desarrollo basado en iteraciones, es decir, fases en las que se dividen los proyectos. Cada iteración debe estar compuesta de varias actividades puntuales orientadas a un mismo fin. Pero dicho fin no es definitivo, siempre es parcial y se relaciona con la siguiente iteración. Los plazos para la ejecución de cada iteración varían en función de la naturaleza de cada proyecto. Sin embargo, lo más habitual es

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

que no sobrepasen las 2 o 4 semanas. Los resultados se entregan al final de cada iteración. Es decir, cada uno de los resultados va sumando para el resultado final. Los distintos agentes vinculados al proyecto interactúan de forma activa en cada fase. El cliente hace recomendaciones, observaciones, críticas o señala eventuales fallos en cada nueva iteración.

Entre las ventajas que ofrece esta metodología están: cumplimiento de expectativas, flexibilidad a cambios, reducción del tiempo, mayor productividad, maximiza el retorno de la inversión, reducción de riesgos. [25]

Actividades estructurales de Scrum

- **Planificación de la iteración:** Se realiza el primer día en reunión con el cliente. Tiene dos partes:
 - **Selección de requisitos:** se hace el levantamiento de requisitos con el cliente y se seleccionan los más prioritarios que el equipo podrá completar en la iteración, de manera que puedan ser entregados si el cliente lo solicita.
 - **Planificación de la iteración:** el equipo elabora la lista de tareas de la iteración necesarias para desarrollar los requisitos seleccionados. La estimación de esfuerzo se hace de manera conjunta y los miembros del equipo se auto-asignan las tareas, logrando una distribución de los objetivos de trabajo.
- **Ejecución de la iteración:** Cada día el equipo realiza una reunión de sincronización, normalmente delante de un tablero físico o pizarra (Scrum Task Board). El equipo inspecciona el trabajo que el resto está realizando (dependencias entre tareas, progreso hacia el objetivo de la iteración, obstáculos que pueden impedir este objetivo) para poder hacer las adaptaciones necesarias que permitan cumplir con la previsión de objetivos a mostrar al final de la iteración.
- **Inspección y adaptación:** El último día de la iteración se realiza la reunión de revisión de la iteración. Tiene dos partes:
 - **Revisión (demostración):** El equipo presenta al cliente los requisitos completados en la iteración, en forma de incremento de producto preparado para ser entregado con el mínimo esfuerzo. En función de los resultados mostrados y de los cambios que haya habido en el contexto del proyecto, el cliente realiza las adaptaciones necesarias de manera objetiva, ya desde la primera iteración, re-planificando el proyecto.

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

- **Retrospectiva:** El equipo analiza cómo ha sido su manera de trabajar y cuáles son los problemas que podrían impedirle progresar adecuadamente, mejorando de manera continua su productividad. El Facilitador se encargará de eliminar o escalar los obstáculos identificados que estén más allá del ámbito de acción del equipo.

Metodología XP

La Programación Extrema (XP por sus siglas en inglés) es una de las metodologías ágiles más populares. Se ha demostrado que es muy exitosa en muchas compañías de diferentes tamaños e industrias en todo el mundo.

Es exitosa porque enfatiza la satisfacción del cliente. En lugar de entregar todo lo que pueda desear en una fecha lejana en el futuro, este proceso entrega el software que necesita a medida que lo necesita durante iteraciones que duran entre 1 y 3 semanas. El proceso de iteraciones permite a sus desarrolladores responder con confianza a los cambios en los requisitos del cliente, incluso al final del ciclo de vida.

La Programación Extrema enfatiza el trabajo en equipo. Los gerentes, clientes y desarrolladores son socios iguales en un equipo colaborativo. Implementa un entorno simple pero efectivo que permite a los equipos ser altamente productivos. El equipo se auto-organiza alrededor del problema para resolverlo de la manera más eficiente posible. Aplica la re-fabricación de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.

La programación extrema usa un enfoque orientado a objetos como paradigma preferido de desarrollo, y engloba un conjunto de reglas y prácticas que ocurren en el contexto de cuatro actividades estructurales: planeación, diseño, codificación y pruebas. Las reglas establecen expectativas entre los miembros del equipo, pero no son la meta final en sí mismas. Estas reglas definen un entorno que promueve la colaboración y el empoderamiento del equipo. Una vez logrado, el trabajo en equipo productivo continuará incluso a medida que se cambien las reglas para adaptarse a las necesidades específicas de su empresa.

[25]

Actividades estructurales de XP

- **Planeación:** se basa en recabar requerimientos que permite que los miembros técnicos del equipo XP entiendan el contexto del negocio para el software y adquieran la sensibilidad de la salida y

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

características principales y funcionalidad que se requiere. Se logra a partir de obtener las Historias de Usuario (HU), que son tomadas por los desarrolladores para modelar los requisitos.

Los clientes y desarrolladores trabajan juntos para decidir cómo agrupar las historias en la siguiente entrega (el siguiente incremento de software) que desarrollará el equipo XP. Una vez que se llega a un compromiso sobre la entrega (acuerdo sobre las historias por incluir, la fecha de entrega y otros aspectos del proyecto), el equipo XP ordena las historias que serán desarrolladas en una de tres formas:

- Todas las historias se implementarán de inmediato (en pocas semanas).
 - Las historias con más valor entrarán a la programación de actividades y se implementarán en primer lugar.
 - Las historias más riesgosas formarán parte de la programación de actividades y se implementarán primero.
- **Diseño:** El diseño XP sigue rigurosamente el principio MS (mantenlo sencillo). Un diseño sencillo siempre se prefiere sobre una representación más compleja. Además, el diseño guía la implementación de una historia conforme se escribe: nada más y nada menos. Se desalienta el diseño de funcionalidad adicional porque el desarrollador supone que se requerirá después. XP estimula el uso de las tarjetas CRC como un mecanismo eficaz para pensar en el software en un contexto orientado a objetos. Las tarjetas CRC (Clase-Responsabilidad-Colaborador) identifican y organizan las clases orientadas a objetos que son relevantes para el incremento actual de software. Las tarjetas CRC son el único producto del trabajo de diseño que se genera como parte del proceso XP.
 - **Codificación:** XP recomienda que dos personas trabajen juntas en una estación de trabajo con el objeto de crear código para una historia. A medida que las parejas de programadores terminan su trabajo, el código que desarrollan se integra con el trabajo de los demás. En ciertos casos, esto lo lleva a cabo a diario un equipo de integración. En otros, las parejas de programadores tienen la responsabilidad de la integración. Esta estrategia de “integración continua” ayuda a evitar los problemas de compatibilidad de interfaces y brinda un ambiente de “prueba de humo” que ayuda a descubrir a tiempo los errores.
 - **Pruebas:** La creación de pruebas unitarias antes de que comience la codificación es un elemento clave del enfoque de XP, ya que esto asegura la calidad del software.

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

El estudio de las metodologías Scrum y XP permitió determinar que ambas son factibles para guiar el desarrollo del módulo. La Dirección de Informatización (DIN) requiere el empleo de la metodología Scrum para poder incluir el módulo desarrollado en el expediente de proyecto y garantizar el soporte al mismo, dado que esta es la metodología empleada por los especialistas del centro. Sin embargo, de la metodología XP se emplearán artefactos que otorgarán un valor agregado a la documentación entregada.

1.3 Conclusiones del capítulo

En el presente capítulo se abordaron elementos referentes al caso de estudio en cuestión, en tal sentido se puede arribar la siguiente conclusión:

- El análisis de sistemas homólogos, permitió identificar las tendencias en el desarrollo de herramientas informáticas para el inventario de medios tecnológicos y la gestión de solicitudes, así como comprender la necesidad de desarrollar un módulo adaptado a las necesidades específicas de la institución.

CAPÍTULO 2: ANÁLISIS Y DISEÑO

En el presente capítulo se hace una descripción del módulo a desarrollar. Se presentan los artefactos obtenidos, una vez aplicada la fase de planeación de la iteración que propone la metodología Scrum. Son plasmados los diferentes roles que interactúan con el módulo. Se presentan artefactos auxiliares obtenidos de la metodología XP. Se establece el entendimiento entre el cliente y el equipo de trabajo en relación con los objetivos a lograr, realizando un correcto análisis y diseño del módulo.

2.1 Objeto de automatización

En el análisis del problema se identificó la solicitud de uso de medio tecnológico personal como el principal proceso que sustenta la actividad del negocio. A continuación se describe dicho proceso.

2.1.1 Descripción del proceso

Proceso Solicitud de uso de medio tecnológico personal: Surge de la necesidad del control de los medios tecnológicos personales que consumen servicios de red en la Universidad. Como precondition se cuenta con una planilla de registro de medio tecnológico, amparado en el marco legal del Reglamento de Seguridad Informática de la Universidad de las Ciencias Informáticas y bajo la supervisión del Director de Seguridad Informática.

El proceso comienza cuando el solicitante se dirige a su Directivo del Área para realizar la solicitud de registro de un medio tecnológico. El Directivo del Área cuenta a priori con un modelo de la solicitud impreso o lo descarga del sitio de Seguridad Informática de la Universidad y lo imprime. Luego rellena la planilla con los datos personales del solicitante y específicos del medio. Una vez completada, el solicitante se retira y espera a ser notificado. En ese instante comienza el proceso de aprobación, el cual se desglosa en 4 niveles describiendo un orden por el que debe transitar la solicitud. Dichos niveles son:

1. Área.
2. Consejo Universitario.
3. Dirección de Seguridad y Protección.
4. Dirección de Seguridad Informática.

El responsable en cada nivel comprueba si posee validez la planilla y si la persona solicitante está autorizada, de ser así la aprueba con firma y cuño, pasando al próximo nivel. De no ser aprobada, se cancela la solicitud y termina inmediatamente el proceso. Cuando una solicitud ha sido aprobada por

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

todos los niveles termina el proceso y es enviada al Área para que sea entregada al usuario que emitió la solicitud. En caso del solicitante ser subordinado directo del rector, el ciclo de aprobación de su solicitud comienza directamente desde el segundo nivel referente al Miembro del Consejo Universitario que representa su área.

La figura 1 del diagrama de procesos muestra lo antes descrito:

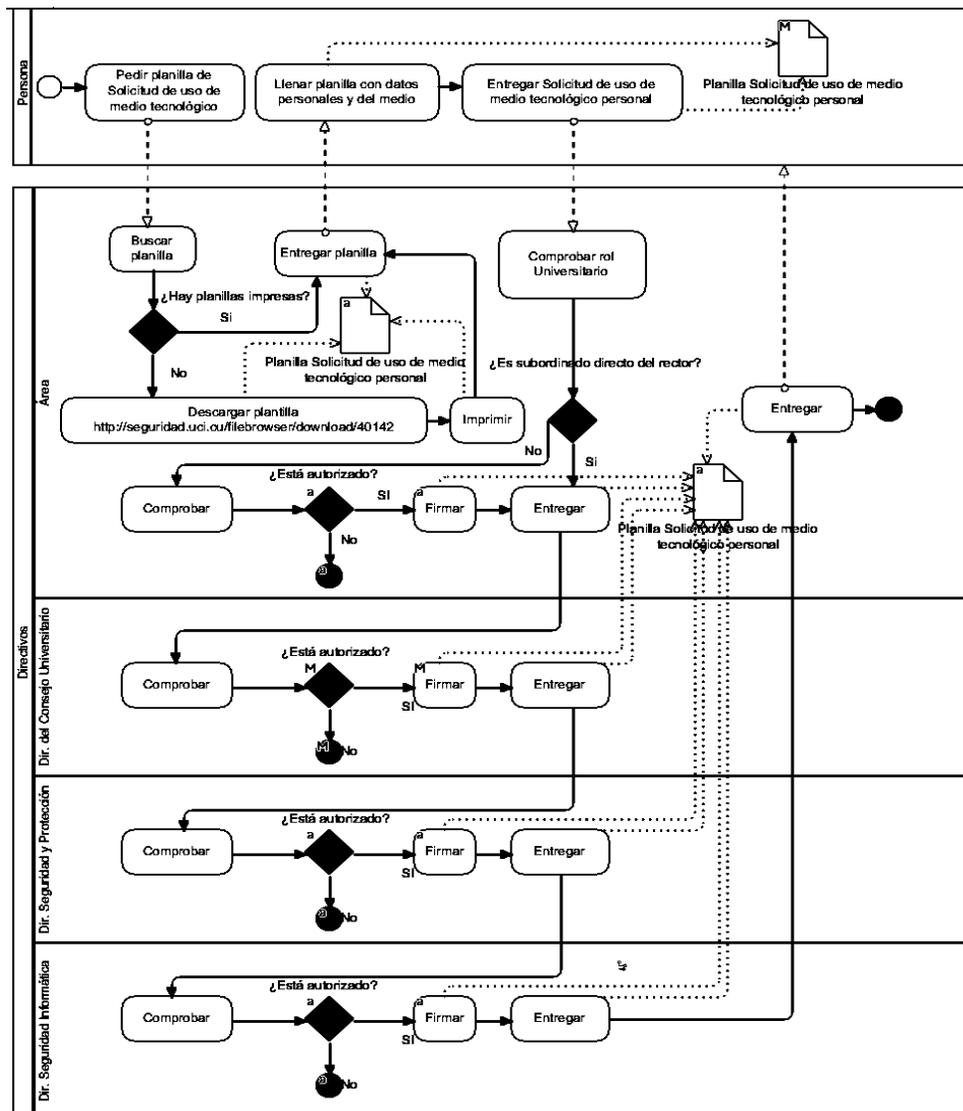


Figura 1: Diagrama de Proceso Solicitud de uso de medio tecnológico personal

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

La figura 2 del diagrama del modelo conceptual complementa lo antes descrito:

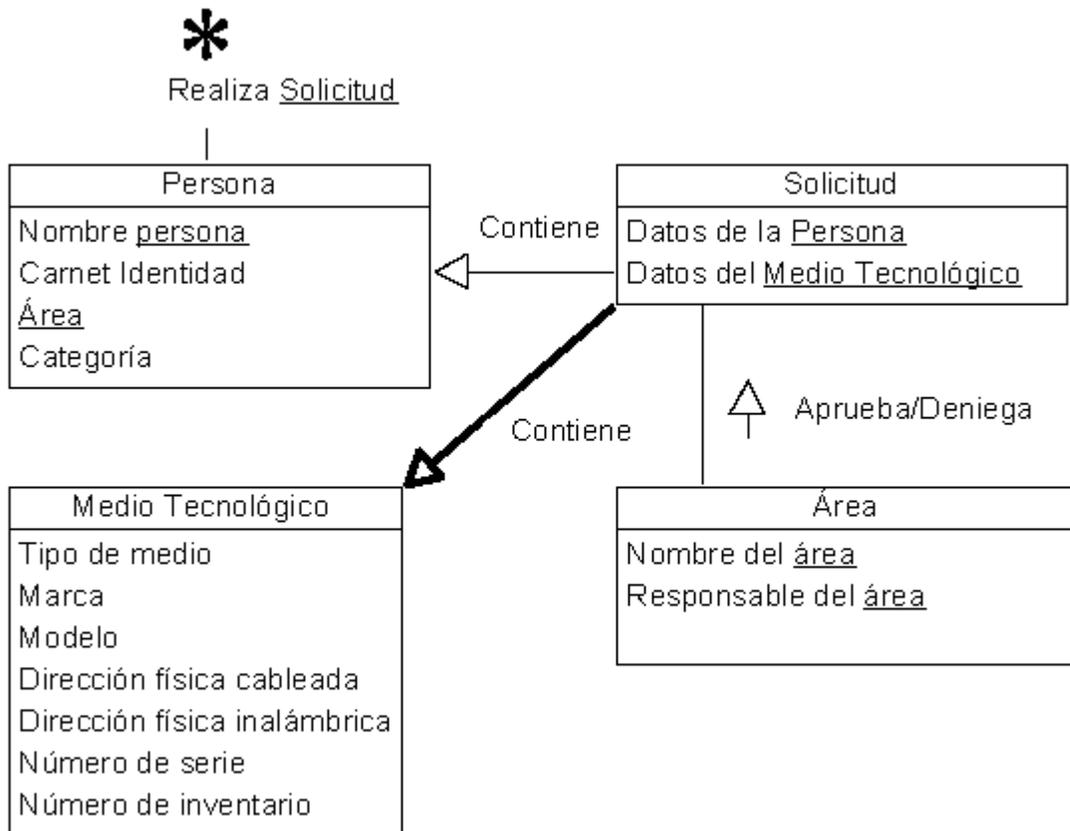


Figura 2: Diagrama Modelo Conceptual.

2.2 Descripción de la propuesta solución

El módulo gestiona el proceso de solicitud de uso de un conjunto de medios tecnológicos aprobados dentro de la Universidad de las Ciencias Informáticas. Se crea con el objetivo de responder a la necesidad de informatización de los procesos de la organización para un correcto funcionamiento y centralización de la información.

Tiene como objetivos específicos:

1. Permitir el acceso al módulo a las personas involucradas.

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

2. Lograr la gestión de los elementos de configuración general del módulo.
3. Gestionar las solicitudes de uso de medios tecnológicos dentro de la entidad.
4. Lograr el registro de medios tecnológicos dentro de la entidad.
5. Notificar a los usuarios las acciones importantes realizadas en el módulo.

Se encarga de llevar el proceso de registro de medios tecnológicos personales. El módulo permite:

1. Gestionar las solicitudes de uso de medios tecnológicos personales.
2. Gestionar las solicitudes creadas por los usuarios en los diferentes niveles de aprobación.
3. Gestionar tipos de medios y marcas de los medios tecnológicos.
4. Configurar los tipos de medios permitidos por roles universitarios
5. Gestionar las áreas de aprobación.
6. Gestionar estados de aprobación.
7. Configurar el flujo de aprobación.
8. Deshabilitar las solicitudes aprobadas de personas con estado de baja.
9. Generar documento de aprobación del medio solicitado por el usuario.
10. Generar reportes.

2.3 Planeación

La planeación es la etapa inicial de todo proyecto basado en la metodología Scrum. El equipo de trabajo conjunto con el cliente se enfoca en recabar requerimientos, lo cual permite lograr un entendimiento mutuo sobre contexto del negocio y el entorno en el que se va a desenvolver. Se logra a partir de obtener las Historias de Usuario, que son tomadas por los desarrolladores para modelar los requisitos. El cliente establece la prioridad de cada Historia de Usuario y los programadores realizan la estimación del esfuerzo que costará implementar cada una de ellas. Además se identifican el número y el tamaño de las iteraciones, se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma de ejecución en conjunto. [25]

2.3.1 Requisitos Funcionales

Son declaraciones de los servicios y funciones que provee un sistema. Establecen a grandes rasgos o de manera descriptiva qué es lo que debe hacer el sistema. En ocasiones también explican qué no debe hacer el sistema ante determinadas situaciones específicas. [26]

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

De la reunión con el cliente se obtuvo un total de 36 requisitos funcionales los cuáles se listan en la tabla 4. Estos requisitos son descritos en las Historias de Usuario.

Tabla 4: Requisitos funcionales. (Elaboración propia)

N°	Nombre	N°	Nombre
RF 1	Listar tipos de medio tecnológico	RF 19	Crear solicitud de uso de medio tecnológico personal
RF 2	Crear tipo de medio tecnológico	RF 20	Modificar solicitud de uso de medio tecnológico personal
RF 3	Modificar tipo de medio tecnológico	RF 21	Ver detalles de la solicitud de uso de medio tecnológico personal
RF 4	Ver detalles del tipo de medio tecnológico	RF 22	Cancelar solicitud de uso de medio tecnológico personal
RF 5	Modificar cantidad del tipo de medio tecnológico por roles universitarios	RF 23	Listar solicitudes de uso de medio tecnológico por área de aprobación
RF 6	Listar marcas	RF 24	Valorar solicitudes de uso de medio tecnológico por área de aprobación
RF 7	Crear marca	RF 25	Ver detalles de la solicitud de uso de medio tecnológico por área
RF 8	Modificar marca	RF 26	Deshabilitar solicitudes activas de personas de baja
RF 9	Ver detalles de la marca	RF 27	Listar reporte de solicitudes por medio tecnológico
RF 10	Listar estados de solicitud	RF 28	Exportar reporte de solicitudes por medio tecnológico
RF 11	Crear estado de solicitud	RF 29	Listar historial de solicitudes
RF 12	Modificar estado de solicitud	RF 30	Exportar historial de solicitudes
RF 13	Ver detalles del estado de solicitud	RF 31	Listar reporte de cantidad de medios
RF 14	Listar áreas de aprobación	RF 32	Exportar reporte de cantidad de medios
RF 15	Crear área de aprobación	RF 33	Listar solicitudes pendientes a impresión

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

RF 16	Modificar área de aprobación	RF 34	Exportar documento de autorizo de uso de medio tecnológico personal
RF 17	Ver detalles del área de aprobación	RF 35	Listar solicitudes del sistema
RF 18	Listar solicitudes de uso de medio tecnológico personal (Usuario)	RF 36	Desactivar solicitud

2.3.2 Requisitos no Funcionales

Son propiedades o cualidades que el producto debe tener. No se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes del mismo que hacen al producto atractivo, usable, rápido, confiable, etc. Normalmente son rasgos que se aplican al sistema en su totalidad y no a funciones específicas. [27]

A continuación se enuncian los requisitos no funcionales identificados:

- Interfaz: **RnF1** Integrarse al Sistema de Gestión Universitaria empleando las pautas de diseño y arquitectura de información definidas por la Dirección de Informatización y encontradas en el documento Arquitectura de Información.
- Funcionalidad: **RnF2** El módulo soportará la conexión simultánea de todos los posibles usuarios habituales, un promedio de 1000 y un máximo de 3000.
- Confiabilidad/ Fiabilidad: **RnF3** Garantizar la integridad y confidencialidad de la información mediante mecanismos de seguridad.
- Confiabilidad/ Tolerancia al error: **RnF4** Mostrar la información necesaria del error.
- Usabilidad/ Instructibilidad: **RnF5** Mostrar elementos de apoyo a la navegación.
- Usabilidad/ Operatividad: **RnF6** Mostrar funcionalidades de acuerdo a los privilegios.
- Disponibilidad: **RnF7** Disponibilidad del módulo las 24 horas del día.
- Hardware: **RnF8**
 - El Servidor de Bases de Datos y Servidor Web debe tener: Microprocesador de 8 núcleos, 4GB de memoria RAM, 250 GB de disco duro y una fuente de 800 W como mínimo.
 - El cliente debe tener: Microprocesador de 1GHz, 512 Mbs de memoria RAM, 1GB de disco duro, Mozilla Firefox 36.0 como mínimo.
- Soporte: **RnF9** El módulo cumplirá con las normas de codificación, conversiones para nomenclatura, bibliotecas de clase definidas para el Sistema de Gestión Universitaria.

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

- Soporte: **RnF10** El módulo contará con toda la documentación definida en el expediente de proyecto asociada a su proceso de desarrollo para las actividades de soporte.
- Restricciones de diseño: **RnF11** El módulo deberá ser desarrollado empleando las herramientas y la arquitectura de información definida para el Sistema de Gestión Universitaria.

2.3.3 Definición de roles

Los actores del negocio son aquellas personas que interactúan con el negocio para beneficiarse de sus resultados. Ellos se agrupan en diferentes roles los cuáles son descritos a continuación:

- **Administrador de tecnología:** Tiene todos los privilegios del módulo.
- **Administrador de área:** Aprueba las solicitudes correspondientes a su área.
- **Encargado de impresión:** Exporta e imprime documento de aprobación del medio solicitado por el usuario.
- **Usuario:** Crea solicitud de medio tecnológico personal. Crea solicitud de marca y modelo. Cancela una solicitud activa propia.

En la tabla 5 se define el rol que desempeñan algunos de los actores del negocio.

Tabla 5: Definición de roles. (Elaboración propia)

Actores que intervienen en el negocio	Rol que desempeñan en el módulo
Estudiante	Usuario
Trabajador	Usuario
Trabajador (designado)	Encargado de impresión
Decano	Administrador de área
Director de Centro	Administrador de área
Miembro del Consejo Universitario de su área	Administrador de área
Director de Seguridad y Protección	Administrador de área
Director de Seguridad Informática	Administrador de tecnología

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

2.3.4 Historias de Usuario

Las Historias de Usuario (HU) es un artefacto generado por metodologías ágiles como Scrum y XP. Son utilizadas con el fin de especificar los requisitos del software. A través de un conjunto de tablas se describen brevemente las características que desea el cliente. Contienen la información suficiente para que los desarrolladores puedan producir una estimación razonable del esfuerzo necesario para su implementación. Su contenido está estructurado de forma tal que se puedan descomponer en tareas para su posterior desarrollo entre 2 y 4 semanas. [28]

En el marco de esta investigación fueron definidos los siguientes parámetros a tener en cuenta en las HU:

- **Número:** número asignado a la HU.
- **Nombre:** nombre de la HU.
- **Programador:** nombre del programador responsable de la HU.
- **Iteración asignada:** iteración a la que pertenece la HU en el plan de iteraciones.
- **Prioridad:** nivel de prioridad de la HU señalado por el cliente (Alta, Media, Baja).
- **Puntos estimados:** estimación que hace el equipo de desarrollo del tiempo necesario para implementar la HU. El 1 equivale a una semana ideal de trabajo – 5 días hábiles por 8 horas diarias. Por lo que 0.5 equivale a 2 días y medio, para un total de 20 horas.
- **Puntos reales:** el tiempo real en el que se realizó la HU.
- **Descripción:** breve descripción de la HU.
- **Precondiciones:** condiciones que deben ocurrir para que se desarrolle la HU.
- **Flujo básico:** camino esperado que debe seguir el usuario durante el desarrollo de la HU.
- **Flujo alternativo:** caminos alternativos que puede seguir el usuario durante el desarrollo de la HU.
- **Pos-condiciones:** resultado obtenido del desarrollo de la HU.
- **Validaciones:** caracterización de los campos a validar durante el desarrollo de la HU.
- **Conceptos:** descripción de conceptos asociados a la HU.
- **Botones e hipervínculos:** botones e hipervínculos presentes en el prototipo de interfaz de la HU.
- **Requisitos especiales:** requisitos que escapan de la capacidad de respuesta de la HU.
- **Observaciones:** aspectos importantes de interés para el cliente.
- **Prototipo de interfaz:** prototipo visual del diseño esperado.

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

A continuación se presenta la HU_19 de un total de 34 HU realizadas, las cuáles se encuentran registradas en el anexo 2.

Tabla 6: HU_19 Crear solicitud de uso de medio tecnológico personal. (Elaboración propia)

Historia de Usuario	
Número: HU_19	Nombre: Crear solicitud de uso de medio tecnológico personal
Programador: Guillermo Armada González	Iteración asignada: 1
Prioridad: Alta	Puntos estimados: 1
Complejidad: Alta	Puntos reales: 1
<p>Descripción: Permite crear una nueva solicitud de uso de medio tecnológico personal.</p> <p>Precondiciones: El usuario debe estar autenticado en el sistema con el rol Usuario.</p> <p>Flujo básico Crear solicitud de uso de medio tecnológico personal.</p> <ol style="list-style-type: none"> 1. Seleccionar del módulo Medios Tecnológicos del subsistema Tecnologías, la opción Mis solicitudes del menú de funcionalidades Medios Tecnológicos. 2. Se muestra una interfaz con el listado de todas las solicitudes existentes en el sistema realizadas por el usuario, y las opciones de Crear, Modificar, Ver detalles. 3. Seleccionar la opción Crear en el área de íconos flotantes. 4. Se muestra una interfaz con una descripción de los términos de uso y la opción Listar. 5. Marcar la opción Estoy de acuerdo. 6. Seleccionar la opción Siguiente. 7. Se muestra una interfaz con campos a llenar para crear una solicitud de uso de medio tecnológico personal y la opción Listar. 8. Llenar campos. 9. Seleccionar la opción Aceptar. <p>Pos-condiciones: El usuario deberá llenar los campos marcados como obligatorios para poder crear la solicitud de uso de medio tecnológico personal.</p>	

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

Flujos alternativos

Flujo alternativo 4.a y 7.a

- 1- En caso de que el usuario seleccione la acción de Listar en el área de íconos flotantes el sistema mostrará una interfaz con el listado con todas las solicitudes existentes en el sistema realizadas por el usuario.

Pos-condiciones: N/A

Flujo alternativo 9.a

- 1- En caso de seleccionar la opción Anterior se muestra la interfaz anterior. Flujo básico 4.

Flujo alternativo 6.a y 9.a

- 1- En caso de cancelar la acción se muestra un mensaje de advertencia. “¿Está seguro de cancelar la acción?”.
- 2- Se selecciona la opción Aceptar o Cancelar.

Pos-condiciones: N/A

Validaciones:

Tipo de medio tecnológico:

- Campo de selección
- Campo obligatorio

Marca:

- Campo obligatorio
- Campo de texto
- Cantidad de caracteres por palabra es 30
- Cantidad de caracteres máxima es 250
- Cantidad de caracteres mínima es 2

Serie:

- Campo obligatorio
- Campo de texto

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

- Cantidad de caracteres por palabra es 30
- Cantidad de caracteres máxima es 250
- Cantidad de caracteres mínima es 2

MAC Wi-Fi

- Campo obligatorio
- Campo de texto
- Solo admite grupos de dos números hexadecimales separados por dos puntos o guion
- Cantidad de caracteres es 17

MAC Ethernet

- Campo obligatorio
- Campo de texto
- Solo admite grupos de dos números hexadecimales separados por dos puntos o guion
- Cantidad de caracteres es 17

Inventario:

- Campo de texto
 - Cantidad de caracteres por palabra es 30
 - Cantidad de caracteres máxima es 250
 - Cantidad de caracteres mínima es 2
- 1- En caso que el elemento ya existe se muestra un mensaje de error "El elemento ya existe".
 - 2- En caso que el elemento se cree correctamente se muestra un mensaje "El elemento ha sido creado satisfactoriamente".
 - 3- En caso de que introduzca datos incorrectos en los campos de MAC Wi-Fi y Ethernet se muestra un mensaje de error "Entre 6 grupos de dos números hexadecimales separados por dos puntos o guion".
 - 4- En caso de que se deje un campo de los obligatorios vacío se muestra un mensaje de error de color rojo "Campo obligatorio" en el campo que debe ser llenado obligatorio.
 - 5- Si se introduce más caracteres de los permitidos para una palabra el sistema muestra el mensaje "Ha excedido el número de caracteres permitidos para una palabra".
 - 6- Si se introduce más caracteres de los permitidos para un campo el sistema no permite la entrada de más caracteres.

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

Conceptos

- Tipo de medio tecnológico: Define el tipo de medio tecnológico al que se va a realizar una solicitud.
- Marca: Define la marca de un tipo de medio tecnológico.
- Modelo: Define el modelo del medio.
- Serie: Define el número de serie único que poseen todos los medios.
- MAC Wi-Fi: Define la dirección de control de acceso por IEE 802.11 del medio
- MAC Ethernet: Define la dirección de control de acceso por IEE 802.3 del medio.
- Inventario: Define el número de inventario que posee el medio en caso de ser un medio básico de la Universidad.

Botones e hipervínculos

Ícono Crear: Muestra la interfaz que permite crear una nueva solicitud.

Ícono Modificar: Se muestra para todos los elementos del listado de solicitudes. Muestra la interfaz que permite modificar la solicitud seleccionada.

Ícono Ver detalles: Se muestra para todos los elementos del listado de solicitudes. Muestra la interfaz que permite ver los detalles de la solicitud seleccionada.

Botón Aceptar: Permite subir la solicitud para ser creada.

Botón Cancelar: Cancela la acción de crear una solicitud.

Requisitos especiales: En caso del módulo no contar con la marca requerida, el usuario tiene que solicitar su incorporación a la base de datos.

Observaciones: Para poder crear una solicitud el usuario debe primeramente aceptar los términos de uso. Los campos Tipo de medio tecnológico, Marca, Modelo, Serie y al menos una de las MAC son obligatorios. Luego de completar los campos obligatorios se da aceptar y queda creada la solicitud.

Prototipo de interfaz:

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas



Crear solicitud de uso de medio tecnológico personal

Terminos de uso ▶ Solicitud

Terminos

Los datos expuestos se ajustan a la realidad y registrarán mi comportamiento como usuario de la Red Universitaria.

He leído y entendido las Normas de Uso (al dorso de éste formulario) y por tanto acataré todo lo expresado en dicho documento.

Es de mi responsabilidad la seguridad del equipo o dispositivo en cuestión, exonerando a la Universidad de todo compromiso en caso de pérdida o extravío.

Cumpliré con las regulaciones y normas de seguridad informáticas vigentes.

Presentaré el equipo o dispositivo personal ante una auditoría cuando se solicite por las autoridades competentes.

Estoy de acuerdo

Siguiete Cancelar

Crear solicitud

Términos ▶ Solicitud

Nota: Si la marca del medio que usted posee no está registrada en la base de datos debe comunicarlo a la Dirección de Informatización para que sea agregada.

Tipo de medio tecnológico:* <input type="text" value="-Seleccione-"/>	Marca:* <input type="text" value="-Seleccione-"/>	MAC:*
Modelo:* <input type="text"/>	Número de serie:* <input type="text"/>	Cableada: <input type="text"/>
<input checked="" type="checkbox"/> Medio básico:	No. inventario: <input type="text"/>	Wifi: <input type="text"/>

Anterior Aceptar Cancelar

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

2.3.5 Estimación del esfuerzo por grupo de Historias de Usuario

Se acordó con el cliente la conveniencia de desarrollar en una misma iteración las HU que mantuvieran una estrecha relación con la funcionalidad que representan. En tal sentido quedan agrupadas las HU como se muestra en la tabla 7, obteniéndose de ello una nueva estimación del esfuerzo.

Tabla 7: Estimación del esfuerzo por grupo de Historias de Usuario. (Elaboración propia)

ID	Descripción	Historias de Usuario comprendidas	Esfuerzo estimado
G₁	Gestionar tipo de medio tecnológico	1,2,3,4	1
G₂	Modificar cantidad del tipo de medio tecnológico por roles universitarios	5	0.5
G₃	Gestionar marca	6,7,8,9	1
G₄	Gestionar estado de solicitud	10,11,12,13	1
G₅	Gestionar área de aprobación	14,15,16,17	1
G₆	Gestionar solicitud de uso de medio tecnológico	18,19,20,21	3
G₇	Cancelar solicitud de uso de medio tecnológico	22	0.5
G₈	Gestionar solicitudes por área	23,24,25	2
G₉	Deshabilitar solicitudes activas de personas de baja	26	0.5
G₁₀	Reporte de solicitud por medio tecnológico	27,28	1
G₁₁	Reporte de historial de solicitudes	29,30	1
G₁₂	Reporte de cantidad de medios	31,32	1
G₁₃	Cola de impresión	33,34	1
G₁₄	Solicitudes en el sistema	35,36	1

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

2.4 Plan de publicaciones

(Artefacto auxiliar de la metodología XP)

Una vez identificadas las Historias de Usuario (HU) y estimado el esfuerzo para cada una de ellas, el equipo de desarrollo determina junto con el cliente la planificación de la fase de codificación. Partiendo de la información recogida en cada una de las HU se determina el número de iteraciones que le llevará al grupo de trabajo realizar su labor. Al final de cada iteración se obtiene una versión parcial del producto que luego se muestra al cliente para verificar que lo desarrollado hasta el momento es lo que él desea.

2.4.1 Plan de iteraciones

(Artefacto auxiliar de la metodología XP)

Este artefacto es definido por la metodología XP y se adiciona a la documentación como aporte auxiliar para lograr una mayor organización del trabajo. Se planifican 4 iteraciones. En cada una de ellas fueron distribuidas las Historias de Usuario (HU) según el orden de prioridad establecido por el cliente. La tabla 8 muestra los resultados de lo antes señalado.

Tabla 8: Plan de Iteraciones. (Elaboración propia)

Iteración	Grupos de Historias de Usuario a implementar	Prioridad
1	G₆ Gestionar solicitud de uso de medio tecnológico	Alta
1	G₇ Cancelar solicitud de uso de medio tecnológico	Alta
2	G₈ Gestionar solicitudes por área	Alta
2	G₉ Deshabilitar solicitudes activas de personas de baja	Alta
2	G₁₃ Cola de impresión	Alta
3	G₁₀ Reporte de solicitud por medio tecnológico	Alta
3	G₁₁ Reporte de historial de solicitudes	Alta
3	G₁₂ Reporte de cantidad de medios	Alta
3	G₁₄ Solicitudes en el sistema	Alta
4	G₁ Gestionar tipo de medio tecnológico	Media

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

4	G₂ Modificar cantidad de tipo de medio tecnológico por roles universitarios	Alta
4	G₃ Gestionar marca	Media
4	G₄ Gestionar estado de solicitud	Baja
4	G₅ Gestionar área de aprobación	Media

2.4.2 Plan de duración de iteraciones

(Artefacto auxiliar de la metodología XP)

Este plan se encarga de mostrar las Historias de Usuario (HU) agrupadas y ordenadas según la iteración en la que le corresponde ser implementada. Además se estima la duración total de cada iteración, esto dará paso a la confección del plan de entregas parciales y final de la aplicación. La tabla 9 muestra los resultados obtenidos.

Tabla 9: Duración de las iteraciones. (Elaboración propia)

Iteración	Grupos de Historias de Usuario a implementar	Duración (semanas)
Iteración 1	G₆ Gestionar solicitud de uso de medio tecnológico G₇ Cancelar solicitud de uso de medio tecnológico	3
Iteración 2	G₈ Gestionar solicitudes por área G₉ Deshabilitar solicitudes activas de personas de baja G₁₃ Cola de impresión	3
Iteración 3	G₁₀ Reporte de solicitud por medio tecnológico G₁₁ Reporte de historial de solicitudes G₁₂ Reporte de cantidad de medios G₁₄ Solicitudes en el sistema	3
Iteración 4	G₁ Gestionar tipo de medio tecnológico G₂ Modificar cantidad de tipo de medio tecnológico por roles universitarios	3

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

G₃ Gestionar marca G₄ Gestionar estado de solicitud G₅ Gestionar área de aprobación	
---	--

2.4.3 Plan de Entregas

(Artefacto auxiliar de la metodología XP)

El plan de entregas comprende la fecha en la que se despacha con el cliente cada versión del producto. En ese momento ya deben estar implementadas las Historias de Usuario (HU) correspondientes a la iteración planificada. En esta etapa se puede constatar que se hayan cumplido los compromisos contraídos y revisar para un posible cambio o mejora.

El equipo de trabajo consideró factible separar las iteraciones con una semana intermedia, ello permite aplicar oportunamente las correcciones y cambios señalados por el cliente, obteniendo una validación de los requisitos. Teniendo en cuenta que se comienza la fase de codificación con fecha 9 de diciembre del 2018, queda conformado en la tabla 10 el Plan de Entregas.

Tabla 10: Plan de Entregas. (Elaboración propia)

Sistema: Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas.			
Iteración	Fecha de iteración	Fecha de entrega	Entrega
1	17/12/18 - 04/1/19	11/01/19	Versión 0.1
2	14/01/19 - 01/02/19	08/02/19	Versión 0.2
3	11/02/19 - 01/03/19	08/03/19	Versión 0.3
4	11/03/19 - 29/03/19	01/04/19	Versión 0.4

2.5 Patrones

Un patrón es un tipo de tema de sucesos u objetos recurrentes. Puede definirse como aquella serie de variables constantes, identificables dentro de un conjunto mayor de datos los cuáles se repiten de una manera predecible. Una plantilla o modelo puede usarse para generar objetos o partes de ellos, especialmente si los objetos que se crean tienen lo suficiente en común para que se infiera la estructura del patrón fundamental, en cuyo caso, se dice que los objetos exhiben un único patrón. En la ingeniería

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

del software, un patrón constituye el apoyo para la solución a los problemas más comunes que se presentan durante las diferentes etapas del ciclo de vida de un software. [29]

2.5.1 Patrón arquitectónico

Los patrones arquitectónicos se utilizan para expresar una estructura de organización base o esquema para un software. Proporcionando un conjunto de sub-sistemas predefinidos, especificando sus responsabilidades, reglas, directrices que determinan la organización, comunicación, interacción y relaciones entre ellos. [30] [31]

Modelo Vista Controlador

El patrón arquitectónico Modelo-Vista-Controlador (MVC) especifica que una aplicación consta de un modelo de datos, de información de presentación y de información de control. El patrón requiere que cada uno de estos elementos esté separado en distintos objetos [32]

El **modelo** (por ejemplo, la información de datos) contiene únicamente los datos puros de la aplicación; no contiene la lógica que describe cómo pueden presentarse los datos a un usuario.

La **vista** (por ejemplo, la información de presentación) presenta al usuario los datos del modelo. La vista sabe cómo acceder a los datos del modelo, pero no sabe el significado de estos datos ni lo que el usuario puede hacer para manipularlos.

Por último, el **controlador** (por ejemplo, la información de control) está entre la vista y el modelo. Escucha los sucesos desencadenados por la vista (u otro origen externo) y ejecuta la reacción apropiada a estos sucesos. En la mayoría de los casos, la reacción es llamar a un método del modelo. Puesto que la vista y el modelo están conectados a través de un mecanismo de notificación, el resultado de esta acción se reflejará automáticamente en la vista.

La figura 3 muestra el diagrama de la relación Modelo Vista Controlador.

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

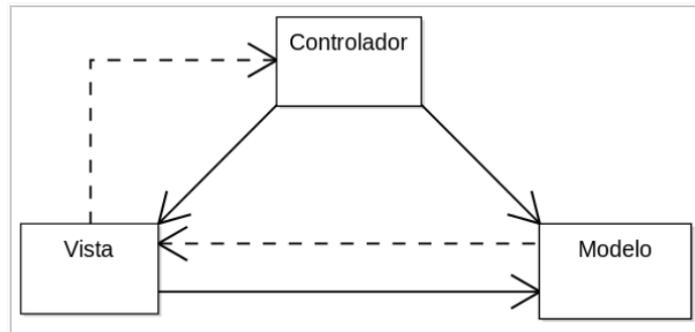


Figura 3: Diagrama Modelo Vista Controlador

2.5.2 Patrones de diseño

Los patrones de diseño son soluciones a problemas típicos y recurrentes que se pueden encontrar a la hora de desarrollar una aplicación. Aunque la aplicación sea única, tendrá partes comunes con otras aplicaciones: acceso a datos, creación de objetos, operaciones entre sistemas, entre otras. Incluso algunas funcionalidades y estructuras similares dentro del propio sistema.

Establecen un lenguaje común para programar. Identifica clases, instancias, roles, colaboraciones y distribución de responsabilidades. Esto permite un lenguaje de programación a alto nivel. Es la forma práctica de describir aspectos de la organización de un programa. [31]

Patrones de Software de Asignación de Responsabilidades Generales (GRASP por sus siglas en inglés) entre los cuales se encuentran y son utilizados:

Experto: El GRASP de experto en información es el principio básico de asignación de responsabilidades. Determina cuál es la clase que debe asumir una responsabilidad a partir de la información que posee. O sea, la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. De este modo se obtiene un diseño con mayor cohesión y así la información se mantiene encapsulada (disminución del acoplamiento).

Este patrón es evidenciado en la definición de las clases de acuerdo a las funcionalidades que deben realizar a partir de la información manejada dentro del componente, como por ejemplo las clases controladoras. La figura 4 muestra el uso del patrón Experto.

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

```
public function __construct()
{
    $this->_ci = & get_instance();
    $this->_ci->load->model('tb_nmarca_md1');
}

public function registrarMarca($datosMarca)
{
    $datosFinales = array();
    $datosFinales['nombre_marca'] = $datosMarca['nombre'];
    $datosFinales['descripcion'] = isset($datosMarca['descripcion']) ? $datosMarca['descripcion'] : '';
    $datosFinales['activo'] = empty($datosMarca['activo']) ? 'f' : 't';
    return $this->_ci->tb_nmarca_md1->registrar($datosFinales);
}
```

Figura 4: Patrón Experto

Creador: Ayuda a identificar quién debe ser el responsable de la creación (o instanciación) de nuevos objetos o clases.

Se identifica al asignar a la clase B la responsabilidad de crear una instancia de clase A si se cumple uno o más de los casos siguientes:

- B agrega objetos de A.
- B contiene objetos de A.
- B registra instancias de objetos de A.
- B utiliza más estrechamente objetos de A.
- B tiene los datos de inicialización que se pasarán a un objeto de A cuando sea creado (por tanto, B es un Experto con respecto a la creación de A).
- B es un creador de los objetos A.

Una de las consecuencias de usar este patrón es la visibilidad entre la clase creada y la clase creador. Una ventaja es el bajo acoplamiento, lo cual supone facilidad de mantenimiento y reutilización. La creación de instancias es una de las actividades más comunes en un sistema orientado a objetos. En consecuencia, es útil contar con un principio general para la asignación de las responsabilidades de creación. Si se asignan bien, el diseño puede soportar un bajo acoplamiento, mayor claridad, encapsulación y reutilización. La figura 5 muestra el uso del patrón Creador.

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

```
public function __construct()
{
    $this->_ci = & get_instance();
    $this->_ci->load->model('tb_nmarca_mdl');
}

public function registrarMarca($datosMarca)
{
    $datosFinales = array();
    $datosFinales['nombre_marca'] = $datosMarca['nombre'];
    $datosFinales['descripcion'] = isset($datosMarca['descripcion']) ? $datosMarca['descripcion'] : '';
    $datosFinales['activo'] = empty($datosMarca['activo']) ? 'f' : 't';
    return $this->_ci->tb_nmarca_mdl->registrar($datosFinales);
}
```

Figura 5: Patrón Creador

Controlador: Sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado. Sugiere que la lógica de negocios debe estar separada de la capa de presentación, esto permite aumentar la reutilización de código y a la vez tener un mayor control. Se recomienda dividir los eventos del sistema en un mayor número de controladores para poder aumentar la cohesión y disminuir el acoplamiento.

Este patrón es evidenciado en la especificación de la clase controladora “area_aprobacion.php”. El método registrar se encarga de cargar previamente los datos que serán mostrados en la vista “crear_area_aprobacion_view.php” una vez que el mismo la llame. La figura 6 muestra el uso del patrón Controlador.

```
public function registrar()
{
    $filtro['filtro']['activo'] = "t";
    $areas = $this->ioc->estructura_composicion->obtenerEstructurasActivas(TRUE);
    $this->template->set_data('areas', $areas);
    echo $this->template->render('area_aprobacion/crear_area_aprobacion_view');
}
```

Figura 6: Patrón Controlador

Alta Cohesión: Indica que la información que almacena una clase debe ser coherente y debe estar (en la medida de lo posible) relacionada con la clase.

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

Este patrón es evidenciado en la estrecha relación que guardan los nombre de las funcionalidades y variables. A continuación se muestra el método modificar que carga los datos de un tipo de medio específico y levanta la vista para que el usuario pueda modificar el mismo. La figura 7 muestra el uso del patrón Alta-Cohesión.

```
public function modificar()
{
    $id = $this->input->all_post('true');
    $tipoMedio = $this->tipo_medio_lib->obtenerTipoMedioDadoId($id['id']);
    $this->template->set_data('tipo_medio', $tipoMedio);
    echo $this->template->render('tipo_medio/modificar_tipo_medio_view');
}
```

Figura 7: Patrón Alta-Cohesión

Bajo Acoplamiento: Medida de la fuerza con que una clase está conectada a otras clases. Propone tener las clases lo menos ligadas entre sí, de tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases. Potencia la reutilización de código y disminuye la dependencia.

Este patrón es evidenciado claramente en el método “obtenerSolicitudesPorPagina (...)” de la clase entidad “tb_dsolicitud_medio_tecnologico_mdl.php”. Para obtener las solicitudes es necesario acceder a diferentes tablas. Aunque cada tabla posee una clase entidad capaz de resolver lo buscado es preferible desde la consulta local generar dichos datos sin depender de una funcionalidad externa. La figura 8 muestra el uso del patrón Bajo-Acoplamiento.

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

```
sconsulta = "SELECT
    tb_dsolicitud_medio_tecnologico.id_propietario,
    tb_dsolicitud_medio_tecnologico.id_solicitud_medio_tecnologico,
    tb_dsolicitud_medio_tecnologico.id_medio_tecnologico,
    tb_dsolicitud_medio_tecnologico.id_estado_solicitud,
    tb_ntipo_medio.nombre_tipo_medio,
    tb_nmarca.nombre_marca,
    tb_dmedio_tecnologico.modelo,
    tb_nestado_solicitud.nombre_estado_solicitud
FROM
    sq_medios_tecnologicos.tb_dsolicitud_medio_tecnologico,
    sq_medios_tecnologicos.tb_dmedio_tecnologico,
    sq_medios_tecnologicos.tb_ntipo_medio,
    sq_medios_tecnologicos.tb_nmarca,
    sq_medios_tecnologicos.tb_nestado_solicitud
WHERE
```

Figura 8: Patrón Bajo acoplamiento

2.5.3 Patrones GoF

Los patrones GoF utilizados en el desarrollo de la aplicación son:

Patrón de comportamiento: Comprenden la asignación de responsabilidades entre objetos y algoritmos. Estos no solo conciernen a los objetos y las clases sino también la comunicación entre estas, además caracterizan flujos de control complejos que son difíciles de seguir en tiempo de ejecución. [31] [33]

Fachada (Facade): Permite dar soporte a reglas del negocio conectables. Es decir, en puntos predecibles de los escenarios, como cuando tiene lugar una acción de registrar o modificar un elemento. Las reglas permiten al sistema desde una misma interfaz adaptar ligeramente su comportamiento ante diversos escenarios o incluso cancelar una acción.

Este patrón se evidencia en el método “registrarSolicitud()” de la clase controladora “solicitud_medio_tecnologico”. Dentro del método se hace un llamado a la función “chequearMedioTecnologico(\$datos, true)” que verifica que el medio para el cual se está realizando la solicitud no coincida con un medio registrado en el sistema para el cual haya alguna solicitud activa. La figura 9 muestra el uso del patrón Fachada.

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

```
public function registrarSolicitud()
{
    $token = $this->session->userdata('token');
    $selementosToken = $this->ioc->seguridad->obtenerElementosDadoToken($token);

    if (!isset($selementosToken['id_persona']))
        throw new Exception_Error('SYS023');

    $datos = $this->input->all_post(true);
    $datos['id_persona'] = $selementosToken['id_persona'];

    if ($this->solicitud_medio_tecnologico_lib->chequearMedioTecnologico($datos, true))
        throw new Exception_Error('SYS221');

    if ($this->solicitud_medio_tecnologico_lib->registrarSolicitud($datos) != false)
        $this->message('SYS0119');
    else
        throw new Exception_Error('SYS004');
}
```

Figura 9: Patrón Fachada

Mediador (Mediator): Este patrón se evidencia en las librerías, las cuales son mediadoras entre las clases controladoras y los modelos o acceso a datos.

A continuación se muestra la figura 10 referente al método “registrarMarca()” de la clase controladora “marca.php” el cual realiza peticiones a los métodos “chequearNombreRepetido()” y “registrarMarca()” de la clase “marca_lib.php”. Seguido de la figura del método “registrarMarca()” que se encarga de establecer la comunicación entre la clase controladora y la librería, además normaliza los datos con los que será añadida una nueva marca. De tal manera se hace presente el patrón mediador en la clase “marca_lib”.

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

```
public function registrarMarca()
{
    $verificar = array('nombre');
    if ($this->input->is_post_back($verificar)) {
        $marca = $this->input->all_post(true);

        if ($this->marca_lib->chequearNombreRepetido($marca)) {
            throw new Exception_Error('SYS005');
        }
        if ($this->marca_lib->registrarMarca($marca) != false) {
            $this->message('SYS001');
        } else {
            throw new Exception_Error('SYS004');
        }
    } else {
        throw new Exception_Error('SYS007');
    }
}
```

Figura 10a: Patrón Mediator

```
public function registrarMarca($datosMarca)
{
    $datosFinales = array();
    $datosFinales['nombre_marca'] = $datosMarca['nombre'];
    $datosFinales['descripcion'] = isset($datosMarca['descripcion']) ? $datosMarca['descripcion'] : '';
    $datosFinales['activo'] = empty($datosMarca['activo']) ? 'f' : 't';
    return $this->_ci->tb_nmarca_md1->registrar($datosFinales);
}
```

Figura 10b: Patrón Mediator

Patrones creacionales: Solucionan problemas de creación de instancias. Ayudan a encapsular y abstraer dicha creación:

Instancia única (Singleton): Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Restringe la instanciación de una clase o valor de un tipo a un solo objeto.

En la figura 11 se muestra la clase librería “marca_lib.php” la cual tiene en su constructor una instancia de CodeIgniter “_ci”. De esta manera no hay necesidad de instanciar a CodeIgniter en cualquier otra parte de la clase. Su intención consiste en garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella.

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

```
public function __construct()
{
    $this->_ci = & get_instance();
    $this->_ci->load->model('tb_nmarca_md1');
}

public function obtenerMarcasPorPagina($inicio = null, $limite = null, $elementoOrdenar = '', $dirOrdenar = 'asc', $params = array())
{
    $marca = $this->_ci->tb_nmarca_md1->obtenerMarcaPorPagina($inicio, $limite, $elementoOrdenar, $dirOrdenar, $params);
    return $marca;
}
```

Figura 11: Patrón Instancia Única

2.6 Modelo de datos

El diseño de una base de datos consiste en definir la estructura de los datos que debe tener un sistema de información determinado. Para ello se suelen seguir por regla general unas fases en el proceso de diseño, definiendo para ello el modelo conceptual, el lógico y el físico.

En el modelo relacional las dos capas de diseño conceptual y lógico, se parecen mucho. Generalmente se implementan mediante diagramas de Entidad/Relación (modelo conceptual) y tablas y relaciones entre éstas (modelo lógico). Este es el modelo utilizado por los sistemas gestores de datos más habituales (SQL Server, Oracle, MySQL, etc). [34]

A continuación la figura 12 muestra el diagrama Entidad/Relación que modela la base de datos de la aplicación.

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

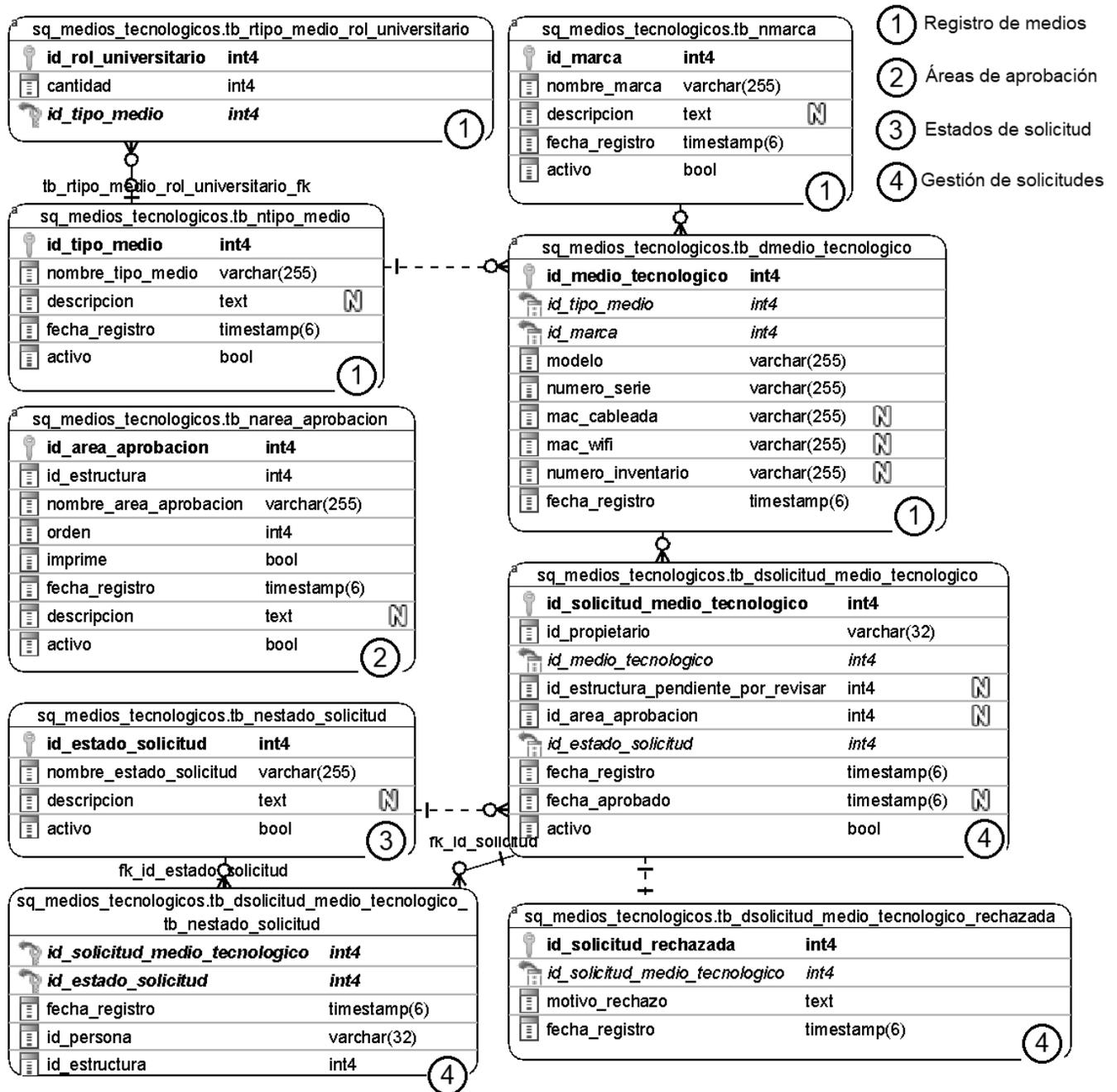


Figura 12: Diagrama de Entidad/Relación

Se identificaron un total de 9 tablas, de ellas: 4 nomencladores, 4 de datos y 1 de relación. Como se observa se muestran agrupadas por valores numéricos según el concepto que trabajan. Con el número

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

uno (1) se agrupa las tablas donde se maneja el registro de medios tecnológicos. Con el número dos (2) la tabla donde se maneja las áreas de aprobación que valoran cada solicitud. Con el número tres (3) la tabla donde se manejan los estados que adquiere la solicitud en cada instante de su flujo de aprobación. Finalmente, con el número cuatro (4) las tablas donde se maneja la gestión de solicitudes

2.7 Conclusiones del capítulo

Después de realizado el análisis y diseño de la propuesta de solución y haber generado los diferentes artefactos que dispone la metodología Scrum, se arriba a las siguientes conclusiones:

- La conformación de Historias de Usuario con el cliente permitió recoger 36 requisitos funcionales y 11 requisitos no funcionales.
- La obtención de requisitos facilitó una correcta modelación del problema.
- La estimación aproximada del tiempo de implementación de cada requisito y la planificación de las iteraciones permitió establecer un tiempo de 3 semanas para cada iteración con una semana extra para la revisión con el cliente.
- Se emplea el estilo arquitectónico Modelo Vista Controlador y se describen 5 patrones de diseño utilizados durante la codificación.

CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN DE LA PROPUESTA SOLUCIÓN

En el presente capítulo se definen los estándares de codificación con los que debe cumplir el equipo de desarrollo durante la implementación, así como los métodos y técnicas para la realización de pruebas. El proceso de pruebas está dirigido a componentes del software, con el objetivo de medir el grado en que se cumplen los requisitos exigidos por el cliente y si posee la calidad requerida.

La etapa de implementación del software es el proceso de convertir una especificación del sistema en un sistema ejecutable. Esta fase comprende la materialización, en forma de código, de todos los artefactos, descripciones y arquitectura propuestos en la etapa de análisis y diseño; con el objetivo de conformar el producto final requerido por el cliente. [31]

3.1 Estándares de codificación

Los estándares de codificación son reglas de codificación que permiten tener una programación homogénea, comprendiendo todos los aspectos de la generalización del código, pues la aplicación debe de estar implementada como si un único programador escribiera el código de una sola vez. La usabilidad de estos permite conservar el código fuente entendible y fácil de mantener.

Para el desarrollo del componente se utilizaron los estándares de codificación establecidos por la DIN con el propósito de estandarizar las nomenclaturas en la implementación del mismo y obtener un producto estable.

3.1.1 Indentación, llaves de apertura y cierre, y tamaño de líneas

Usar una indentación³ sin tabulaciones, con un equivalente a 4 espacios. El uso de las llaves “{}” será en una nueva línea. La longitud de las líneas de código es aproximadamente de 75-80 caracteres. Para mantener la legibilidad del código.

La indentación se ve reflejada en la función `index()` de la controladora “`area_aprobacion.php`”, tal y como se muestra en la figura 13:

³ **Indentación:** es un anglicismo de la palabra inglesa *indentation*, de uso común en la informática. No es un término reconocido por la Real Academia Española. En los lenguajes de programación es un tipo de notación secundaria utilizado para mejorar la legibilidad del código fuente por parte de los programadores.

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

```
public function index()
{
    echo $this->template->render('area_aprobacion/listar_area_aprobacion_view');
}
```

Figura 13: Indentación

3.1.2 Convención de nomenclatura

Variables: Se rigen por la nomenclatura camelCase⁴. Siempre comienzan con minúscula (lowerCamelCase) y en caso de nombres compuestos la primera letra de cada palabra comienza con mayúscula excepto la primera. La figura 14 muestra en el método registrarMarca() como se emplean las variables:

```
public function registrarMarca($datosMarca)
{
    $datosFinales = array();
    $datosFinales['nombre_marca'] = $datosMarca['nombre'];
    $datosFinales['descripcion'] = isset($datosMarca['descripcion']) ? $datosMarca['descripcion'] : '';
    $datosFinales['activo'] = empty($datosMarca['activo']) ? 'f' : 't';
    return $this->_ci->tb_nmarca_md1->registrar($datosFinales);
}
```

Figura 14: Definición de variables

Clases: Siempre comienzan con mayúscula, en caso de nombre compuesto las palabras se separan con guion bajo “_” y el resto en minúscula. Las figuras 15 y 16 muestran un ejemplo de la nomenclatura de una clase simple y una compuesta.

<?php

```
/**
 *
 * @author Guillermo Armada Gonzalez
 * @property marca
 */
class Marca extends MY_Controller
{
```

Figura 15: Definición de clase con nombre simple

⁴ **CamelCase:** Es un estilo de escritura que se aplica a frases o palabras compuestas. El nombre se debe a que las mayúsculas a lo largo de una palabra en camelCase se asemejan a las jorobas de un camello. El término case se traduce como "caja tipográfica", que a su vez implica si una letra es mayúscula o minúscula.

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

```
<?php

/**
 *
 * @author Guillermo Armada Gonzalez
 * @property area_aprobacion
 */
class Area_aprobacion extends MY_Controller
{
```

Figura 16: Definición de clase con nombre compuesto

Funciones: Se rigen por la nomenclatura camelCase. Siempre comienzan con minúscula (lowerCamelCase) y en caso de nombres compuestos la primera letra de cada palabra comienza con mayúscula excepto la primera. Los parámetros son separados por espacio luego de la coma que los separa. La figura 17 muestra un ejemplo de la función “registrarMarca()”, donde se emplea la nomenclatura para funciones:

```
public function registrarMarca($datosMarca)
{
    $datosFinales = array();
    $datosFinales['nombre_marca'] = $datosMarca['nombre'];
    $datosFinales['descripcion'] = isset($datosMarca['descripcion']) ? $datosMarca['descripcion'] : '';
    $datosFinales['activo'] = empty($datosMarca['activo']) ? 'f' : 't';
    return $this->_ci->tb_nmarca_md1->registrar($datosFinales);
}
```

Figura 17: Definición de funciones

Ficheros: Todo siempre en minúscula y en caso de nombres compuestos se usa el guion bajo, seguido del sufijo definido para cada tipo de fichero.

Vistas: Intuitivo y relacionado con el formulario y/o vista que representa. Sufijo `__view`. Ejemplo: `detalles_marca_view.php`.

Modelos: Mismo nombre de la tabla de la base de datos para la cual se crea. Sufijo `__mdl`. Ejemplo: `tb_nmarca_md1.php`.

Librerías: Mismo nombre de la clase que representa. Sufijo `__lib`. Ejemplo: `marca_lib.php`.

Controladoras: Mismo nombre de la clase que representa. Ejemplo: `marca.php`.

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

3.1.3 Estructura de control

Las estructuras de control incluyen if, for, foreach, while, switch. Entre las estructuras de control y los paréntesis debe de existir un espacio. Se recomienda utilizar siempre llaves de apertura y cierre, incluso en situaciones en las que técnicamente son opcionales. Con esto se aumenta la legibilidad del código y se disminuye la probabilidad de errores lógicos

3.1.4 Documentación

Todos los archivos deben de tener la documentación asociada al mismo. Se debe de cumplir con el siguiente bloque al principio de cada clase, como se muestra en la figura 18 referente a la clase “Marca”:

```
/**
 *
 * @author Guillermo Armada Gonzalez
 * @property marca
 */
```

Figura 18: Empleo de Documentación

3.1.5 Espacios en blanco

Se deben usar dos líneas en blanco entre las diferentes secciones de un fichero de código fuente. Una línea en blanco entre métodos, variables locales de un método y la primera sentencia, antes de un bloque o un comentario de línea, y entre secciones lógicas dentro de un fichero.

3.1.6 Buenas prácticas

- Los valores constantes siempre se escriben con mayúscula, ejemplo: NULL, FALSE, TRUE.
- Se debe usar un salto de línea antes de las estructuras de control y definición de las funciones.
- Las funciones echo y print realizan la misma tarea. Sin embargo, echo es sensiblemente más rápida que print por lo cual es preferible usarla.
- Al trabajar con cadenas evitar el uso de comillas dobles. La razón es que PHP analiza el contenido de las comillas dobles en búsqueda de variables que deban ser interpretadas.
- Separar y reutilizar el código agrupando las funciones que serán utilizadas frecuentemente y evitando que sea muy extenso el código.
- Minimizar el número de variables globales. Operar con variables globales es más costoso en tiempo que trabajar con variables locales.

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

- No implementar todas las estructuras de datos como objetos. Los arreglos son más rápidos y consumen menos memoria que instanciar un objeto.
- Pre (Incrementar o Decrementar) variables es más rápido que Pos (Incrementar o Decrementar). La razón es que para Pos (Incrementar o Decrementar) PHP necesita crear una variable temporal en la que almacenar el valor a ser (Incrementado o Decrementado).
- Minimizar el número de consultas a la base de datos. Las consultas son temporalmente costosas.
- Utilizar `isset()` cuando queramos conocer si una cadena posee una cantidad de caracteres mayor a un número determinado. En tanto `strlen()` es una función conocida pero lenta en comparación con `isset()` que es una construcción del lenguaje. [35]

3.2 Validación de requisitos

Es importante asegurar la validez de los requisitos previamente a comenzar un desarrollo de software. Para ello debe de hacerse una comprobación de la correspondencia entre las descripciones iniciales y si el modelo es capaz de responder al planteamiento inicial.

La validación de requisitos examina las especificaciones para asegurar que todos los requisitos del sistema han sido establecidos sin ambigüedad, sin inconsistencias, sin omisiones, que los errores detectados hayan sido corregidos y que el resultado del trabajo se ajusta a los estándares establecidos para el proyecto y el producto. [36]

Esta fase permite mitigar el riesgo de implementar una mala especificación, con el costo que eso conlleva. Los parámetros a validar en los requisitos son:

Validez: No basta con preguntar a un usuario, todos los potenciales usuarios pueden tener puntos de vista distintos y necesitar otros requisitos.

Consistencia: No debe haber contradicciones entre unos requisitos y otros.

Complejidad: Deben estar todos los requisitos. Esto es imposible en un desarrollo iterativo, pero, al menos, deben estar disponibles todos los requisitos de la iteración en curso.

Realismo: Se pueden implementar con la tecnología actual.

Verificabilidad: Tiene que existir alguna forma de comprobar que cada requisito se cumple.

3.2.1 Técnicas de validación de requisitos

Existen varias técnicas bien definidas para la validación de requisitos, ellas son:

Revisiones de verificabilidad: Consiste en la lectura y corrección de la documentación o modelado de la definición de requisitos. Con ello solamente se puede validar la correcta interpretación de la información transmitida.

Auditorías: Consiste en un chequeo de los resultados.

Matrices de trazabilidad: Consiste en marcar los objetivos del sistema y chequearlos contra los requisitos del mismo. Es necesario ir viendo que objetivos cubre cada requisito para detectar inconsistencias u objetivos no cubiertos.

Prototipos: Muestran en imágenes la interfaz esperada de las funcionalidades descritas. [36]

3.2.2 Resultados de aplicar las técnicas de revisión:

- De las revisiones de verificabilidad se obtuvieron los siguientes resultados:
 - Se amplió la descripción de requisitos que se encontraban poco detallados.
 - Se detectaron y corrigieron errores ortográficos y de concordancia en la redacción de las historias de usuario.
- Se modelaron prototipos de interfaz que fueron incorporados a las historias de usuario oportunamente, excepto para el requisito RF_26 que no requiere interfaz de usuario.
- De la aplicación de matrices de trazabilidad se pudo verificar que se cumplen en un 100% los rasgos generales señalados en el epígrafe **Descripción de la propuesta solución**, (Ver epígrafe 2.2).
- Mediante la técnica de auditoría se fueron señalando al final de cada iteración los requisitos que fueron implementados correctamente. Esto permitió que se identificara fácilmente el trabajo pendiente que quedaba de la iteración anterior.

3.3 Estrategia de pruebas

Para evaluar la calidad del componente que se está desarrollando y verificar el cumplimiento de los objetivos trazados, se diseña una estrategia de prueba para aplicar a la solución desarrollada. [31]

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

Tabla 11: Estrategia de pruebas. (Elaboración propia)

| Prueba | Método | Técnica |
|--|-------------|-----------------------|
| Prueba de unidad | Caja blanca | Camino básico |
| Prueba de integración | Caja negra | Incremental |
| Prueba de validación | Caja negra | Partición equivalente |
| Prueba de sistema:
Prueba de Integridad de Datos y Bases de Datos
Prueba de Seguridad y Control de Acceso
Prueba de desempeño | Caja negra | Semiautomático |

3.3.1 Pruebas de unidad

Las pruebas de unidad enfocan el esfuerzo de verificación en la unidad más pequeña de diseño de software, el componente o módulo de software. Al utilizar la descripción de diseño a nivel de componente como guía, se prueban rutas de control importantes para descubrir errores dentro del límite del módulo. La complejidad relativa de las pruebas y los errores que esas pruebas descubren están limitados por el alcance restringido establecido para las pruebas unitarias. La prueba de unidad se centra en la lógica de procesamiento interno y las estructuras de datos dentro de los límites de un componente. Este tipo de prueba se puede realizar en paralelo para múltiples componentes.

Esta prueba se realiza de forma automática en el sistema, para ello el marco de trabajo GUUD posee la librería `unit_test` para la ejecución de pruebas estructurales. La librería permite determinar con certeza si un código específico produce el tipo de dato y resultado esperado. Para ejecutar la prueba se utiliza la línea de código: `$this->unit->run(código, resultado esperado, 'nombre de prueba');` donde “código” es el segmento de código que se desea probar, “resultado esperado” es lo que debe devolver la evaluación del código, que puede ser un tipo de dato o un valor literal y “nombre de la prueba” es el nombre que se le puede dar a la prueba. [31]

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

A continuación se muestra el caso de prueba del método chequearMedioTecnologico() de la funcionalidad registrarSolicitud().

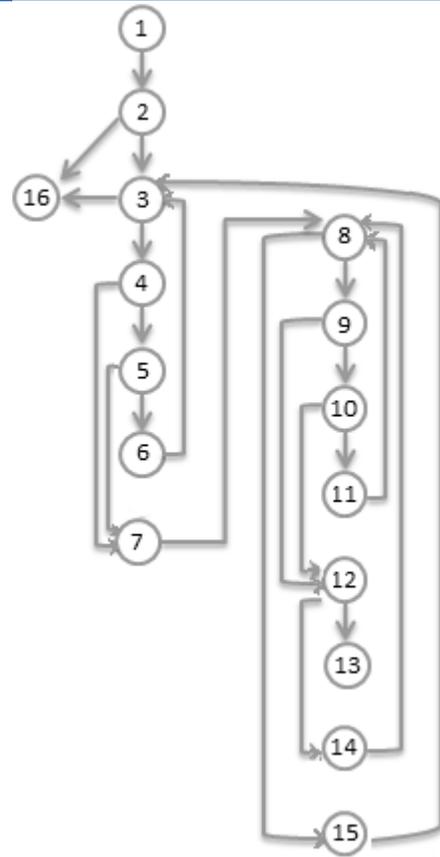
Tabla 12: Prueba estructural de caja blanca para la funcionalidad registrarSolicitud(). (Elaboración propia)

| Prueba estructural de caja blanca | Código de caso de prueba |
|--|----------------------------------|
| | |
| Probador | Guillermo Armada González |
| Tipo de dato esperado | Booleano |
| Código al que se aplica: | |
| <pre> public function chequearMedioTecnologico(\$datos, \$screar) { \$medios = \$this->_ci->tb_dmedio_tecnologico_mdل->obtenerMediosDadoAtributosEspecificos(\$datos); if (!empty(\$medios)) { foreach (\$medios as \$medio) { if (\$screar == 'false') if (\$medio->id_medio_tecnologico == \$datos['id_medio_tecnologico']) continue; \$solicitudes = \$this->_ci->tb_dsolicitud_medio_tecnologico_mdل->obtenerDadoAtributos (array('id_medio_tecnologico' => \$medio->id_medio_tecnologico)); foreach (\$solicitudes as \$solicitud) { if (\$screar == 'false') if (\$solicitud->id_solicitud_medio_tecnologico == \$datos['id_solicitud_medio_tecnologico']) continue; if (\$this->verificarEstadoSolicitud(\$solicitud->id_estado_solicitud)) return true; //existe el medio asignado a otra solicitud en proceso de aprobacion o aceptada } } } return false; } </pre> | |
| Complejidad ciclomática:
$V(G) = E - N + 2$
$V(G) = 22 - 16 + 2 = 8$
Caminos independientes:
<ol style="list-style-type: none"> 1. 1,2,16 2. 1,2,3,16 3. 1,2,3,4,5,6,3,16 4. 1,2,3,4,7,8,15,3,16 5. 1,2,3,4,7,8,9,12,13 6. 1,2,3,4,7,8,9,10,12,13 | Representación del grafo: |

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

7. 1,2,3,4,7,8,9,12,14,8,15,3,16

8. 1,2,3,4,7,8,9,10,11,8,15,3,16



Caso de prueba para los caminos básicos:

Tipo de dato esperado:

Booleano

Función de evaluación:

```
public function index()
{
    $datos = array();
    $datos['tipo_medio'] = '1';
    $datos['marca'] = '1';
    $datos['modelo'] = 'unit test';
    $datos['no_serie'] = 'prueba de integridad';
    $datos['id_medio_tecnologico'] = '50';
    $datos['id_solicitud_medio_tecnologico'] = '50';
    echo $this->unit->run($this->solicitud_medio_tecnologico_lib->
        chequearMedioTecnologico($datos, $datos['crear']), FALSE, 'Existe medio');
}
```

Resultados de la prueba:

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

| | |
|--------------------------|---|
| Test Name | Existe medio |
| Test Datatype | Boolean |
| Expected Datatype | Boolean |
| Result | Passed |
| File Name | /var/www/proyecto_sgu/base/application/tecnologia/medios_tecnologicos_b/controllers/solicitud_medio_tecnologico.php |
| Line Number | 29 |

| | |
|---------------------------------------|---------------|
| Evaluación del caso de prueba: | Satisfactoria |
|---------------------------------------|---------------|

Se llevaron a cabo dos iteraciones para detectar no conformidades, las cuales arrojaron los siguientes resultados como se muestra en la tabla 13.

Tabla 13: Iteración de las pruebas de caja blanca. (Elaboración propia)

| Iteración | Cantidad de no conformidades | Asociadas a |
|-----------|------------------------------|---|
| Primera | 17 | Errores de entradas y salidas incorrectas.
Errores de consultas en base de datos.
Errores de excepción con poca información o incorrecta. |
| Segunda | 0 | |

3.3.2 Pruebas de integración

Las pruebas de integración son una técnica sistemática para construir la arquitectura del software y, al mismo tiempo, realizar pruebas para descubrir errores asociados con la interconexión. El objetivo es tomar componentes probados por unidades y construir la estructura del programa que se diseñó.

A menudo hay una tendencia a intentar la integración no incremental; es decir, para construir el programa usando un enfoque de "Big Bang". Todos los componentes se combinan de antemano. Todo el programa se prueba como un todo. Y el caos suele resultar. Se encuentra un conjunto de errores. La corrección es difícil porque el aislamiento de las causas se complica por la vasta extensión de todo el programa. Una vez que se corrigen estos errores, aparecen otros nuevos y el proceso continúa en un bucle aparentemente interminable.

La integración incremental es la antítesis del enfoque del "Big Bang". El programa se construye y se prueba en pequeños incrementos, donde los errores son más fáciles de aislar y corregir. Es más probable que las interfaces se prueben completamente y se puede aplicar un enfoque de prueba sistemático. [31]

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

Se emplea el método de integración incremental por las facilidades que brinda al desarrollador de realizar pruebas al programa menos complejas y obteniendo mejores resultados. A continuación, se presentan dos estrategias del método seleccionado.

Estrategias de integración

Top-down (descendente): La prueba de integración descendente es un enfoque incremental para la construcción de la arquitectura del software. Los módulos se integran moviéndose hacia abajo a través de la jerarquía de control, comenzando con el módulo de control principal (programa principal). Los módulos subordinados al módulo de control principal se incorporan a la estructura ya sea en profundidad o primero en amplitud. El proceso de integración se realiza en una serie de cinco pasos:

1. El módulo de control principal se utiliza como controlador de prueba y los apéndices se sustituyen por todos los componentes directamente subordinados al módulo de control principal.
2. Dependiendo del enfoque de integración seleccionado (es decir, primero en profundidad o primero en amplitud), los apéndices subordinados se reemplazan uno a la vez con componentes reales.
3. Las pruebas se realizan a medida que se integra cada componente.
4. Al completar cada conjunto de pruebas, se reemplaza otro apéndice con el componente real.
5. Se pueden realizar pruebas de regresión para garantizar que no se hayan introducido nuevos errores.

El proceso continúa desde el paso 2 hasta que se construye toda la estructura del programa.

Bottom-up (ascendente): Las pruebas de integración ascendente, como su nombre lo indica, comienzan la construcción y las pruebas con módulos atómicos (es decir, componentes en los niveles más bajos en la estructura del programa). Debido a que los componentes se integran de abajo hacia arriba, la funcionalidad proporcionada por los componentes subordinados a un nivel dado siempre está disponible y se elimina la necesidad de apéndices. Una estrategia de integración de abajo hacia arriba se puede implementar con los siguientes pasos:

1. Los componentes de bajo nivel se combinan en grupos (a veces llamados construcciones) que realizan una sub-función de software específica.
2. Un controlador (un programa de control para pruebas) está escrito para coordinar el ingreso y salida de prueba.

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

3. Se prueba el conjunto o construcción.
4. Los controladores se eliminan y los grupos se combinan moviéndose hacia arriba en la estructura del programa.

En este caso se emplea un acercamiento práctico (a veces llamado Sandwich testing) que utiliza Top-down para los niveles superiores de la estructura del programa y Bottom-up para los niveles inferiores. Combinando lo mejor de ambas técnicas. [31]

Detalles de la prueba

La prueba se encuentra dividida en tres fases.

- 1- **Interacción de usuario:** En esta fase se revisan los datos de entrada y salida, el formato de presentación de los datos, el procesamiento de errores y la presentación de los mismos.
- 2- **Procesamiento de datos:** En esta fase se revisan la obtención de datos, manipulación del formato de los datos, determinación de condiciones de procesamiento, acciones requeridas como consecuencia de las condiciones encontradas.
- 3- **Comunicación:** En esta fase se revisan el traspaso de datos entre funciones y clases, la llamada a funciones, las rutas de comunicación entre clases y funciones.

En cada prueba se aplicaron los siguientes criterios:

- **Integridad de interfaz:** Son comprobadas las interfaces internas y externas de cada subsistema o funcionalidad que es integrada al módulo.
- **Validación funcional:** Se aplican pruebas diseñadas para descubrir errores funcionales.
- **Contenido de información:** Se aplican pruebas diseñadas para descubrir errores asociados con estructuras de datos locales o globales.
- **Desempeño:** Se aplican pruebas durante el diseño del software para verificar el desempeño del programa se encuentre entre los límites establecidos.

Ejemplo de prueba de integración

Se aplica la prueba de integración al componente “Configurar medio tecnológico”, obteniéndose los resultados que a continuación se muestran en cada fase.

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

Tabla 14: Caso de prueba de integración del componente: Configurar medio tecnológico. (Elaboración propia)

| Caso de prueba de integración del componente: Configurar medio tecnológico | |
|---|---|
| Sistema/componente al que se integra | Tipo de medio. |
| Condiciones de ejecución | Existe al menos un tipo de medio el cual va a ser configurado. |
| Descripción de la prueba | Comprobar que el componente “Configurar medio tecnológico” proporcione la información referente al tipo de medio tecnológico seleccionado, los roles universitarios existentes en el sistema y la configuración de la cantidad establecida en la relación del tipo de medio con cada rol. Validar la correcta introducción de las cantidades en el campo de cantidad. Además permite guardar los cambios realizados y notificar al usuario de cada acción ocurrida. |
| Pasos de ejecución | El probador interactúa con el componente.
Se modifica el campo de cantidad con valores válidos y no válidos.
Se está atento en todo momento a la emisión o no de alertas del componente según corresponda en cada instante.
Comprobar en base de datos que los valores han sido modificados satisfactoriamente. |
| Resultados esperados | El componente “Configurar medio tecnológico” brinda toda la información solicitada. Posee las validaciones pertinentes al campo cantidad y guarda satisfactoriamente los cambios realizados. |
| Evaluación | Prueba satisfactoria |

3.3.3 Pruebas de validación

La prueba de validación comienza en la culminación de la prueba de integración, cuando se han ejercitado los componentes individuales, el software se ensambla completamente como un paquete y los errores de interfaz se han detectado y corregido. En la validación o nivel del sistema, la distinción entre diferentes categorías de software desaparece. Las pruebas se centran en las acciones visibles por el usuario y en la salida del sistema reconocible por el usuario.

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

La validación se puede definir de muchas maneras, pero una definición simple (aunque severa) es que la validación tiene éxito cuando el software funciona de una manera que el cliente puede esperar razonablemente. [31]

Si se ha desarrollado una especificación de requisitos de software, esta describe todos los atributos del software visibles al usuario y contiene una sección de Criterios de validación que forma la base para un enfoque de prueba de validación. Dicha sección fue recogida en las Historias de Usuario, donde se describe detalladamente los campos que muestran cada interfaz y cómo deben ser validados.

Criterio de prueba de validación

La validación del software se logra a través de una serie de pruebas que demuestran la conformidad con los requisitos. Un plan de pruebas describe las clases de pruebas que se realizarán y un procedimiento de pruebas define casos de prueba específicos que están diseñados para garantizar que se cumplan todos los requisitos funcionales, se logren todas las características de comportamiento, todo el contenido sea preciso y se presente correctamente, la documentación es correcta, y se cumplen la usabilidad y otros requisitos (por ejemplo, transportabilidad, compatibilidad, recuperación de errores, mantenimiento). Si se descubre una desviación de la especificación, se crea una lista de deficiencias.

Particiones de equivalencia

La partición de equivalencia es un método de prueba de caja negra que divide el dominio de entrada de un programa en clases de datos a partir de los cuales se pueden derivar los casos de prueba. Un caso de prueba ideal por sí solo descubre una clase de errores (por ejemplo, el procesamiento incorrecto de todos los datos de caracteres) que de otro modo podrían requerir que se ejecuten muchos casos de prueba antes de que se observe el error general.

La técnica de particiones de equivalencia consiste en ejecutar el flujo básico de las funcionalidades utilizando datos válidos e inválidos. De su aplicación se generaron los artefactos “Diseño de caso de prueba”. Por cada requisito funcional del sistema se generó un documento donde se recogen todos los datos necesarios para probar la interfaz tal y como se muestra a continuación.

Diseño de caso de prueba

Seguidamente se describe el caso de prueba Crear Marca.

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

Tabla 15: Diseño de caso de prueba para validación Crear Marca. (Elaboración propia)

| Escenario | Descripción | Marca | Descripción | Habilitado | Respuesta del sistema | Flujo central |
|---|--|------------|------------------------|--------------|---|---|
| EC 1.1
Insertar datos correctos | Mediante este escenario se inserta en el sistema una marca. | V
Acer | V
Breve descripción | V
Marcado | El sistema crea la nueva marca y muestra el mensaje: "El elemento ha sido creado satisfactoriamente." | El usuario una vez autenticado en el Sistema de Gestión Universitaria selecciona el Subsistema de Tecnologías y luego el módulo Medios tecnológicos.
-El sistema muestra las opciones de menú.
-El usuario selecciona en la agrupación funcional "Configuración" la funcionalidad "Marcas".
-El sistema muestra las marcas registradas.
-El usuario selecciona en el área de iconos flotantes la opción: Crear, llena los datos correctamente y presiona el botón Aceptar |
| EC 1.2
Insertar elementos repetidos. | Mediante este escenario se introducen datos para insertar una marca que ya existe en el sistema. | I
Acer | NA | NA | El sistema muestra un mensaje de error indicando: "El elemento ya existe." | El usuario una vez autenticado en el Sistema de Gestión Universitaria selecciona el Subsistema de Tecnologías y luego el módulo Medios tecnológicos.
-El sistema muestra las opciones de menú.
-El usuario selecciona en la agrupación funcional "Configuración" la funcionalidad "Marcas".
-El sistema muestra las marcas registradas.
-El usuario selecciona en el área de iconos flotantes la opción: Crear, llena los datos de un elemento que ya esté en el sistema y presiona el botón Aceptar. |
| EC 1.3
Insertar datos incompletos. | Mediante este escenario no se introducen todos los datos para crear una marca. | I
Vacío | NA | NA | El sistema muestra encima del componente un mensaje en color rojo indicando: "Campo requerido." | El usuario una vez autenticado en el Sistema de Gestión Universitaria selecciona el Subsistema de Tecnologías y luego el módulo Medios tecnológicos.
-El sistema muestra las opciones de menú.
-El usuario selecciona en la agrupación funcional "Configuración" la funcionalidad "Marcas".
-El sistema muestra las marcas registradas.
-El usuario selecciona en el área de iconos flotantes la opción: Crear, llena los datos de un elemento de forma incompleta y presiona el botón Aceptar. |

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

| | | | | | | |
|---|--|--------------------------------|--|----|---|--|
| EC 1.4 | Mediante este escenario se introducen datos incorrectos. | I | NA | NA | El sistema muestra en rojo el mensaje encima del componente: "Entre solo letras, números, apóstrofe, paréntesis guion y espacio o guion bajo entre palabras." | El usuario una vez autenticado en el Sistema de Gestión Universitaria selecciona el Subsistema de Tecnologías y luego el módulo Medios tecnológicos.
-El sistema muestra las opciones de menú.
-El usuario selecciona en la agrupación funcional "Configuración" la funcionalidad "Marcas".
-El sistema muestra las marcas registradas.
-El usuario selecciona en el área de íconos flotantes la opción: Crear, llena los datos de forma incorrecta y presiona el botón Aceptar. |
| | | "Acer%ad
^" | | | | |
| | | NA | I | NA | El sistema muestra en color rojo encima del componente el mensaje: "Entre al menos 2 caracteres." | |
| | | | S | | | |
| | | I | I | NA | El sistema no permite la entrada de más caracteres. | |
| | | Escribir más de 250 caracteres | Escribir una descripción que exceda de 1000 caracteres. | | | |
| | | NA | I | NA | El sistema muestra en rojo encima del componente: "Solo admite espacio entre palabras." | |
| | | | Escribir una descripción que termine con espacio. | | | |
| I | I | NA | El sistema muestra en rojo encima del componente: Ha excedido el número de caracteres permitidos para una palabra. | | | |
| El usuario entra una palabra con más de 30 caracteres | El usuario entra una palabra con más de 30 caracteres. | | | | | |
| EC 1.5 | Se cancela la creación de la marca. | NA | NA | NA | El sistema muestra el mensaje de confirmación: ¿Está seguro de realizar la acción?
Si el usuario presiona | El usuario una vez autenticado en el Sistema de Gestión Universitaria selecciona el Subsistema de Tecnologías y luego el módulo Medios tecnológicos.
-El sistema muestra las opciones de menú.
-El usuario selecciona en la agrupación |
| | | | | | | |

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

| | | | | | |
|--|--|--|--|---|---|
| | | | | <p>el botón Aceptar el sistema retorna a la página que le dio inicio sin guardar las últimas operaciones. Si el usuario presiona el botón Cancelar se mantiene en la misma vista, mostrando las últimas operaciones realizadas.</p> | <p>funcional "Configuración" la funcionalidad "Marcas".</p> <ul style="list-style-type: none"> -El sistema muestra las marcas registradas. -El usuario selecciona en el área de iconos flotantes la opción: Crear, llena o no los datos y presiona el botón Cancelar. |
|--|--|--|--|---|---|

Para obtener la calidad del funcionamiento de los requisitos funcionales se llevaron a cabo tres iteraciones, las cuales arrojaron los resultados que se muestra en la tabla 16. Las no conformidades encontradas fueron resueltas en su totalidad.

Tabla 16: Resultado de las pruebas de validación. (Elaboración propia)

| Iteración | Cantidad de no conformidades | Asociadas a |
|-----------|------------------------------|----------------------------------|
| Primera | 6 | Errores de validación de campos |
| Segunda | 4 | Errores de validación de estados |
| Tercera | 0 | - |

3.3.4 Pruebas de sistema

Las pruebas del sistema deben enfocarse en requisitos que puedan ser tomados directamente de casos de uso, reglas y funciones del negocio. El objetivo de estas pruebas es verificar el ingreso, procesamiento y recuperación apropiada de datos, y la implementación apropiada de las reglas del negocio. Este tipo de pruebas se basan en técnicas de caja negra. Guiadas a verificar el sistema (y sus procesos internos), la interacción con aplicaciones que lo usen y analizar las salidas o resultados. [37]

En esta fase se determina qué pruebas de Sistema (usabilidad, volumen, desempeño, etc.) asegurarán que la aplicación alcanzará sus objetivos de negocio. Con tal motivo se aplican pruebas de Integridad de Datos y Bases de Datos, Seguridad y Control Acceso, Desempeño y Funcionalidad, las cuales se documentan a continuación.

➤ **Prueba de Integridad de Datos y Bases de Datos**

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

Tienen como objetivo asegurar que los métodos de acceso y procesos funcionan adecuadamente y sin ocasionar corrupción de datos. Una forma sencilla de aplicar esta prueba es revisar de forma manual que al insertar o manipular una fila en base de datos la misma contenga valores acordes a su dominio y que correspondan a la acción realizada sin alterar otra fila o conjunto de datos.

Esta prueba fue aplicada para cada uno de los requisitos funcionales una vez que el sistema se encontraba integrado sin obtener no conformidades. El principal motivo de la ausencia de no conformidades al realizar esta prueba es que durante la integración paulatina del módulo se fueron realizando pruebas similares.

➤ Prueba de Seguridad y Control de Acceso

El objetivo de esta prueba es verificar que un actor solo pueda acceder a las funciones y datos que su usuario tenga permitido, y que solo los actores con acceso al sistema y a la aplicación estén habilitados para accederla.

Se detectaron tres (3) no conformidades relacionadas al rol “Designado de impresión” las cuales fueron corregidas oportunamente.

➤ Pruebas de desempeño

Las pruebas de desempeño miden tiempos de respuesta, índices de procesamiento de transacciones y otros requisitos sensibles al tiempo. El objetivo de esta prueba es validar el tiempo de respuesta para las transacciones o funciones de negocios bajo las siguientes dos condiciones:

- **Volumen normal anticipado:** Estimación del volumen de datos y peticiones con el que se espera trabaje el sistema normalmente.
- **Volumen máximo anticipado:** Estimación del volumen de datos y peticiones con el que se espera trabaje o sea capaz de procesar el sistema en el peor de los casos.

Resultados:

Se realizaron las pruebas para comprobar que el tiempo de respuesta del componente no exceda a los 5 segundos de forma general, así como el tiempo de respuesta por cada transacción promedio sea un máximo de 3 segundos. La figura 19 muestra la prueba de desempeño realizada en el navegador web Mozilla Firefox a la funcionalidad crear marca.

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

| | | | |
|-------|------|---|-------------------------|
| ▶ XHR | POST | http://localhost/proyecto_sgu/base/medios_tecnologicos_b/marca/registrar | [HTTP/1.1 200 OK 165ms] |
| ▶ XHR | POST | http://localhost/proyecto_sgu/base/medios_tecnologicos_b/marca/registrarMarca | [HTTP/1.1 200 OK 197ms] |
| ▶ XHR | POST | http://localhost/proyecto_sgu/base/notificaciones_alertas/alertas/obtenerCantidadAlertas | [HTTP/1.1 200 OK 154ms] |
| ▶ XHR | POST | http://localhost/proyecto_sgu/base/notificaciones_alertas/notificaciones/obtenerCantidad... | [HTTP/1.1 200 OK 182ms] |
| ▶ XHR | POST | http://localhost/proyecto_sgu/base/medios_tecnologicos_b/marca/listar | [HTTP/1.1 200 OK 147ms] |
| ▶ XHR | POST | http://localhost/proyecto_sgu/base/medios_tecnologicos_b/marca/obtenerMarcas | [HTTP/1.1 200 OK 167ms] |

Figura 19: Prueba de desempeño funcionalidad crear marca.

3.4 Validación de hipótesis

Para realizar la validación de la hipótesis de investigación se aplicó el método Criterio de expertos a través del cumplimiento de los pasos siguientes.

1. Identificación de los posibles expertos.
2. Determinación del coeficiente de competencia de los candidatos a expertos.
3. Selección de los expertos.
4. Realización del cuestionario a los expertos.
5. Aplicación del escalamiento de Likert.

Identificación de los posibles expertos: Para la identificación de los posibles expertos se tuvo en cuenta la experiencia profesional en el tema, de modo que tuviera la capacidad de ofrecer valoraciones y hacer recomendaciones pertinentes en relación con los aspectos que les serían consultados. [38]

Determinación del coeficiente de competencia de los candidatos a expertos: Una vez identificados los posibles expertos, se les realizó una encuesta para valorar el grado de conocimiento y el grado de influencia que cada una de las fuentes ha tenido en ese conocimiento.

Selección de expertos: Fueron seleccionados expertos con nivel alto y medio del coeficiente de competencia, dado que el coeficiente de competencia promedio del grupo de expertos no era inferior a 0.7 manteniéndose un nivel alto en el conjunto. En la siguiente tabla se muestran los expertos seleccionados.

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

Tabla 17: Selección de expertos. (Elaboración propia)

| Experto | Área | Cargo | Coeficiente |
|-----------------------------------|---|--------------------------------|-------------|
| Yamila Mateu Romero | Dirección de Informatización.
Departamento de Desarrollo | Profesor | 0.80 |
| Laritzza González
Marrero | Dirección de Informatización.
Departamento de Desarrollo | Profesor | 0.87 |
| José Alejandro García
Calderón | Dirección de Informatización.
Departamento de Desarrollo | Jefe de
Departamento | 0.95 |
| Deborah Serrano
García | Dirección de Informatización.
Departamento de Desarrollo | Diseñador de Bases
de Datos | 0.70 |
| Alejandro García
Gómez | Dirección de Informatización.
Departamento de Desarrollo | Programador | 0.72 |
| Mary Nelsa Bonne Cuza | Dirección de Informatización.
Departamento de Desarrollo | Analista | 0.80 |
| Roberkis Terrero
Galano | Dirección de Informatización.
Departamento de Desarrollo | Profesor | 0.80 |

Realización del cuestionario a los expertos: Una vez seleccionado los expertos se les realizó un cuestionario compuesto por ocho (8) afirmaciones evaluativas que permiten conocer la opinión del experto. Los parámetros evaluativos empleados fueron: MA (muy de acuerdo), DA (de acuerdo), N (Ni de acuerdo ni en desacuerdo), ED (en desacuerdo), CD (completamente en desacuerdo).

Los parámetros evaluativos se cuantificaron a través de la siguiente escala:

- Muy de acuerdo (100)
- De acuerdo (75)
- Ni de acuerdo ni en desacuerdo (50)
- En desacuerdo (25)
- Completamente en desacuerdo (0)

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

Aplicación del escalamiento de Likert: Para el procesamiento de los resultados se empleó un método que consiste en identificar la media de las diferentes categorías de la escala Likert. Dicho método se aplicó por separado a cada una de las afirmaciones. Luego se obtuvo la media de las valoraciones individuales para obtener una valoración integral del sistema. [39]

La aplicación del escalamiento Likert evidencia que la propuesta solución tiene una alta valoración por parte de los expertos. Durante el proceso se constataron criterios favorables en el uso del módulo para mejorar la eficiencia en el proceso de registro de medios tecnológicos personales. Se obtuvo un resultado satisfactorio, con un valor superior a un 90 % de aprobación por parte de los expertos.

3.5 Conclusiones del capítulo

En el presente capítulo se abordaron elementos de la implementación del módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas, así como las pruebas realizadas y los resultados obtenidos. El autor arriba a las siguientes conclusiones:

- La identificación de estándares y buenas prácticas de codificación guiaron un desarrollo claro y fluido de las funcionalidades del módulo.
- Las pruebas realizadas permitieron identificar y corregir oportunamente errores en el módulo.
- La validación de hipótesis aplicando el método Criterio de Expertos arrojó resultados favorables con un valor superior a un 90% de aprobación.

CONCLUSIONES GENERALES

Una vez realizadas las tareas trazadas en la presente investigación se concluye lo siguiente:

- El estudio de diferentes sistemas para el inventario de medios tecnológicos y procedimiento para la gestión de solicitudes permitieron obtener una idea temprana de los principales conceptos asociados a la investigación.
- El levantamiento de requisitos aplicando Historias de Usuario permitió realizar una correcta descripción y modelación del problema, obteniéndose 36 requisitos funcionales y 11 requisitos no funcionales.
- Los cinco patrones de diseño identificados se cohesionan al estilo arquitectónico Modelo-Vista-Controlador. Esto evidencia que la propuesta solución cuenta con un alto grado de flexibilidad ante posibles modificaciones.
- El empleo de iteraciones de Scrum promovió un flujo constante de retroalimentación con el cliente y poder corregir oportunamente las no conformidades encontradas.
- Una vez aplicadas las pruebas de software al módulo, se garantiza el cumplimiento de las necesidades y requisitos establecidos por el cliente.
- La validación de la hipótesis con criterio de expertos reflejó que el módulo contribuye a mejorar la eficiencia en el proceso de registro de medios tecnológicos.

RECOMENDACIONES

El autor realiza la siguiente recomendación:

- Incorporar una funcionalidad que permita capturar los atributos del medio de manera automática al interactuar con el módulo del sistema.

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

BIBLIOGRAFÍA

- [1] M. Lavin, «¿Qué es BYOD?, ventajas e inconvenientes», *ComputerHoy*, 10-dic-2013. [En línea]. Disponible en: <https://computerhoy.com/noticias/moviles/que-es-byod-ventajas-e-inconvenientes-7250>. [Accedido: 16-may-2019].
- [2] P. Espeso, «BYOD, trae tu propio dispositivo: el modelo que quiere revolucionar la educación», *EDUCACIÓN 3.0*, 05-oct-2018. [En línea]. Disponible en: <https://www.educaciontrespuntocero.com/noticias/byod-bring-your-own-device-educacion/32857.html>. [Accedido: 16-may-2019].
- [3] «Misión | Universidad de las Ciencias Informáticas». [En línea]. Disponible en: <https://www.uci.cu/universidad/mision>. [Accedido: 16-may-2019].
- [4] A. A.-A. Jaramillo y Gabriel Jaime Correa-Henao, «Construcción de un aplicativo para el control de activos tecnológicos al interior de una organización», p. 11, 2013.
- [5] «Definición de Inventario», *Definición ABC*. [En línea]. Disponible en: <https://www.definicionabc.com/economia/inventario.php>. [Accedido: 16-may-2019].
- [6] «Network Inventory Advisor: Software de gestión de activos de PC», 14-nov-2018. [En línea]. Disponible en: <https://www.network-inventory-advisor.com/es/>. [Accedido: 14-nov-2018].
- [7] Partido Comunista de Cuba, «Actualización de los lineamientos de la política económica y social del Partido y la Revolución para el período 2016-2021.» abr-2016.
- [8] superintendencia de Sociedades, «Procedimiento de gestión de cambios en ambiente productivo». 03-feb-2017.
- [9] Darién Rodríguez Prieto, «Sistema automatizado para el control de medios informáticos», Universidad Central Marta Abreu de las Villas, Villa Clara, Santa Clara, 2014.
- [10] R. L. L. Bladimir Almeida Rodríguez, «Módulo de migración de información de OCS Inventory NG 2.0.5 a GRHS 1.0», tesis de grado, Universidad de las Ciencias Informáticas, La Habana, Cuba, 2015.
- [11] «What is Unified Modeling Language (UML)? - Definition from Techopedia», 14-nov-2018. [En línea]. Disponible en: <https://www.techopedia.com/definition/3243/unified-modeling-language-uml>. [Accedido: 14-nov-2018].
- [12] «Visual Paradigm Product Overview», 14-nov-2018. [En línea]. Disponible en: https://www.visual-paradigm.com/support/documents/vpuserguide/12/13/5963_visualparadi.html. [Accedido: 14-nov-2018].
- [13] «Git - Book», 14-nov-2018. [En línea]. Disponible en: <https://git-scm.com/book/es/v2>. [Accedido: 14-nov-2018].
- [14] P. mastermarketingdigital_s5giyb, «php lenguaje de programacion ☺ ▷ 【 ¿Qué es el PHP? 】 », *Master Marketing Digital* .
- [15] «Definición de JavaScript» Concepto en Definición ABC», 14-nov-2018. [En línea]. Disponible en: <https://www.definicionabc.com/tecnologia/javascript.php>. [Accedido: 14-nov-2018].
- [16] «¿Qué es HTML5? | HTML5», 14-nov-2018. [En línea]. Disponible en: <https://www.arkaitzgarro.com/html5/capitulo-1.html>. [Accedido: 14-nov-2018].
- [17] «Capítulo 1. Introducción (Introducción a CSS)», 14-nov-2018. [En línea]. Disponible en: <https://librosweb.es/libro/css/capitulo-1.html>. [Accedido: 14-nov-2018].
- [18] J. F.- js.foundation, «jQuery» .
- [19] «JSON», 14-nov-2018. [En línea]. Disponible en: <https://www.json.org/json-es.html>. [Accedido: 14-nov-2018].
- [20] «nginx», 04-dic-2018. [En línea]. Disponible en: <https://nginx.org/en/>. [Accedido: 04-dic-2018].
- [21] «PostgreSQL: The world's most advanced open source database», 14-nov-2018. [En línea]. Disponible en: <https://www.postgresql.org/>. [Accedido: 14-nov-2018].

Módulo para el registro de medios tecnológicos personales en la Universidad de las Ciencias Informáticas

- [22] «pgAdmin - PostgreSQL Tools», 14-nov-2018. [En línea]. Disponible en: <https://www.pgadmin.org/>. [Accedido: 14-nov-2018].
- [23] «Información NetBeans IDE 6.1», 14-nov-2018. [En línea]. Disponible en: https://netbeans.org/community/releases/61/index_es.html. [Accedido: 14-nov-2018].
- [24] «Metodología de desarrollo de software», *Concepto de - Definición de*, 05-ago-2016. .
- [25] «Metodología de Desarrollo Ágil: XP y Scrum – INGENIERÍA DEL SOFTWARE», 09-dic-2018. [En línea]. Disponible en: <https://ingsoftwarekarlacevallos.wordpress.com/2015/05/08/metodologia-de-desarrollo-agil-xp-y-scrum/>. [Accedido: 09-dic-2018].
- [26] «Qué es SCRUM – Proyectos Ágiles», 14-nov-2018. [En línea]. Disponible en: <https://proyectosagiles.org/que-es-scrum/>. [Accedido: 14-nov-2018].
- [27] P. por pmoinformatica.com, «Requerimientos no funcionales: Ejemplos». .
- [28] E. Ledesma, «Certificado en Fundamentos de Scrum SFC™ GRATIS», *Capacitación, Consultoría Dirección Proyectos TenStep*. [En línea]. Disponible en: <https://www.tenstep.ec/portal/articulos-boletin-tenstep/41-scrum/253-scrum-como-escribir-historias-de-usuarios-sin-morir-en-el-intento>. [Accedido: 17-may-2019].
- [29] «IBM notice: The page you requested cannot be displayed», 21-may-2016. [En línea]. Disponible en: undefined. [Accedido: 17-may-2019].
- [30] «Los 10 patrones comunes de arquitectura de software». [En línea]. Disponible en: <https://medium.com/@maniakhitoccori/los-10-patrones-comunes-de-arquitectura-de-software-d8b9047edf0b>. [Accedido: 17-may-2019].
- [31] B. R. M. Roger S. Pressman, *Software Engineering. A Practitioner's Approach*, 8th ed. New York: McGraw-Hill Education, 2015.
- [32] «Modelo vista controlador (MVC)». [En línea]. Disponible en: <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>. [Accedido: 17-may-2019].
- [33] C. A. Guerrero, J. M. Suárez, y L. E. Gutiérrez, «Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web», *Inf. Tecnológica*, vol. 24, n.º 3, pp. 103-114, 2013.
- [34] campusMVP, «Diseñando una base de datos en el modelo relacional», *campusMVP.es*. [En línea]. Disponible en: <https://www.campusmvp.es/recursos/post/disenando-una-base-de-datos-en-el-modelo-relacional.aspx>. [Accedido: 17-may-2019].
- [35] «Guía de buenas prácticas para programar en PHP», *Formación Online*, 22-jun-2016. [En línea]. Disponible en: <https://formaciononline.eu/buenas-practicas-para-programar-en-php/>. [Accedido: 17-may-2019].
- [36] «Técnicas de validación de requisitos | Marco de Desarrollo de la Junta de Andalucía». [En línea]. Disponible en: <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/419>. [Accedido: 17-may-2019].
- [37] J. H. A. Londoño, «Ingeniería de Software: TIPOS DE PRUEBAS DE SOFTWARE», *Ingeniería de Software*, 06-abr-2005. .
- [38] «CRITERIO DE EXPERTOS. SU PROCESAMIENTO A TRAVÉS DEL MÉTODO DELPHY». [En línea]. Disponible en: http://www.ub.edu/histodidactica/index.php?option=com_content&view=article&id=21:criterio-de-expertos-su-procesamiento-a-traves-del-metodo-delphy&catid=11:metodologia-y-epistemologia&Itemid=103. [Accedido: 17-may-2019].
- [39] O. Llauradó, «La escala de Likert: qué es y cómo utilizarla». [En línea]. Disponible en: <https://www.netquest.com/blog/es/la-escala-de-likert-que-es-y-como-utilizarla>. [Accedido: 17-may-2019].