

Universidad de las Ciencias Informáticas

Facultad 2



Título: “Sistema de Control de Laboratorios (CLAB).”

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

Autores: Luis Olfrides Pérez Rodríguez.

Daliana González Martínez.

Tutora:

Dainelys Espinosa Marrero.

Declaración de autoría

Declaramos ser los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Daliana González Martínez

Luis Olfrides Pérez Rodríguez

Firma del autor

Firma del autor

Dainelys Espinosa Marrero

Firma del tutor

Pensamiento

“Las ideas no necesitan ni de las armas, en la medida que sean capaces de conquistar a las grandes masas.”

Fidel Castro Ruz



Agradecimientos

A mi mamá y a mi papá por confiar en mí, y formar parte indispensable en mi vida, gracias a ellos por ser de mí lo que soy hoy en día.

A todos mis hermanos por ayudarme en todo lo que necesitaba.

A toda mi familia y en especial a mi tía Rosita que de una forma u otra han puesto su granito de arena en todo este tiempo.

A Asiel por haber pasado muchos momentos conmigo y por toda la ayuda que me ha brindado en los últimos 2 años.

A mis amigas Danima y Linet por todos sus consejos y estar siempre pendientes de mí. Muchas gracias por brindarme su amistad.

A Frank Ernesto por ayudarme en el desarrollo de este trabajo sin importarle la hora, es mi programador favorito.

A mi compañero de tesis por su dedicación y por tener mucha paciencia conmigo.

A mi tutora y a todos los profesores que han influido en mi formación.

Gracias a todos por ser realidad este sueño.

De Daliana.

A mi familia por el apoyo y la confianza depositados en mí.

A mi compañera de Tesis por estar siempre preocupada de lo que faltaba por hacer; que por suerte hoy no falta nada.

A mis amigos por su incondicionalidad mostrada en todo momento durante estos cinco años.

A mis profesores por enseñarme tanto en tan poco tiempo, en especial los de Inglés.

A todas las personas que de una forma u otra me han ayudado en esta etapa de mi vida para que pudiera realizar este sueño.

De Luis.

Dedicatoria

A mi mamá, a mi papá, a mis hermanos y a toda mi familia, en especial a mi abuela que ya no está conmigo.

De Daliana.

A mi familia, por saber estar ahí cuando más los necesitaba, por impulsarme y alentarme a seguir adelante, cuando todo parecía que no tenía salida. A ustedes les debo lo que soy, a ustedes que han sabido educarme. Gracias por ser tan especiales en mi vida.

De Luis.

Resumen

En la Universidad de las Ciencias Informáticas (UCI) el plan de estudio incluye el vínculo de estudiantes y profesores a proyectos productivos. Para ello se crearon laboratorios de proyectos que son atendidos por técnicos informáticos; ellos son los encargados de realizar los diferentes procesos que se llevan a cabo en dichos laboratorios.

No obstante, el control de los medios del laboratorio no cumple en su totalidad con los procesos definidos para ello, y el acceso de personal ajeno al Centro no es debidamente controlado; todo ello conlleva a que personal ajeno a los proyectos acceda a los equipos informáticos y a información que puede ser delicada. Además, los reportes de control de medios se realizan actualmente en modelos de papel, corriendo riesgos de pérdida o deterioro de información y dificultando su búsqueda; y no se cuenta con un medio digitalizado de asignar tareas a dichos técnicos y darles seguimiento y evaluación dejando una constancia de ello.

Por tanto el Centro de Telemática (TLM) de la Facultad 2 de la UCI propuso la realización de una aplicación informática que fuera capaz de gestionar los procesos relacionados con el control de acceso a los laboratorios y los medios existentes en los mismos, lo cual constituye el objetivo del presente trabajo de diploma.

Palabras Clave: Control, Laboratorios, Medios, Reportes.

Índice de contenidos

Introducción:	1
Capítulo 1: Fundamentación Teórica.....	5
1.1. Introducción:.....	5
1.2. Estudio del Estado del Arte	5
1.2.1 Sistemas para el Control de Acceso del Personal	5
1.2.2. Sistemas de Gestión de Reportes de las afectaciones de los medios	7
1.3. Lenguaje de programación, metodología, herramientas y tecnologías.....	8
1.3.1. Metodología de desarrollo del Software.....	8
1.3.2. Herramienta CASE para la modelación del sistema	9
1.3.2.1. Visual Paradigm 8.0	9
1.3.3. Lenguaje de Programación	10
1.3.4. Entorno de Desarrollo Integrado	11
1.3.5. Servidor de Base de Datos	11
1.3.6. Tecnología del lado del servidor	12
1.3.7. Tecnología del lado del cliente.....	13
1.4. Conclusiones parciales	14
Capítulo 2: Características y Propuesta del Sistema.	15
2.1. Introducción.....	15
2.2. Propuesta del Sistema	15
2.3. Lista de reserva del producto	16
2.4. Requisitos no funcionales del sistema	17
2.5. Planeación.....	18
2.5.1. Involucrados en el Sistema	18
2.5.2. Historias de Usuario.....	19
2.5.3. Estimación de esfuerzo por HU	26
2.5.4. Plan de Iteraciones.	28
2.6. Conclusiones Parciales.	31
Capítulo 3: Diseño.	32
3.1. Introducción.....	32
3.2. Patrones de Arquitectura:.....	32
3.2.1. Patrón arquitectónico: Modelo Vista Controlador (Model View Controller, MVC)	32
3.3. Patrones de Diseño.....	32
3.3.1. Patrones para Asignar Responsabilidades (GRASP)	33

3.4.	Estilo Arquitectónico.....	34
3.4.1.	Componentes del estilo arquitectónico Cliente/Servidor.....	35
3.4.2.	Elementos principales	35
3.5.	Arquitectura del Sistema	36
3.5.1.	Arquitectura N-Capas	36
3.6.	Diseño de la Base de Datos.....	37
3.7.	Tarjetas Clases-Responsabilidad-Colaborador.....	38
3.8.	Conclusiones Parciales	40
Capítulo 4: Implementación y Prueba.....		41
4.1.	Introducción.....	41
4.2.	Fase de Implementación	41
4.2.1.	Iteración 1	41
4.2.2.	Iteración 2	47
4.2.3.	Iteración 3	48
4.3.	Diagrama de Despliegue.....	48
4.4.	Pruebas	49
4.4.1.	Pruebas Unitarias.....	49
4.4.2.	Pruebas de Aceptación	50
4.5.	Conclusiones Parciales	52
Conclusiones Generales.....		53
Recomendaciones		54
Referencias Bibliográficas		55

Índice de Tablas:

Tabla 2.1: Involucrados en el Sistema.....	19
Tabla 2.2: Historia de Usuario Autenticar Usuario.....	21
Tabla 2.3: Historia de Usuario Cerrar sesión.....	21
Tabla 2.4: Historia de Usuario Insertar Usuario.....	21
Tabla 2.5: Historia de Usuario Modificar Usuario.	22
Tabla 2.6: Historia de Usuario Buscar Usuario.....	22
Tabla 2.7Historia de Usuario: Eliminar Usuario.....	23
Tabla 2.8: Historia de Usuario: Insertar Medio.	23
Tabla 2.9: Historia de Usuario: Modificar Medio.	23
Tabla 2.10: Historia de Usuario: Buscar Medio.	24
Tabla 2.11: Historia de Usuario: Eliminar Medio.	24
Tabla 2.12: Historia de Usuario: Insertar Laboratorio.....	25
Tabla 2.13: Historia de Usuario: Modificar Laboratorio.	25
Tabla 2.14: Historia de Usuario: Buscar Laboratorio.....	26
Tabla 2.15: Historia de Usuario: Eliminar Laboratorio.....	26
Tabla 2.16: Estimación de esfuerzo por historia de usuarios.....	28
Tabla 2.17: Plan de duración de las iteraciones.....	30
Tabla 3.1: Tarjetas CRC Clase: Usuario.	39
Tabla 3.2: Tarjetas CRC Clase: Medio.....	39
Tabla 3.3: Tarjetas CRC Clase: Laboratorio.....	39
Tabla 3.4: Tarjetas CRC Clase: Reporte.....	40
Tabla 3.5: Tarjetas CRC Clase: Tarea.	40
Tabla 4.1: Historias de Usuarios de la iteración 1.	42
Tabla 4.2: Tarea de Ingeniería # 1: Autenticar Usuario.....	42
Tabla 4.3: Tarea de Ingeniería # 2: Cerrar Sesión.	43
Tabla 4.4: Tarea de Ingeniería # 3: Insertar Usuario.....	43
Tabla 4.5: Tarea de Ingeniería # 4: Modificar Usuario.	43
Tabla 4.6: Tarea de Ingeniería # 5: Buscar Usuario.....	44
Tabla 4.7: Tarea de Ingeniería # 6: Eliminar Usuario.	44
Tabla 4.8: Tarea de Ingeniería # 7: Insertar Medio.	45
Tabla 4.9: Tarea de Ingeniería # 8: Modificar Medio.....	45
Tabla 4.10: Tarea de Ingeniería # 9: Buscar Medio.	45
Tabla 4.11: Tarea de Ingeniería # 10: Eliminar Medio.....	46
Tabla 4.12: Tarea de Ingeniería # 11: Insertar Laboratorio.....	46
Tabla 4.13: Tarea de Ingeniería # 12: Modificar Laboratorio.....	47

Tabla 4.14: Tarea de Ingeniería # 13: Buscar Laboratorio.....	47
Tabla 4.15: Tarea de Ingeniería # 14: Eliminar Laboratorio.....	47
Tabla 4.16: Historias de Usuarios de la iteración 2.....	48
Tabla 4.17: Historias de Usuarios de la iteración 3.....	48

Índice de Figuras:

Figura 1: Control Diario.....	2
Figura 2: Estilo arquitectónico cliente/servidor	36
Figura 3: Arquitectura N-Capas	37
Figura 4: Modelo Entidad Relación.....	38
Figura 5: Diagrama de Despliegue	49
Figura 6: Prueba Unitaria a la clase IIN_Proyecto.....	50

Introducción:

La Universidad de las Ciencias Informáticas (UCI), es una universidad que cuenta con un plan de estudio diferente al del resto de las universidades del país teniendo como misión: “Formar profesionales comprometidos con su Patria y altamente calificados en la rama de la Informática. Producir aplicaciones y servicios informáticos, a partir de la vinculación estudio-trabajo como modelo de formación. Servir de soporte a la industria cubana de la informática”.(1)

Para una mayor organización en la producción la universidad está formada por siete (7) facultades donde cada una responde a diferentes perfiles productivos. Entre ellas se encuentra la Facultad 2 donde radican dos centros de producción: Informatización de la Seguridad Ciudadana (ISEC) y Telemática (TLM). Estos centros están compuestos por varios proyectos, donde se realizan actividades en función de la fabricación de software. Dichas actividades son llevadas a cabo en el área de los laboratorios dedicados especialmente a la producción.

Actualmente en el Centro TLM los procesos relacionados con el control de acceso de las personas a los laboratorios no se están realizando correctamente, el personal responsable de realizar esta tarea, conocidos como técnicos de laboratorios no cuenta con los datos ni los medios necesarios para verificar quién tiene permiso para acceder al local. Como consecuencia a lo planteado anteriormente accede e interactúa con los medios, personal no autorizado.

El control diario de los medios básicos se realiza manualmente por parte de la dirección de laboratorios, siendo los técnicos los encargados de manejar dicha información, almacenando los datos en un modelo de papel, resultando engorroso a la hora de manipular los mismos; dígase realizar una búsqueda, o modificar algún dato, afectando así la calidad y rapidez del proceso. Todo esto trae consigo:

- ✓ La búsqueda y recuperación de datos a la hora de hacer reportes es lenta.
- ✓ La pérdida y deterioro de datos.
- ✓ Se acumulan grandes cantidades de modelos.

A continuación se presenta el modelo de control diario que utilizan los técnicos actualmente para el control de los medios.

CONTROL DIARIO DOCENCIA				Turno		Turno		Laboratorio			Area	FECHA						
				Ent	Rec	Ent	Rec					Día	Mes	Año				
				Cantidades(U)				Afectaciones Generales.			Afectación al Servicio							
Perifericos	Ent	Rec	Ent	Rec	Ent	Aspectos	N/E	N/ENF		Por:	No de PCs		T					
PC						Aire 1				DVD/CD								
Monitores						Aire 2				HDD								
Teclados						Medios no Tec	B	R	M	Red/PC								
Mouse					Est.puertas								Red/Tor-Par					
Sillas					Est.Ventanas								Mouse					
Mesas						Est.Luces				Teclado								
Switch						Estad.de la RED				Video								
Cables de Red						Limpieza				Se Reinicia								
C.Aliment.										S_Rotos								
UPS																		
Sellos/Cdados																		
Cesto						Técnicos Involucrados en la Entrega												
Observaciones:				Entrega		Hora	Día	Mes	Año	Recibe		Hora	Día	Mes	Año			
				Nombre:								Nombre:						
				Solapín								Solapín						
				Firma								Firma						
				Entrega		Hora	Día	Mes	Año	Recibe		Hora	Día	Mes	Año			
				Nombre:								Nombre:						
				Solapín								Solapín						
				Firma								Firma						
				HDD(Disco Duro),B(Bien),R(Regular),M(Mal),N/E(No Enciende),N/ENF(No Enfria),T(Total)														

Figura 1: Control Diario.

Por otra parte no se cuenta con un sistema informático mediante el cual el asesor de tecnología pueda asignarles y evaluarles las tareas de desempeño a los técnicos y estos puedan notificar el estado en el que se encuentra.

Con el fin de solucionar las dificultades antes planteadas, surge el siguiente **problema a resolver**: ¿Cómo contribuir al correcto funcionamiento de los procesos de control de los laboratorios en el Centro de Telemática de la UCI? Según el problema identificado anteriormente el **objeto de estudio** se enmarca en los procesos que se llevan a cabo para el control de los laboratorios teniendo como **campo de acción** los procesos que se llevan a cabo para el control de los laboratorios del Centro de TLM.

El **objetivo general** de la investigación es desarrollar un sistema informático que sea capaz de gestionar los procesos que se llevan en el Centro de Telemática de la UCI para el control de los laboratorios.

Con el fin de alcanzar el objetivo planteado se definen los siguientes **objetivos específicos**:

- Identificar los problemas existentes en el proceso de control de los laboratorios del Centro de Telemática de la UCI.
- Realizar un estudio sobre los sistemas que se utilizan en la UCI para llevar a cabo el proceso de control de los laboratorios.
- Diseñar e implementar un sistema para el control de los laboratorios en el Centro de Telemática de la UCI.

- Validar el sistema propuesto.

Con el propósito de cumplir con todos los objetivos planteados se definen las siguientes **tareas de investigación:**

- Análisis de los sistemas existentes en el mundo, en Cuba y en la UCI, para identificar características aplicables a la propuesta de solución.
- Descripción de las herramientas a usar en la implementación del sistema para lograr un mejor desarrollo en el sistema propuesto.
- Descripción de los requisitos del sistema para lograr un mejor alcance de las necesidades del cliente.
- Implementar las funcionalidades definidas en los requisitos para obtener el sistema deseado.
- Realizar pruebas al sistema para validar su correcto funcionamiento.

Con el desarrollo de este trabajo se obtendrá como **resultado** un sistema capaz de gestionar los procesos de control de laboratorios identificando las funcionalidades permitidas a cada usuario según el rol que presente en el sistema.

Para la investigación se utilizan los siguientes métodos:

Métodos teóricos:

Analítico-Sintético: Este método se utilizará con el fin de analizar libros, páginas web y otras bibliografías para buscar los elementos más importantes que se relacionan con esta investigación.

Modelación: Este método se utilizará con el objetivo de reproducir la interacción de los objetos en la vida real mediante la creación de modelos, para descubrir nuevas relaciones y cualidades del objeto de estudio. Mediante la utilización del lenguaje de modelado UML, permitirá reflejar la estructura, relaciones internas y características de la solución a través de diagramas.

El **método empírico** utilizado:

Entrevista: Este método se utilizará para establecer una conversación planificada con los usuarios a los cuales va dirigido la aplicación sobre el sistema que utilizan y sobre el sistema a desarrollar. Los requerimientos provienen de las respuestas a estas preguntas.

El presente documento consta de 4 capítulos, estructurados de la siguiente forma:

Capítulo 1: Fundamentación Teórica.

En este capítulo se realiza el estudio del estado del arte y de aplicaciones web existentes como posibles soluciones al control de los laboratorios. Se realiza un estudio de las metodologías, herramientas y tecnologías que existen seleccionando las más adecuadas para dar cumplimiento al objetivo propuesto.

Capítulo 2: Características y propuesta del Sistema.

En este capítulo se identifican los requisitos funcionales y no funcionales a tener en cuenta. Se realiza un estudio del negocio para identificar los procesos fundamentales que se quieren automatizar. Se describen las historias de usuarios.

Capítulo 3: Diseño.

Este capítulo contiene todo el diseño del sistema propuesto. Se definen los patrones de diseño así como las clases del negocio a través de las Tarjetas de Cargo o Clase, Responsabilidad y Colaboración (CRC).

Capítulo 4: Implementación y Prueba.

Se expone todo lo relacionado a los procesos de implementación y pruebas del sistema.

Capítulo 1: Fundamentación Teórica.

1.1. Introducción:

En este capítulo se realiza el estudio del estado del arte y de aplicaciones existentes en el ámbito internacional, nacional y en específico en la UCI que pueden representar posibles soluciones al control de los laboratorios. Se realiza un estudio de las metodologías, herramientas y tecnologías que existen seleccionando las más adecuadas para dar cumplimiento al objetivo propuesto.

1.2. Estudio del Estado del Arte

1.2.1 Sistemas para el Control de Acceso del Personal

1.2.1.1 Biomesys Control de Asistencia

Biomesys Control de Asistencia: es un sistema que aprovecha las bondades de las tecnologías que aplican la biometría, para registrar los eventos de asistencia en una organización por medio de la identificación de los empleados y de la autenticación de su identidad mediante un sensor biométrico de huellas dactilares. A partir de la captura de identificaciones biométricas únicas, el sistema se convierte en un generador de datos altamente confiable por su bajo o casi nulo nivel de vulnerabilidad por la suplantación de identidad. De utilidad para los que pretenden apoyarse en un instrumento sencillo con un enfoque flexible y adaptable a la estructura funcional de la organización, que propone herramientas personalizables para el control y la toma de decisiones en el área de los recursos humanos.

Características distintivas:

- ✓ No establece limitaciones de implementación asociadas al tipo de organización pudiendo ser instalado en la mayoría de los negocios públicos o privados no importa cuál sea su tamaño.
- ✓ Se puede integrar de forma rápida con diferentes medios de autenticación como escáneres biométricos, credenciales de bandas magnéticas, tarjetas de códigos de barras y proximidad.
- ✓ Posee un módulo de captura biométrica para el control de las entradas y las salidas del personal.

- ✓ Posee Interfaz de trabajo en ambiente WEB, para la administración y obtención de informes estadísticos, documentos que facilitan a los decisores el control de la asistencia del personal.(2)

No obstante para un óptimo aprovechamiento del sistema se requiere de tecnología como el lector biométrico de huellas dactilares que requiere y no se cuenta con ella en el Centro de TLM.

1.2.1.2 Sistema de Informatización UCID¹/Control de Acceso

Sistema de Informatización UCID es una aplicación desarrollada por UCI-FAR² que cuenta con varios módulos; entre ellos se encuentran el Control de Acceso, que presenta como principal objetivo gestionar todas las personas que existen en esta área con el fin de controlar su acceso. Este proceso se lleva a cabo mediante un lector de código de barra que lee el código existente en el solapín de la persona que desea acceder al local. El sistema le comunica si tiene permiso o no, en caso de tenerlo registra el horario de entrada y salida de esta persona.

Este sistema fue desarrollado para UCI-FAR y presenta características militares a la hora de brindar solución a las diferentes problemáticas que se atienden en el sistema, lo que hace imposible su aplicación para el control de laboratorios, pero se tomó en cuenta como objeto de estudio

1.2.1.3 Control de Acceso a Comedores (UCI)

Control de Acceso a Comedores tiene como objetivo fundamental gestionar los accesos a los comedores de la UCI, este proceso consiste en: registrar accesos comprobando que dicha persona existe, tiene autorización para pasar por esa puerta y no ha accedido con anterioridad, sincronizar accesos o actualizar cada cierto tiempo la información registrada en la base de datos local con la base de datos central a través de un servicio web; y brindar reportes donde se muestre al usuario la cantidad de personas por tipo (trabajadores y estudiantes) que han tenido acceso hasta el momento. Su desarrollo estuvo a cargo de un equipo de trabajo de la propia universidad.

Ventajas del producto

- ✓ Evita el acceso de comensales no autorizados a los comedores.

¹Unidad de compatibilización, integración y desarrollo de productos informáticos para la defensa.

²Fuerzas Armadas Revolucionarias.

- ✓ Conexión con lector de código de barras.
- ✓ Cantidad máxima de accesos permitidos por la puerta de acceso en un evento determinado.
- ✓ Reportes de consumos por categorías.(3)

Este sistema se tomó en cuenta al estar desarrollado en la propia universidad y por establecer controles de acceso por áreas específicas, utilizando el medio de identificación de la UCI. Esta aplicación no presenta funcionalidades como la gestión de personas, de reportes y medios.

1.2.2. Sistemas de Gestión de Reportes de las afectaciones de los medios

1.2.2.1. GATSERVER

GATSERVER es una aplicación web desarrollada por el Grupo de Asistencia Técnica (GAT) de la UCI para la gestión de las afectaciones tecnológicas, que permite la gestión de reportes para las afectaciones de (medios computacionales, redes y telefonía). La aplicación permite insertar reportes de roturas y generar las soluciones de los mismos. La misma está desarrollada con la tecnología privativa ASP (Active Server Page), además de que no contempla la gestión de personas ni de tareas de usuarios.

1.2.2.2. GESPRO

GESPRO es un paquete para la gestión de proyectos desarrollado por la Universidad de las Ciencias Informáticas que se utiliza para el seguimiento y control de los proyectos de la universidad, dentro de las funcionalidades que brinda el paquete, se encuentra la seguridad y control de los usuarios, la evaluación de las tareas de desempeño que estén asignadas para cada usuario, los reportes estadísticos a nivel de universidad, por centros y por proyectos.

Los principales módulos que componen el sistema son los siguientes:

- ✓ Paquete de herramientas de gestión de proyectos en la organización (módulo básico)
- ✓ Subsistemas de dirección integrada de proyectos.
- ✓ Subsistemas de gestión de reportes dinámicos.
- ✓ Subsistemas para la gestión documental y el control de versiones.
- ✓ Subsistema de autenticación y seguridad.
- ✓ Subsistema de noticias y trabajo colaborativo y la vigilancia tecnológica.

- ✓ Subsistema de comunicación con correo y la mensajería.
- ✓ Subsistema para análisis estadísticos y monitoreo.
- ✓ Paquete de servicios de soporte y acompañamiento.
- ✓ Paquete de servicios de salvadas automáticas de recuperación de la información.
- ✓ Paquete de cursos de formación avanzados en Gestión de Proyectos (Maestría en Gestión de Proyectos).(4)

Sin embargo, la estructura que se desea desarrollar es más sencilla y orientada a la gestión de medios, dicha funcionalidad no está incluida en el sistema.

1.3. Lenguaje de programación, metodología, herramientas y tecnologías

1.3.1. Metodología de desarrollo del Software

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software.(5) Las metodologías de desarrollo se clasifican en dos grupos; las tradicionales o robustas y las ágiles, a continuación se explican en qué consiste cada una de ellas:

1.3.1.1. Metodologías robustas o tradicionales

Son aquellas con mayor énfasis en la planificación y control del proyecto, en especificación precisa de requisitos y modelado. Estas metodologías tradicionales imponen una disciplina de trabajo sobre el proceso de desarrollo del software, con el fin de conseguir un software más eficiente. Para ello, se hace énfasis en la planificación total de todo el trabajo a realizar y una vez que está todo detallado, comienza el ciclo de desarrollo del producto software. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada.(6)

1.3.1.2. Metodologías ágiles

Su objetivo es esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente, respondiendo a los cambios que puedan surgir a lo largo del proyecto. Se pretende ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas.(7)

Principales características de las metodologías ágiles:

- ✓ Se encarga de valorar al individuo y las iteraciones del equipo más que a las herramientas o los procesos utilizados.
- ✓ Se hace mucho más importante crear un producto software que funcione que generar mucha documentación.
- ✓ El cliente está en todo momento colaborando en el proyecto.

1.3.1.2.1. Programación Extrema (eXtreme Programming XP)

Programación Extrema es una metodología ágil, utilizada para proyectos de corta duración, pequeños equipos de desarrolladores y breve tiempo de entrega. Plantea que se trabaje directamente con el cliente, realizando pequeñas iteraciones, supliendo los requisitos y casos de usos que propone RUP (Rational Unified Process o Proceso Racional Unificado) por HU (History of User o Historia de Usuario). También define un estándar de codificación, facilitando que, en vez de que los programadores desarrollen cada uno en su propio estilo, lo hagan sobre uno solo, el que está definido por la metodología, logrando organización e igualdad. Debido a estas características, que se ajustan a las condiciones actuales de los diplomantes y a las siguientes razones, se tomó XP como metodología a utilizar.

- ✓ Los requisitos tienden a cambiar frecuentemente, según avanza el trabajo, el cliente puede agregar nuevas HU, dividir las o simplemente eliminarlas.
- ✓ El cliente forma parte del equipo de desarrollo.
- ✓ Los programadores y el cliente trabajan juntos a lo largo del proyecto.

1.3.2. Herramienta CASE para la modelación del sistema

Las herramientas CASE (Computer Aided Software Engineering o Ingeniería de Software Asistida por Computadora) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costos, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.(8)

1.3.2.1. Visual Paradigm 8.0

Visual Paradigm es una herramienta CASE, proporciona al usuario gran ayuda para el desarrollo de aplicaciones informáticas, desde la planificación, el análisis y diseño, la generación del código fuente de los programas y la documentación.

A continuación se muestran las razones de su elección:

- ✓ Tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la definición de base de datos a partir del esquema de las clases.
- ✓ Se integra con varios IDEs.
- ✓ Disponibilidad en múltiples plataformas, ejemplo Microsoft Windows y Linux.

1.3.3. Lenguaje de Programación

Un Lenguaje de Programación es un conjunto de reglas, notaciones, símbolos y/o caracteres que permiten a un programador poder expresar el procesamiento de datos y sus estructuras en la computadora. Cada lenguaje posee sus propias sintaxis.(9)

1.3.3.1. Java 7

Es un lenguaje potente que trabaja con sus datos como objetos y con interfaces a esos objetos. Soporta las tres características propias del paradigma de la orientación a objetos: encapsulación, herencia y polimorfismo Sus características principales son:

- ✓ **Orientado a objetos:** Diseñado como un lenguaje orientado a objetos. Los objetos agrupan en estructuras encapsuladas tanto sus datos como los métodos que manipulan esos datos
- ✓ **Distribuido:** Proporciona una colección de clases para su uso en aplicaciones de red, que permiten abrir sockets, establecer y aceptar conexiones con servidores o clientes remotos, facilitando así la creación de aplicaciones distribuidas.
- ✓ **Robusto:** Diseñado para crear software altamente fiable. Proporciona numerosas comprobaciones en compilación y en tiempo de ejecución. Sus características de memoria liberan a los programadores de una familia entera de errores, ya que se ha prescindido por completo de los punteros, y la recolección de basura elimina la necesidad de liberación explícita de memoria.
- ✓ **Portable:** El lenguaje fue desarrollado para que las aplicaciones pudiesen ser inmediatamente ejecutables en cualquier ordenador, y en cualquier sistema operativo; esto es posible debido a que Java trabaja con una máquina virtual que genera su propio código máquina. De esta forma con tener una máquina virtual de Java para una determinada plataforma dicho programa será completamente operativo.(10)

1.3.4. Entorno de Desarrollo Integrado

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación; es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes.

1.3.4.1. NetBeansIDE 7.01

Es un entorno de desarrollo integrado (IDE, por sus siglas en inglés), modular, escrito en el lenguaje de programación Java. Consiste en un IDE de código abierto y una plataforma de aplicación que pueden ser usadas como una estructura de soporte general para compilar cualquier tipo de aplicación. Es compatible con Windows, Linux, Mac OS X y Solaris y además es un producto de código abierto, gratuito, sin restricciones de uso.(11)

A continuación se muestran las razones de elección:

- ✓ Rapidez de compilado y en el manejo de errores.
- ✓ Soporta la edición de HTML5.
- ✓ Soporte para la validación de Bean.
- ✓ Ajuste de línea en el Editor.

1.3.5. Servidor de Base de Datos

Un Sistema Gestor de Bases de Datos (SGBD) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos.(12)

1.3.5.1. PostgreSQL 9.1

PostgreSQL es un potente sistema de gestión de bases de datos objeto-relacional multiusuario, centralizado y de propósito general, que está siendo desarrollado desde 1977 y que está liberado bajo la licencia Berkeley Software Distribution (BSD). Fue pionero en muchos conceptos que estuvieron disponibles en algunos sistemas de bases de datos comerciales de alto calibre como por ejemplo: control de acceso simultáneo,

gestión de transacciones, puntos de seguridad; fue uno de los primeros intentos en implementar un motor de bases de datos relacional.(13)

Características:

- ✓ Soporta distintos tipos de datos:
- ✓ Incorpora una estructura de datos array.
- ✓ Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc.
- ✓ Permite la declaración de funciones propias, así como la definición de disparadores.
- ✓ Soporta el uso de índices, reglas y vistas.
- ✓ Incluye herencia entre tablas
- ✓ Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.

Ventajas:

- ✓ Instalación ilimitada.
- ✓ Mejor soporte que los proveedores comerciales.
- ✓ Ahorros considerables en costos de operación.
- ✓ Estabilidad y confiabilidad legendarias.
- ✓ Extensible.
- ✓ Multiplataforma.
- ✓ Herramientas gráficas de diseño y administración de bases de datos.

1.3.6. Tecnología del lado del servidor

1.3.6.1. Spring Framework 3.0

Spring Framework es una plataforma que nos proporciona una infraestructura que actúa de soporte para desarrollar aplicaciones Java. Se ha popularizado en la comunidad de programadores en Java al ser considerado como una alternativa del modelo de Enterprise JavaBean. Por su diseño el framework ofrece mucha libertad a los desarrolladores en Java y soluciones muy bien documentadas y fáciles de usar para las prácticas comunes en la industria. Es potente en cuanto a la gestión del ciclo de vida de los componentes y fácilmente ampliable.

Spring proporciona:

- ✓ Una potente gestión de configuración basada en JavaBeans, aplicando los principios de Inversión de Control. Esto hace que la configuración de aplicaciones sea rápida y sencilla.
- ✓ Una capa de abstracción JDBC³ que ofrece una significativa jerarquía de excepciones, simplifica el manejo de errores, y reduce considerablemente la cantidad de código necesario.
- ✓ Instalaciones de prueba avanzadas para pruebas unitarias, así como para las pruebas de integración.(14)

1.3.6.2. Hibernate 3.0

Hibernate es una herramienta para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML⁴) que permiten establecer estas relaciones. Hibernate proporciona además un lenguaje para el manejo de consultas a la base de datos. Este lenguaje es similar a SQL⁵ y es utilizado para obtener objetos de la base de datos según las condiciones especificadas en el HQL (Hibernate Query Language). El uso de HQL permite usar un lenguaje intermedio que según la base de datos que use y el dialecto que se especifique será traducido al SQL dependiente de cada base de datos de forma automática y transparente.(15)

1.3.7. Tecnología del lado del cliente

1.3.7.1. JQuery 1.10

JQuery es una biblioteca de JavaScript que permite simplificar la manera de interactuar con las vistas en una aplicación web, permitiendo manejar eventos y desarrollar animaciones a las páginas web. Además, permite el intercambio asíncrono de datos entre cliente y servidor de manera sencilla. Esta biblioteca incluye las funcionalidades de DOM⁶, se puede manipular el estilo CSS⁷ de las páginas o elementos HTML(16)

³Java Database Connectivity.

⁴eXtensible Markup Language.

⁵*Structured Query Language.*

⁶Document Object Model.

⁷Cascading Style Sheets.

1.3.7.2. Apache Tomcat 7.0.22

Apache Tomcat es un servidor web con soporte de servlets y JSPs (del inglés: Java Server Pages). Es desarrollado en un entorno abierto y participativo y publicado bajo la licencia Apache versión 2.0. Es la intención de ser una colaboración de los mejores desarrolladores de su clase de todo el mundo.(17)

Características:

- ✓ Es gratis, fácil de instalar, es compatible con las APIs (*Application Programming Interface*) más recientes de Java, ocupa muy poco espacio y es rápido.
- ✓ Es fiable, su solidez se basa en que miles de desarrolladores contribuyen con su código.
- ✓ Tomcat tiene seguridad de nivel de aplicación a partir de la versión 4.0.

1.4. Conclusiones parciales

Con la realización de este capítulo se demuestra la necesidad de un sistema que permita gestionar todo lo relacionado con el control de laboratorios en el Centro de Telemática de la Facultad 2 de la UCI, descartando aquellas posibilidades que por incompletas u orientadas a otros procesos o por necesitar de una tecnología más avanzada que la que se posee no pueden constituir la solución definitiva. Se seleccionaron además las herramientas y la metodología de desarrollo que constituían la opción más viable para dar solución a dicha aplicación.

Capítulo 2: Características y Propuesta del Sistema.

2.1. Introducción

En el presente capítulo se enfatizan las principales características del sistema a desarrollar. Se identifican los procesos que intervienen en el Sistema de Control de los Laboratorios con el objetivo de desarrollar un correcto diseño e implementación de los mismos. Se definen las historias de usuarios propias de la metodología seleccionada en la fase de planeación, así como los requerimientos no funcionales y se precisan los actores que intervienen en el sistema.

2.2. Propuesta del Sistema

La solución propuesta es una aplicación web, la cual estará diseñada para el Centro Telemática el cual radica en la Facultad 2, para llevar el control de los laboratorios.

El sistema proporcionará una interfaz como forma de presentación al usuario desde la cual se podrá autenticar. Una vez autenticada la persona, la aplicación ofrecerá información general sobre el sistema y le proporcionará de acuerdo a su rol una serie de funciones.

Para ello el sistema contará con las siguientes funcionalidades:

Autenticación de usuarios, control de las personas pertenecientes a cada laboratorio, control de los medios, generación de reportes y asignación y evaluación de las tareas a los técnicos.

Autenticación y Gestión de Usuarios: El sistema contará con una interfaz para validar la entrada al sistema, llenando los campos: usuario y contraseña del dominio uci, una vez autenticado el usuario puede acceder a las funcionalidades que tenga acceso según su rol.

Control de las personas: El sistema permitirá mostrar a la persona que controla la existencia de los usuarios todos los datos. El usuario debe mostrar su solapín para verificar en el sistema si pertenece al laboratorio.

Control de Medios: El sistema debe registrar todos los datos referentes a los medios con que cuentan los laboratorios y las características de los mismos.

Generación de reportes: A partir de la información de los medios que cuentan los laboratorios el sistema le permitirá a los usuarios que tengan permiso gestionar los reportes de afectación de los medios. Esta información será guardada en la base de datos para pueda ser visualizado cuando se desee.

Gestión y evaluación de las tareas: El sistema le permitirá a los usuarios con los permisos necesarios para esta funcionalidad gestionar las tareas con los datos necesarios para ser asignadas a los técnicos y en correspondencia con este procedimiento el sistema debe de ser capaz de permitir a los usuarios autorizados darle una evaluación a la tarea.

Para el trabajo con el sistema se han definido tres roles: Administrador, Técnico y Usuario. El Administrador tendrá acceso a todas las funcionalidades del sistema. Además podrá restringirles a los usuarios el acceso a las funcionalidades de acuerdo a su rol. El Técnico va a registrar la entrada de los usuarios a los laboratorios, gestionar los medios, modificar el estado de las tareas y hacer reportes de los medios. Los Usuarios van a poder visualizar toda la información existente en el sistema, así como podrán modificar el proyecto al que pertenecen. La información de todos los usuarios que interactúen con el sistema se registrará y controlará en la base de datos de este.

2.3. Lista de reserva del producto

Historias de Usuarios.

- HU1.** Autenticar Usuario.
- HU2.** Cerrar Sesión.
- HU3.** Insertar usuario.
- HU4.** Modificar usuario.
- HU5.** Buscar usuario.
- HU6.** Eliminar usuario.
- HU7.** Insertar medio.
- HU8.** Modificar medio.
- HU9.** Buscar medio.
- HU10.** Eliminar medio.
- HU11.** Insertar laboratorio.
- HU12.** Modificar laboratorio.
- HU13.** Buscar laboratorio.
- HU14.** Eliminar laboratorio.

- HU15.** Insertar proyecto.
- HU16.** Modificar proyecto.
- HU17.** Buscar proyecto.
- HU18.** Eliminar proyecto.
- HU19.** Insertar reportes de afectación de los medios.
- HU20.** Modificar reportes de afectación de los medios.
- HU21.** Buscar reportes de afectación de los medios.
- HU22.** Eliminar reportes de afectación de los medios.
- HU23.** Generar reportes de afectación de los medios.
- HU24.** Generar reportes de estadísticas.
- HU25.** Insertar tareas de los técnicos.
- HU26.** Modificar tareas de los técnicos.
- HU27.** Buscar tareas de los técnicos.
- HU28.** Eliminar tareas de los técnicos.
- HU29.** Evaluar a los técnicos por las tareas de desempeño.

2.4. Requisitos no funcionales del sistema

Para lograr un correcto funcionamiento del sistema se deben tener en cuenta un conjunto de requisitos no funcionales. Los requisitos no funcionales son cualidades o propiedades que el producto debe tener. Estos se dividen en varias categorías:

Requisitos de usabilidad

- El sistema podrá ser utilizado por cualquier usuario con las siguientes características:
 - Conocimientos básicos relativos al uso de una computadora.
 - Conocimientos básicos del sistema operativo Windows. y Linux
 - Conocimientos sólidos relativos a los procesos de negocio acorde al rol que desempeñe.
 - El sistema se distribuirá en lenguaje español.

Requisitos de seguridad

- El acceso a la aplicación se realizará a través del modelo RBAC (Control de Acceso Basado en Roles), teniendo solo acceso a esta el personal autorizado y solo a las funcionalidades definidas.

Requisitos de soporte

- El sistema permitirá la modificación o la agregación de nuevas funcionalidades cuando sea necesario, asegurando su extensibilidad y lograr mejores prestaciones.

Requisitos de apariencia o interfaz de usuario

- Interfaz Web: La interfaz deberá ser sencilla con tonalidades azules claras y sin cúmulo de imágenes u objetos que distraigan al cliente del objetivo de su empleo.
- Las ventanas del sistema contendrán bien estructurados los datos, y al mismo tiempo permitirán la interpretación correctamente la información.

Requisitos de hardware

- Para explotación del cliente: PC Pentium 3 o superior, CPU 133 MHZ o superior, 256 RAM mínimo 512 RAM recomendada o superior.
- Para explotación del servidor: CPU Dual Core 2.0 GHZ o superior, memoria RAM de 1 GB, 5 GB de disco duro o superior.

Requisitos de software

- En el servidor sobre el cual se instalará la aplicación debe contar con un Sistema Operativo Windows o Linux.
- El servidor deberá contar además con la instalación de Gestor de Bases de Datos PostgreSQL y el servidor web Apache Tomcat.

2.5. Planeación

La fase de planeación es la primera fase definida por la metodología XP. La actividad de planeación comienza con la creación de las historias de usuarios, las cuales describen las características y funcionalidades requeridas para el software que se construirá.

2.5.1. Involucrados en el Sistema

Se definen como involucrados en el sistema todos aquellos que realizan una función o interactúan con él de una forma u otra.

Involucrados en el Sistema	Justificación
----------------------------	---------------

Administrador	Son las personas facultadas para la gestión del sistema en general. Podrán gestionar toda la información.
Técnico	Es la persona que interactúa directamente con el sistema teniendo permiso para realizar el control del personal perteneciente a cada laboratorio, visualizar y modificar las tareas y la generación de reportes de afectación de los medios.
Usuario	Son las personas que interactúan con el sistema teniendo permiso para visualizar todos los datos y modificarse ellos mismo.

Tabla 0.1: Involucrados en el Sistema.

2.5.2. Historias de Usuario

Las historias de usuario son similares a las descripciones de casos de uso en RUP, con la diferencia de que son descritas por el propio cliente en dependencia de las necesidades del sistema.

Las historias de usuarios solo proporcionan una breve descripción de las mismas, los detalles sobre la estimación de riesgo y el tiempo que llevaría su implementación. El cliente es el encargado de darle una prioridad a cada historia de usuario y el equipo de desarrollo es quien evalúa las historias y le da un costo, este se traduce en las semanas que se llevará el desarrollo de la misma.

Campos de las historias de usuarios.

A las historias de usuario se les asignan una prioridad del negocio y un riesgo de desarrollo, estos pueden ser clasificados como se muestra a continuación:

Prioridad en negocio: Prioridad que se le asigna a la historia de usuario en el negocio. Si es importante debe ser implementada lo antes posible, esta se clasifica en:

- ✓ **Alta:** Se le otorga a las historias de usuarios que resultan funcionalidades fundamentales en el desarrollo del sistema, las que el cliente define como principales para el control integral del sistema.

- ✓ **Media:** Se le otorga a las historias de usuarios que resultan para el cliente como funcionalidades a tener en cuenta, sin que estas tengan una afectación sobre el sistema que se esté desarrollando.
- ✓ **Baja:** Se le otorga a las historias de usuarios que constituyen funcionalidades que sirven de ayuda al control de elementos asociados al equipo de desarrollo, a la estructura y no tienen nada que ver con el sistema en desarrollo.

Riesgo en desarrollo: Riesgo que representa para el desarrollo la historia de usuario. En dependencia del riesgo que represente se le asigna un valor que puede ser:

- ✓ **Alta:** Cuando en la implementación de las historias de usuarios se considera la posible existencia de errores que lleven a la inoperatividad del código.
- ✓ **Media:** Cuando pueden aparecer errores en la implementación de las historias de usuarios que puedan retrasar la entrega de la versión.
- ✓ **Baja:** Cuando pueden aparecer errores que serán tratados con relativa facilidad, sin que traigan perjuicios para el desarrollo del proyecto.

El cliente y el equipo de desarrollo trabajan en conjunto para definir como agrupar las HU para su lanzamiento. A continuación se muestran las HU de prioridad alta, definidas por el cliente y el equipo de desarrollo:

Historia de Usuario	
Número: 1	Nombre de Historia de Usuario: Autenticar Usuario
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Daliana González Luis Olfrides Pérez Rodríguez	Iteración Asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 1
Riesgo en Desarrollo: Medio	Puntos Reales: 1
Descripción: El usuario entra sus datos (usuario y contraseña) en el formulario, el sistema verifica que los datos entrados estén correctos y que se trate de un usuario con privilegios, en caso correcto se le permite el acceso, en caso contrario se le informa un mensaje de aviso indicándole la negativa del acceso.	
Observaciones: Los usuarios deben tener el mínimo privilegio absoluto necesario para realizar las tareas asignadas.	
Prototipo de interfaz	

Tabla 0.2: Historia de Usuario Autenticar Usuario.

Historia de Usuario	
Número: 2	Nombre de Historia de Usuario: Cerrar sesión
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Daliana González Martínez Luis Olfrides Pérez Rodríguez	Iteración Asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1
Descripción: Permite a los diferentes usuarios cerrar sesión.	
Observaciones:	
Prototipo de interfaz	

Tabla 0.3: Historia de Usuario Cerrar sesión.

Historia de Usuario	
Número: 3	Nombre de Historia de Usuario: Insertar usuario
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Daliana González Martínez Luis Olfrides Pérez Rodríguez	Iteración Asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 1
Riesgo en Desarrollo: Medio	Puntos Reales: 1
Descripción: Permite insertar los usuarios que tendrán acceso al sistema por los siguientes parámetros (Nombre, Apellidos, Categoría, Proyecto).	
Observaciones:	
Prototipo de interfaz	

Tabla 0.4: Historia de Usuario Insertar Usuario.

Historia de Usuario	
Número: 4	Nombre de Historia de Usuario: Modificar usuario
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Daliana González Martínez Luis Olfrides Pérez Rodríguez	Iteración Asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 1

Riesgo en Desarrollo: Medio	Puntos Reales: 1
Descripción: Se selecciona el usuario en el listado, se da clic en el ícono que representa la opción modificar usuario, el sistema muestra una vista donde se visualizan los datos de la persona permitiendo ser modificados. Cuando se modifican los datos el sistema muestra el listado de personas actualizado.	
Observaciones:	
Prototipo de interfaz	

Tabla 0.5: Historia de Usuario Modificar Usuario.

Historia de Usuario	
Número: 5	Nombre de Historia de Usuario: Buscar usuario
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Daliana González Martínez Luis Olfrides Pérez Rodríguez	Iteración Asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1
Descripción: El sistema presenta una vista donde se muestra todas las personas registradas permitiendo realizar la búsqueda de uno o varios usuarios en específico, mostrando un selector que permite filtrar por criterios (Nombre, Apellidos, Rol y Proyecto). Cuando se pulsa el botón buscar el sistema muestra un listado con todos los usuarios que cumplen con los criterios establecidos por el usuario que realiza la búsqueda.	
Observaciones: Si no encuentra resultados de la búsqueda se mostrará un mensaje	
Prototipo de interfaz	

Tabla 0.6: Historia de Usuario Buscar Usuario.

Historia de Usuario	
Número: 6	Nombre de Historia de Usuario: Eliminar usuario
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Daliana González Martínez Luis Olfrides Pérez Rodríguez	Iteración Asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1

Descripción: Se selecciona la persona en el listado, se da clic en el ícono que representa la opción de eliminar, luego de pulsar este ícono el sistema actualiza el listado.
Observaciones:
Prototipo de interfaz

Tabla 0.7 Historia de Usuario: Eliminar Usuario.

Historia de Usuario	
Número: 7	Nombre de Historia de Usuario: Insertar medio
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Daliana González Martínez Luis Olfrides Pérez Rodríguez	Iteración Asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1
Descripción: El sistema muestra una vista con todos los nombres de los campos necesarios para que la persona autorizada pueda insertarlos medios.	
Observaciones:	
Prototipo de interfaz	

Tabla 0.8: Historia de Usuario: Insertar Medio.

Historia de Usuario	
Número: 8	Nombre de Historia de Usuario: Modificar medio
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Daliana González Martínez Luis Olfrides Pérez Rodríguez	Iteración Asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1
Descripción: El sistema muestra una vista con todos los nombres de los campos necesarios para que la persona autorizada pueda modificar los medios.	
Observaciones:	
Prototipo de interfaz	

Tabla 0.9: Historia de Usuario: Modificar Medio.

Historia de Usuario	
Número:9	Nombre de Historia de Usuario: Buscar medio
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Daliana González Martínez Luis Olfrides Pérez Rodríguez	Iteración Asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1
Descripción: El sistema muestra a las personas autorizadas acceder a los datos de los medios para buscarlos según el parámetro, en caso de ser necesario.	
Observaciones:	
Prototipo de interfaz	

Tabla 0.10: Historia de Usuario: Buscar Medio.

Historia de Usuario	
Número:10	Nombre de Historia de Usuario: Eliminar medio
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Daliana González Martínez Luis Olfrides Pérez Rodríguez	Iteración Asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1
Descripción: El sistema muestra a las personas autorizadas acceder a los datos de los medios para eliminarlos, en caso de ser necesario.	
Observaciones:	
Prototipo de interfaz	

Tabla 0.11: Historia de Usuario: Eliminar Medio.

Historia de Usuario	
Número:11	Nombre de Historia de Usuario: Insertar laboratorio
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Daliana González Martínez Luis Olfrides Pérez Rodríguez	Iteración Asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1

Descripción: El sistema muestra una vista con todos los nombres de los campos necesarios para que la persona autorizada pueda insertar los laboratorios.
Observaciones:
Prototipo de interfaz

Tabla 0.12: Historia de Usuario: Insertar Laboratorio.

Historia de Usuario	
Número: 12	Nombre de Historia de Usuario: Modificar laboratorio
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Daliana González Martínez Luis Olfrides Pérez Rodríguez	Iteración Asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1
Descripción: El sistema muestra una vista con todos los nombres de los campos necesarios para que la persona autorizada pueda modificar los medios.	
Observaciones:	
Prototipo de interfaz	

Tabla 0.13: Historia de Usuario: Modificar Laboratorio.

Historia de Usuario	
Número: 13	Nombre de Historia de Usuario: Buscar laboratorio
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Daliana González Martínez Luis Olfrides Pérez Rodríguez	Iteración Asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1
Descripción: El sistema muestra una vista con todos los nombres de los campos necesarios para que la persona autorizada pueda buscar los medios.	
Observaciones:	

Prototipo de interfaz

Tabla 0.14: Historia de Usuario: Buscar Laboratorio.

Historia de Usuario	
Número:14	Nombre de Historia de Usuario: Eliminar laboratorio
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Daliana González Martínez Luis Olfrides Pérez Rodríguez	Iteración Asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1
Descripción: El sistema muestra una vista con todos los nombres de los campos necesarios para que la persona autorizada pueda eliminar los medios.	
Observaciones:	
Prototipo de interfaz	

Tabla 0.15: Historia de Usuario: Eliminar Laboratorio.

Las Historias de Usuario con prioridad media pueden verse en el [Anexo I](#)

2.5.3. Estimación de esfuerzo por HU

Para el desarrollo de la aplicación propuesta se ha realizado una estimación de esfuerzo por cada una de las HU identificadas, a continuación se muestra el resultado obtenido:

Nro.	Historia de Usuario	Puntos de Estimación
1	Autenticar usuario.	1
2	Cerrar Sesión.	1
3	Insertar usuario.	1
4	Modificar usuario.	1

5	Buscar usuario.	1
6	Eliminar usuario.	1
7	Insertar medio.	1
8	Modificar medio.	1
9	Buscar medio.	1
10	Eliminar medio.	1
11	Insertar laboratorio.	1
12	Modificar laboratorio.	1
13	Buscar laboratorio.	1
14	Eliminar laboratorio.	1
15	Insertar proyecto.	1
16	Modificar proyecto.	1
17	Buscar proyecto.	1
18	Eliminar proyecto.	1
19	Insertar reportes de afectación de los medios.	1
20	Modificar reportes de afectación de los medios.	1

21	Buscar reportes de afectación de los medios.	1
22	Eliminar reportes de afectación de los medios.	1
23	Generar reportes de afectación de los medios.	1
24	Generar reportes de estadísticas.	1
25	Insertar tareas a los técnicos.	1
26	Modificar tareas a los técnicos.	1
27	Eliminar tareas a los técnicos.	1
28	Eliminar tareas a los técnicos.	1
29	Evaluar a los técnicos por las tareas de desempeño.	1

Tabla 0.16: Estimación de esfuerzo por historia de usuarios

2.5.4. Plan de Iteraciones.

Para una mayor organización del trabajo como lo plantea el ciclo de vida de XP se crea un plan de duración de las iteraciones, en este caso se realizará un solo plan ya que existe un único equipo de desarrolladores. Este plan se realiza con el objetivo de reflejar cuáles serán las historias de usuario implementadas en cada una de las iteraciones, así como el tiempo destinado a cada una de ellas y el orden en que se implementarán, lo que ayuda a obtener una idea general del tiempo que durará la confección total del sistema.

2.5.4.1. Iteración 1:

En la primera iteración se implementan las historias de usuario con mayor prioridad, obteniendo al final de la misma una primera versión del producto y dando al sistema las primeras funcionalidades, centrándose en la autenticación de usuarios, la gestión de los usuarios, medios y laboratorios.

2.5.4.2. Iteración 2:

Durante esta iteración se implementan las restantes funcionalidades con prioridad alta y media, al terminar esta se habrá implementado lo relacionado con las historias de usuarios 15, 16, 17, 18, 19, 20, 21, 22 y 23, las cuales se relacionan con la gestión reportes de afectaciones de los medios y gestión de reportes estadísticos. La versión del producto referente a esta iteración junto a las implementaciones anteriores, serán mostradas al cliente con el objetivo de realizar cambios sobre la base dependiendo de la opinión del mismo.

2.5.4.3. Iteración 3:

En esta iteración serán implementadas el resto de las funcionalidades, estas hacen alusión a todo lo relacionado con gestión y evaluación de tareas a los técnicos. Estas funciones están descritas en las historias de usuario 24, 25, 26, 27,28y29. Como resultado de esta iteración se tendrá la versión 1.0 del producto final. A partir de este momento el sistema será puesto a prueba por un período de tiempo para evaluar el desempeño del mismo.

Plan de duración de las iteraciones

Iteración	Orden de la Historia de Usuario a implementar	Duración Total
1	Autenticar usuario. Cerrar sesión. Insertar usuario. Modificar usuario. Buscar usuario. Eliminar usuario. Insertar medio. Modificar medio. Buscar medio. Eliminar medio.	5 semanas

	<p>Insertar laboratorio.</p> <p>Modificar laboratorio.</p> <p>Buscar laboratorio.</p> <p>Eliminar laboratorio.</p>	
2	<p>Insertar proyecto.</p> <p>Modificar proyecto.</p> <p>Buscar proyecto.</p> <p>Eliminar proyecto.</p> <p>Insertar reportes de afectación de los medios.</p> <p>Modificar reportes de afectación de los medios.</p> <p>Buscar reportes de afectación de los medios.</p> <p>Eliminar reportes de afectación de los medios.</p> <p>Generar reportes de afectación de los medios.</p> <p>Generar reportes de estadísticas.</p>	4 semanas
3	<p>Insertar tareas a los técnicos.</p> <p>Modificar tareas a los técnicos.</p> <p>Buscar tareas a los técnicos.</p> <p>Eliminar tareas a los técnicos.</p> <p>Evaluar a los técnicos por las tareas de desempeño.</p>	3 semanas

Tabla 0.17: Plan de duración de las iteraciones.

2.6. Conclusiones Parciales.

En este capítulo se describió la propuesta del sistema a desarrollar, definiendo las funcionalidades y características que poseerá el mismo a partir de las historias de usuario identificadas por el cliente. Se determinaron las iteraciones a realizar y las HU que se le darán solución en ellas; Se realizó la planificación del esfuerzo dedicado a la realización de cada una y el orden en que se les dará cumplimiento según las necesidades del cliente. Con lo antes expuesto se le da cumplimiento a las tareas trazada al inicio del capítulo referente a la descripción de los requisitos del sistema.

Capítulo 3: Diseño.

3.1. Introducción

En el presente capítulo se hace alusión a la fase de diseño, propio de la metodología de desarrollo XP utilizada para la implementación del sistema que se propone. También se expone los diferentes artefactos generados durante el transcurso del capítulo y se procede al diseño de la base de datos.

3.2. Patrones de Arquitectura:

Los patrones arquitectónicos, o patrones de arquitectura, son soluciones a problemas de arquitectura de software en ingeniería. Brindan una descripción de los elementos y la relación entre ellos, junto a un conjunto de restricciones sobre cómo pueden ser usados. Un patrón arquitectónico expresa un esquema de organización estructural esencial para un sistema de software.(18)

3.2.1. Patrón arquitectónico: Modelo Vista Controlador (Model View Controller, MVC)

El patrón Modelo-Vista-Controlador es una guía para el diseño de arquitecturas de aplicaciones que ofrezcan una fuerte interactividad con usuarios. Este patrón organiza la aplicación en tres modelos separados, el primero es un modelo que representa los datos de la aplicación y sus reglas de negocio, el segundo es un conjunto de vistas que representa los formularios de entrada y salida de información, el tercero es un conjunto de controladores que procesa las peticiones de los usuarios y controla el flujo de ejecución del sistema.(19)

Ventajas de su Uso

Con la utilización de este patrón se garantiza la separación de responsabilidades entre interfaz, lógica de negocio y de control. La ventaja fundamental de este patrón es la división entre capas que se encuentra centrada en que cada una de estas pueden ser sustituidas sin afectar a las otras ya que provee una separación total entre la lógica del negocio y presentación, además pueden existir diferentes vistas para un mismo modelo trayendo como resultado que la división de código de este estilo arquitectónico haga más fácil la portabilidad y la adaptación a los requerimientos del usuario.

3.3. Patrones de Diseño

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaz. Representan una descripción de las clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular. El uso de patrones posibilita estandarizar el modo en que se realiza el diseño y proporciona reusabilidad, extensibilidad y mantenimiento del código.(20)

A continuación se explican los patrones usados:

3.3.1. Patrones para Asignar Responsabilidades (GRASP)

Los Patrones GRASP (*General Responsibility Assignment Software Patterns*, o Patrones Generales de Software para Asignación de Responsabilidades) describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. A continuación se explican los patrones de asignación de responsabilidades que se implementan en el software a realizar:

Experto: Es el principio básico de asignación de responsabilidades, el cual indica que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. De este modo se obtendrá un diseño con mayor cohesión, la información se mantiene encapsulada y permite contar con un sistema robusto y fácil de mantener.

Creador: El patrón creador ayuda a identificar quién debe ser el responsable de la creación o instanciación de nuevos objetos o clases, de forma tal que una instancia de un objeto sólo pueda ser creada por el objeto que contiene la información necesaria para ello.

Alta cohesión: Se aplica para realizar un diseño que evite contener clases con un alto grado de abstracción, que asuman responsabilidades que podían haber delegado a otros objetos o que tengan responsabilidades muy complejas y este plantea que la información que almacena una clase debe de ser coherente y debe estar (en la medida de lo posible) relacionada con la clase.

Bajo Acoplamiento: Es la idea de tener las clases lo menos ligadas entre sí como sea posible. 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, potenciando la reutilización, y disminuyendo la dependencia entre las clases.

Controlador: El patrón controlador sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, este sugiere que la lógica de negocios debe

estar separada de la capa de presentación, esto para aumentar la reutilización de código y a la vez tener un mayor control.

3.3.2. Inversión de Control (Ioc)

La Inversión de Control es un patrón de diseño pensado para permitir un menor acoplamiento entre componentes de una aplicación y fomentar así el reúso de los mismos.

El patrón Inversión de Control se puede utilizar cuando:

- Se desee desacoplar las clases de sus dependencias de manera de que las mismas puedan ser reemplazadas o actualizadas con muy pocos o casi ningún cambio en el código fuente de sus clases.
- Desea escribir clases que dependan de clases cuyas implementaciones no son conocidas en tiempo de compilación.
- Desea testar las clases aisladamente sin sus dependencias.
- Desea desacoplar sus clases de ser responsables de localizar y gestionar el tiempo de vida de sus dependencias (21)

3.3.3. Inyección de Dependencia

La inyección de dependencias (DI) procede de un patrón de diseño más general que es el patrón de Inversión de Control (IoC). El patrón inyección de dependencia, es una técnica que como su nombre lo indica busca facilitar la resolución de dependencias entre objetos. Consiste en resolver las dependencias de cada clase (atributos) generando los objetos cuando se arranca la aplicación y luego inyectarlos en los demás objetos que los necesiten a través de métodos set o bien a través del constructor, pero estos objetos se instancian una vez, se guardan en una factoría y se comparten por todos los usuarios.(22)

3.4. Estilo Arquitectónico

En la construcción de la aplicación se utiliza el estilo arquitectónico cliente/servidor. Este estilo define una relación entre dos aplicaciones en las cuales una de ellas (cliente) envía peticiones a la otra (servidor fuente de datos). De esta manera, cuando el cliente envía un mensaje solicitando un determinado servicio a un servidor, estos últimos responden a la demanda del cliente que la produjo, trayendo consigo los siguientes beneficios:

- Más seguridad ya que los datos almacenados en el servidor que genera más control sobre la seguridad.
- Acceso centralizado a los datos que están el servidor lo que facilita su acceso y actualización.
- Facilita el mantenimiento.(23)

3.4.1. Componentes del estilo arquitectónico Cliente/Servidor

Los elementos fundamentales sobre los cuales se desarrollan e implantan los sistemas cliente/servidor son:

- El proceso cliente que es quien inicia el diálogo.
- El proceso servidor que pasivamente espera a que lleguen peticiones de servicio.

3.4.2. Elementos principales

Entre los elementos principales que componen el estilo están:

Cliente

Es el proceso que permite al usuario formular los requerimientos y pasarlos al servidor.

Las funciones que lleva a cabo el proceso cliente se resumen en los siguientes puntos:

- Administrar la interfaz de usuario.
- Interactuar con el usuario.
- Procesar la lógica de la aplicación y hacer validaciones locales.
- Generar requerimientos de base de datos.
- Recibir resultados del servidor.
- Formatear resultados.

Servidor

Es el proceso encargado de atender a múltiples clientes que hacen peticiones de algún recurso administrado por él. Maneja todas las funciones relacionadas con la mayoría de las reglas del negocio y los recursos de datos. Las principales funciones que lleva a cabo el proceso servidor son las siguientes:

- Aceptar los requerimientos de base de datos que hacen los clientes.
- Procesar requerimientos de base de datos.
- Formatear datos para transmitirlos a los clientes.

- Procesar la lógica de la aplicación y realizar validaciones a nivel de base de datos.

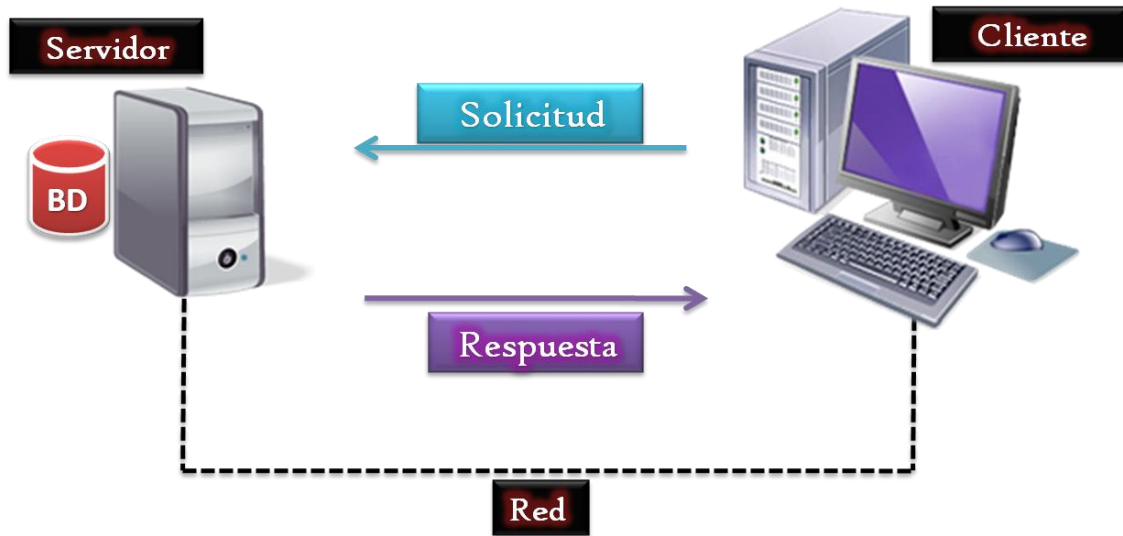


Figura 2: Estilo arquitectónico cliente/servidor

3.5. Arquitectura del Sistema

3.5.1. Arquitectura N-Capas

La arquitectura N-Capas es un estilo de programación donde el objetivo principal es separar los diferentes aspectos del desarrollo, tales como las cuestiones de presentación, lógica de negocio y mecanismo de almacenamiento.

Una de las ventajas principales de esta arquitectura es que el desarrollo se puede utilizar en varios niveles, de este modo, cada grupo de trabajo está totalmente abstraído del resto de los niveles. El diseño más utilizado actualmente es el conocido como diseño en tres capas.

1. **Capa de Presentación:** Es la capa encargada de presentar el sistema al usuario, le comunica la información y captura a su vez la información introducida. Es conocida además como interfaz gráfica y se comunica con la capa de negocio.
2. **Capa de Negocio:** En ella residen los programas que se ejecutan, se encarga en recibir las peticiones de los usuarios y envía las respuestas tras el proceso. Aquí es donde se establecen todas las reglas que deben cumplirse. Recibe las solicitudes de la capa de presentación y presenta los resultados a la misma. Esta capa se comunica además con la capa de datos.

3. **Capa de Acceso a Datos:** Es la encargada de almacenar los datos y acceder a los mismos. Recibe solicitudes de almacenamiento o recuperación de información desde la capa de negocio.



Figura 3: Arquitectura N-Capas

3.6. Diseño de la Base de Datos

La construcción de la base de datos es una de las tareas principales en el diseño de una aplicación, en esta se ponen de manifiesto los datos necesarios para el correcto funcionamiento de la misma. A continuación se muestra el DER (*Entity Relationship Diagrams* o Diagrama Entidad Relación) diseñado para el sistema de control de laboratorios del Centro TLM.

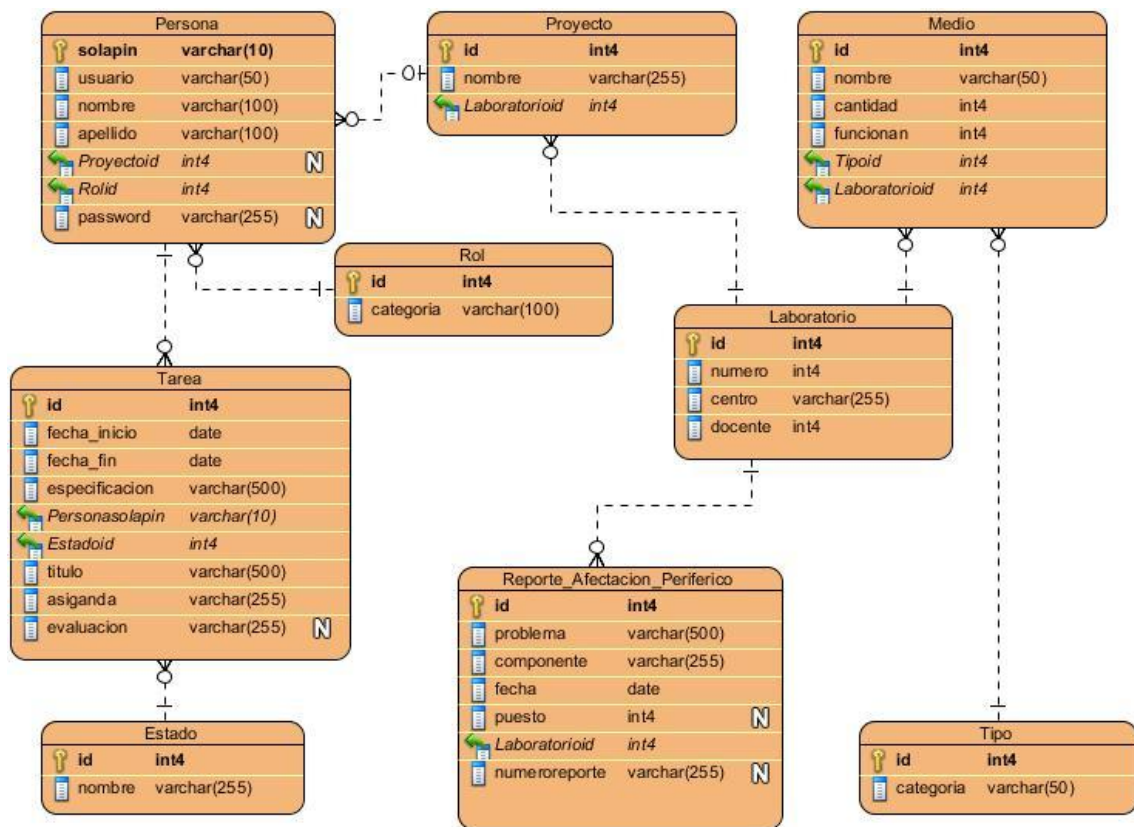


Figura 4: Modelo Entidad Relación

3.7. Tarjetas Clases-Responsabilidad-Colaborador

Las tarjetas CRC constituyen una forma simple de organizar las clases más notables para las funcionalidades del sistema, con el objetivo de desarrollar una representación organizada de las clases.

Un modelo CRC es una colección de tarjetas índices estándar que representan clases. Ellas se dividen en tres secciones. A lo largo del borde superior se escribe el nombre de la clase; en el cuerpo, a la izquierda se listan las responsabilidades de la clase que no es más que lo que la clase sabe o hace y a la derecha los colaboradores que son aquellas clases que se requieren para que una clase reciba la información necesaria para completar una responsabilidad.(24)

Clase:IDAO_Persona		
Responsabilidad		Colaboración
Adicionar_Persona (solapin, usuario, nombre, apellido)	solapin, rol,	IIDAO_Persona,IIN_Persona,IN_Persona,Add_Usuario

Actualizar_Persona(solapin, nombre, proyecto)	IIDAO_Persona, IIN_Persona, IN_Persona, Modificar_Usuario
Obtener_Persona(id_Persona)	IIDAO_Persona, IIN_Persona, IN_Persona, Mostrar_Usuario
Eliminar_Persona(solapin)	IIDAO_Persona, IIN_Persona, IN_Persona, Eliminar_Usuario

Tabla 0.1: Tarjetas CRC Clase: Usuario.

Clase: IDAO_Medio	
Responsabilidad	Colaboración
Adicionar_Medio(medio, lab, docente, nombre, cantidad, funcionan)	IIDAO_Medio, ReporteAfectacionPeriferico, Proyecto, Medio, ReporteEstadistico, IIN_Medio, IN_Medio, Add_Medio
Actualizar_Medio(medio, lab, docente, nombre, cantidad, funcionan)	IIDAO_Medio, ReporteAfectacionPeriferico, Proyecto, Medio, ReporteEstadistico, IIN_Medio, IN_Medio, Modificar_Medio
Obtener_Medio(nombre, lab)	IIDAO_Medio, ReporteAfectacionPeriferico, Proyecto, Medio, ReporteEstadistico, IIN_Medio, IN_Medio, Mostrar_Medio
Eliminar_Medio(lab, docente, medio)	IIDAO_Medio, ReporteAfectacionPeriferico, Proyecto, Medio, ReporteEstadistico, IIN_Medio, IN_Medio, Eliminar_Medio

Tabla 0.2: Tarjetas CRC Clase: Medio.

Clase: IDAO_Laboratorio	
Responsabilidad	Colaboración
Adicionar_Laboratorio(lab, docente, centro)	IIDAO_Laboratorio, ReporteAfectacionPeriferico, Proyecto, Medio, ReporteEstadistico, IIN_Laboratorio, IN_Laboratorio, Add_Laboratorio
Actualizar_Laboratorio(lab, docente, centro)	IIDAO_Laboratorio, ReporteAfectacionPeriferico, Proyecto, Medio, ReporteEstadistico, IIN_Laboratorio, IN_Laboratorio, Modificar_Laboratorio
Listar_Laboratorio()	IIDAO_Laboratorio, ReporteAfectacionPeriferico, Proyecto, Medio, ReporteEstadistico, IIN_Laboratorio, IN_Laboratorio, Mostrar_Laboratorio
Eliminar_Laboratorio(docente, lab)	IIDAO_Laboratorio, ReporteAfectacionPeriferico, Proyecto, Medio, ReporteEstadistico, IIN_Laboratorio, IN_Laboratorio, Eliminar_Laboratorio

Tabla 0.3: Tarjetas CRC Clase: Laboratorio.

Clase:IDAO_Reporte	
Responsabilidad	Colaboración
Adicionar_Reporte_Afectacion(laboratorio,docente,problema,componente,fecha,puesto)	IIDAO_Reporte,IIN_Reporte,IN_Reporte_Afectaciones_Perifericos,IN_Laboratorio,Add_Tarea
Actualizar_Reporte_Afectacion(laboratorio,docente,problema,componente, fecha, puesto)	IIDAO_Reporte,IIN_Reporte,IN_Reporte_Afectaciones_Perifericos,IN_Laboratorio ,Modificar_Reporte
Obtener_Reporte_Afectacion(id_re p_afect)	IIDAO_Reporte,IIN_Reporte,IN_Reporte_Afectaciones_Perifericos,IN_Laboratorio,Modificar_Reporte,Mostrar_Reporte
Eliminar_Reporte_Afectacion(componente,lab, doc, puesto)	IIDAO_Reporte,IIN_Reporte,IN_Reporte_Afectaciones_Perifericos,IN_Laboratorio,Modificar_Reporte,Eliminar_Reporte

Tabla 0.4: Tarjetas CRC Clase: Reporte.

Clase:IDAO_Tarea	
Responsabilidad	Colaboración
Adicionar_Tarea(solapin, fecha_inicio, fecha_fin,especificaciones)	IIDAO_Tarea,IIN_Tarea,IN_Tarea,Add_Tarea
Actualizar_Tarea(id, solapin, fecha_inicio,fecha_fin, especificaciones, estado)	IIDAO_Tarea,IIN_Tarea,IN_Tarea,Modificar_Tarea
Tarea Obtener_Tarea(id_Tarea)	IIDAO_Tarea,IIN_Tarea,IN_Tarea,Mostrar_Tarea
Eliminar_Tarea(id, solapin)	IIDAO_Tarea,IIN_Tarea,IN_Tarea,Eliminar_Tarea

Tabla 0.5: Tarjetas CRC Clase: Tarea.

3.8. Conclusiones Parciales

En este capítulo fueron analizados los patrones de diseño asociados a la arquitectura propuesta, los cuales darán mayor independencia a las clases y facilitarán la implementación. Se realizaron las tarjetas CRC para obtener una representación de las principales clases y funcionalidades del sistema. Como parte del diseño del sistema se realizó el diagrama entidad relación de la base de datos, permitiendo de esta forma almacenar los datos necesarios. Por lo antes planteado se le dio cumplimiento en el transcurso del capítulo a las tareas trazado referente al diseño e implementación del sistema.

Capítulo 4: Implementación y Prueba.

4.1. Introducción

En el presente capítulo se confeccionarán las Tareas de Ingeniería las cuales han sido utilizadas como base para la implementación del software. Luego se evaluará la calidad de la aplicación a través de las pruebas unitarias y las de aceptación, derivadas de las HU y Tareas de la Ingeniería que se han implementado.

4.2. Fase de Implementación

En esta fase se plantea la implementación de las historias de usuario en su correspondiente iteración, obteniéndose en cada una de ellas una versión funcional del producto. Para ello se descomponen las historias de usuario en tareas de desarrollo para ser realizadas, describiendo cada una de estas. Como parte de la planificación realizada en el capítulo anterior se detallan a continuación las iteraciones y las historias de usuario implementadas en cada iteración.

4.2.1. Iteración 1

Historia de Usuario	Estimación
Autenticar usuario.	1
Cerrar sesión.	1
Insertar usuario.	1
Modificar usuario.	1
Buscar usuario.	1
Eliminar usuario.	1
Insertar medio.	1
Modificar medio.	1
Buscar medio.	1

Eliminar medio.	1
Insertar laboratorio.	1
Modificar laboratorio.	1
Buscar laboratorio.	1
Eliminar laboratorio.	1

Tabla 0.1: Historias de Usuarios de la iteración 1.

4.2.1.1. Tareas de Ingeniería de las historias de usuario implementadas en iteración 1.

Tareas de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: 1
Nombre de Tarea: Autenticar Usuario.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Programador Responsable: Daliana González Martínez, Luis Olfrides Pérez Rodríguez.	
Descripción: Mostrar un formulario con los datos de inicio de sección (usuario y contraseña) y el botón aceptar, para acceder al sistema.	

Tabla 0.2: Tarea de Ingeniería # 1: Autenticar Usuario.

Tareas de Ingeniería	
Número Tarea: 2	Número Historia de Usuario: 2
Nombre de Tarea: Cerrar sesión.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Programador Responsable: Daliana González Martínez, Luis Olfrides Pérez Rodríguez.	

Descripción: Mostrar un formulario para cerrar sesión una vez que el usuario esté autenticado.

Tabla 0.3: Tarea de Ingeniería # 2: Cerrar Sesión.

Tareas de Ingeniería	
Número Tarea: 3	Número Historia de Usuario: 3
Nombre de Tarea: Insertar usuario.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Programador Responsable: Daliana González Martínez, Luis Olfrides Pérez Rodríguez.	
Descripción: Mostrar los siguientes datos para la inserción de usuarios: <ul style="list-style-type: none"> • Nombre • Apellidos. • Número de Solapín. • Usuario. 	

Tabla 0.4: Tarea de Ingeniería # 3: Insertar Usuario.

Tareas de Ingeniería	
Número Tarea: 4	Número Historia de Usuario: 4
Nombre de Tarea: Modificar usuario.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Programador Responsable: Daliana González Martínez, Luis Olfrides Pérez Rodríguez.	
Descripción: Mostrar los siguientes datos para la modificación de usuarios: <ul style="list-style-type: none"> • Rol. • Proyecto. 	

Tabla 0.5: Tarea de Ingeniería # 4: Modificar Usuario.

Tareas de Ingeniería

Número Tarea: 5	Número Historia de Usuario: 5
Nombre de Tarea: Buscar usuario.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Programador Responsable: Daliana González Martínez, Luis Olfrides Pérez Rodríguez.	
Descripción: Mostrar los siguientes parámetros para la búsqueda de usuarios: <ul style="list-style-type: none"> • Nombre y Apellidos. • Núm. de Solapín. • Rol. • Proyecto. • Laboratorio. • Docente 	

Tabla 0.6: Tarea de Ingeniería # 5: Buscar Usuario.

Tareas de Ingeniería	
Número Tarea: 6	Número Historia de Usuario: 6
Nombre de Tarea: Eliminar usuarios.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Programador Responsable: Daliana González Martínez, Luis Olfrides Pérez Rodríguez.	
Descripción: Mostrar los datos para la eliminación de usuarios.	

Tabla 0.7: Tarea de Ingeniería # 6: Eliminar Usuario.

Tareas de Ingeniería	
Número Tarea: 7	Número Historia de Usuario: 7
Nombre de Tarea: Insertar medios.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1

Programador Responsable: Daliana González Martínez, Luis Olfrides Pérez Rodríguez.
Descripción: Mostrar una interfaz con los datos requeridos para la inserción de medios.

Tabla 0.8: Tarea de Ingeniería # 7: Insertar Medio.

Tareas de Ingeniería	
Número Tarea: 8	Número Historia de Usuario: 8
Nombre de Tarea: Modificar medio.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Programador Responsable: Daliana González Martínez, Luis Olfrides Pérez Rodríguez	
Descripción: Mostrar una interfaz con los datos requeridos para la modificación de medios.	

Tabla 0.9: Tarea de Ingeniería # 8: Modificar Medio.

Tareas de Ingeniería	
Número Tarea: 9	Número Historia de Usuario: 9
Nombre de Tarea: Buscar medio.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Programador Responsable: Daliana González Martínez, Luis Olfrides Pérez Rodríguez.	
Descripción: Mostrar una interfaz con los parámetros requeridos para la búsqueda de medios.	

Tabla 0.10: Tarea de Ingeniería # 9: Buscar Medio.

Tareas de Ingeniería

Número Tarea: 10	Número Historia de Usuario: 10
Nombre de Tarea: Eliminar medio.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Programador Responsable: Daliana González Martínez, Luis Olfrides Pérez Rodríguez.	
Descripción: Mostrar los datos para la eliminación de los medios.	

Tabla 0.11: Tarea de Ingeniería # 10: Eliminar Medio.

Tareas de Ingeniería	
Número Tarea: 11	Número Historia de Usuario: 11
Nombre de Tarea: Insertar laboratorio.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Programador Responsable: Daliana González Martínez, Luis Olfrides Pérez Rodríguez.	
Descripción: Mostrar una interfaz con los datos requeridos para la inserción de Laboratorios.	

Tabla 0.12: Tarea de Ingeniería # 11: Insertar Laboratorio.

Tareas de Ingeniería	
Número Tarea: 12	Número Historia de Usuario: 12
Nombre de Tarea: Modificar laboratorio.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Programador Responsable: Daliana González Martínez, Luis Olfrides Pérez Rodríguez.	
Descripción: Mostrar una interfaz con los datos requeridos para la modificación de Laboratorios.	

Tabla 0.13: Tarea de Ingeniería # 12: Modificar Laboratorio.

Tareas de Ingeniería	
Número Tarea: 13	Número Historia de Usuario: 13
Nombre de Tarea: Buscar laboratorio.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Programador Responsable: Daliana González Martínez, Luis Olfrides Pérez Rodríguez.	
Descripción: Mostrar una interfaz con los parámetros requeridos para la búsqueda de Laboratorios.	

Tabla 0.14: Tarea de Ingeniería # 13: Buscar Laboratorio.

Tareas de Ingeniería	
Número Tarea: 14	Número Historia de Usuario: 14
Nombre de Tarea: Eliminar laboratorio.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Programador Responsable: Daliana González Martínez, Luis Olfrides Pérez Rodríguez.	
Descripción: Mostrar los datos para la eliminación del Laboratorios.	

Tabla 0.15: Tarea de Ingeniería # 14: Eliminar Laboratorio.

4.2.2. Iteración 2

Historia de Usuario	Estimación
Insertar proyecto.	1
Modificar proyecto.	1
Buscar proyecto.	1
Eliminar proyecto.	1

Insertar reportes de afectación de los medios.	1
Modificar reportes de afectación de los medios.	1
Buscar reportes de afectación de los medios.	1
Eliminar reportes de afectación de los medios.	1
Generar reportes de afectación de los medios.	1
Generar reportes de estadísticas.	1

Tabla 0.16: Historias de Usuarios de la iteración 2.

[Anexo II: Tareas de Ingeniería generadas en la Iteración 2.](#)

4.2.3. Iteración 3

Historia de Usuario	Estimación
Insertar tareas a los técnicos.	1
Modificar tareas a los técnicos.	1
Buscar tareas a los técnicos.	1
Eliminar tareas a los técnicos.	1
Evaluar a los técnicos por las tareas de desempeño.	1

Tabla 0.17: Historias de Usuarios de la iteración 3.

[Anexo III: Tareas de Ingeniería generadas en la Iteración 3.](#)

4.3. Diagrama de Despliegue

Los Diagramas de Despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación.(25)

La configuración propuesta estará conformada por un Servidor Web. Este a su vez se conectará con el servidor LDAP ⁸ para verificar la autenticidad del usuario, posteriormente se conectará con la Base de Datos para almacenar toda la información del sistema. Las máquinas clientes a su vez estarán conectadas con el Servidor Web.

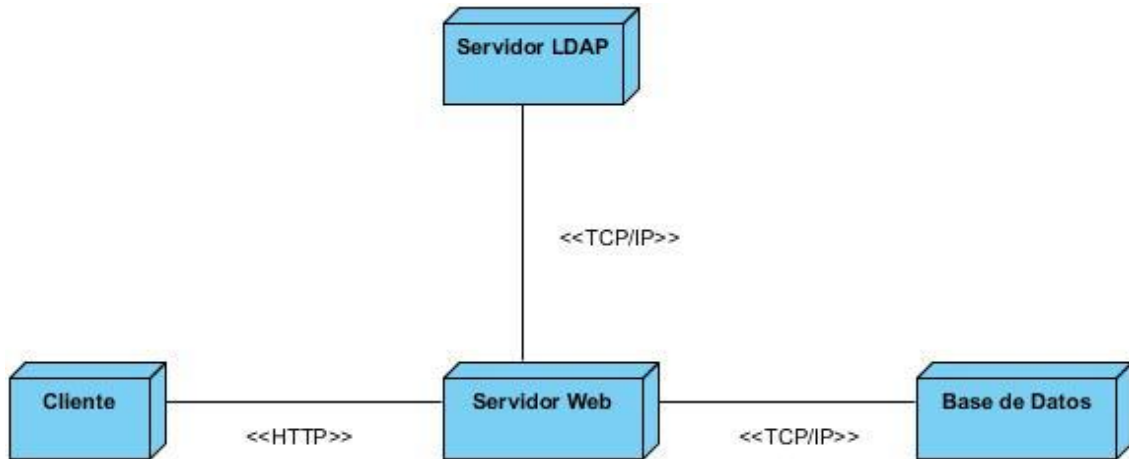


Figura 5: Diagrama de Despliegue

4.4. Pruebas

Las pruebas permiten comprobar la eficacia del sistema; estas son las responsables de verificar si los objetivos trazados fueron cumplidos en la etapa de implementación. Con ellas se reduce el número de errores no detectados durante la implementación, el tiempo entre la introducción de estos en el sistema y su detección; son las encargadas de aumentar la seguridad y de evitar efectos colaterales no deseados a la hora de realizar modificaciones en la aplicación.

4.4.1. Pruebas Unitarias

Son las encargadas de verificar el código y son diseñadas por los programadores. Cada uno de los desarrolladores tiene que ir probando constantemente lo que va obteniendo en el transcurso de la implementación del sistema, para garantizar que las funcionalidades exigidas por el cliente se estén implementando correctamente.

Las pruebas unitarias se realizaron cada vez que se terminaba de implementar una funcionalidad probándola directamente en el entorno real. A continuación se muestra una figura de la prueba unitaria realizada a la clase IIN_Proyecto.

⁸Lightweight Directory Access Protocol

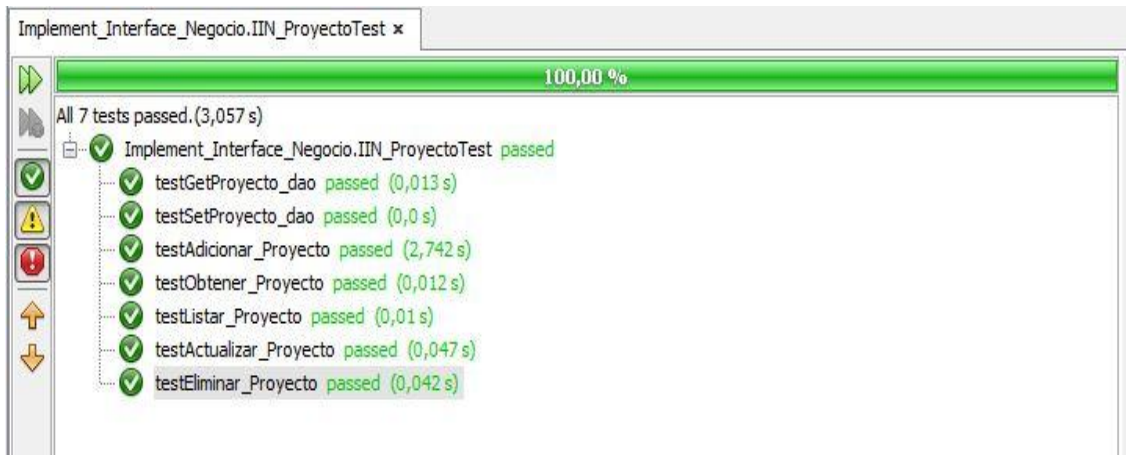


Figura 6: Prueba Unitaria a la clase IIN_Proyecto

4.4.2. Pruebas de Aceptación

Como técnica para garantizar que los requerimientos hayan sido cumplidos y que la aplicación es realmente lo que el cliente necesita, además de asegurar su correcto funcionamiento son realizadas las pruebas de aceptación. Las cuales son creadas a partir de las historias de usuario y desde la perspectiva del cliente. Una historia de usuario puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento.

Estas pruebas funcionan como una caja negra, pues cada una de ellas representa una salida esperada del sistema, donde es responsabilidad del cliente verificar la corrección de las pruebas y tomar decisiones acerca de las mismas.

El objetivo final es lograr que los requerimientos sean cumplidos y que el sistema sea aceptable. Una vez que todas las historias de usuario hayan pasado sus pruebas de aceptación se considera entonces terminada la aplicación.

Como resultado de las pruebas de aceptación se obtendrán artefactos descritos en tablas, estas contarán con los siguientes campos:

Código: Servirá como identificador de la prueba realizada, a su vez será sugerente al nombre de la prueba a la que hace referencia.

UH: Tendrá el nombre de la historia de usuario a la que hace referencia la prueba a realizar.

Nombre: Es el nombre que se le da a la prueba a realizar.

Descripción: Se describe la funcionalidad que se desea probar.

Condiciones de Ejecución: Mostrará las condiciones que deben cumplirse para poder llevar a cabo el caso de prueba, estas condiciones deben ser satisfechas antes de la ejecución del caso de prueba para que se puedan obtener los resultados esperados.

Entradas/Pasos de Ejecución: Se hará la descripción de cada uno de los pasos seguidos durante el desarrollo de la prueba, se tendrá en cuenta cada una de las entradas que hace el usuario con el objetivo de ver si se obtiene el resultado esperado.

Resultado esperado: Se hará una breve descripción del resultado que se espera obtener con la prueba realizada.

Evaluación de la prueba: Acorde al resultado de la prueba realizada se emitirá una evaluación sobre la misma. Esta evaluación tendrá uno de los tres resultados que a continuación se describen:

- Bien: Cuando el resultado de la prueba es exactamente el esperado por el usuario.
- Parcialmente bien: Cuando el resultado no es completamente el esperado por el cliente o usuario de la aplicación y muestra resultados erróneos o fuera de contexto.
- Mal: Cuando el resultado de la prueba realizada genera un error de codificación en la aplicación o muestra como resultado elementos no deseados o fuera de contexto, trayendo como consecuencia que la funcionalidad requerida por el cliente no tenga resultado, lo que invalida también la UH.

Las pruebas de aceptación se encuentran en el [Anexo IV](#).

4.2.3. Resultados de las pruebas de aceptación

Con la realización de las pruebas de aceptación pertenecientes a cada historia de usuario se obtuvo una lista de NC (No Conformidades). A continuación se muestra el resultado obtenido en la siguiente tabla.

Número de Iteración	Total de NC	Cantidad de NC Resuelta	Cantidad de NC que No Proceden
1	11	11	0

2	7	6	1
---	---	---	---

Tabla 0.18: Resultado de las pruebas de aceptación

4.5. Conclusiones Parciales

En el transcurso de este capítulo se ha mostrado que una historia de usuario es solo un resumen de una funcionalidad que debe tener el sistema, y que los desarrolladores van ampliando sus conocimientos a medida que se va implementando; siendo el cliente quien tiene la idea de que es lo que quiere y que va a comprobar mediante las pruebas de aceptación. Por ello, es más práctico que sea el cliente quien elabore las pruebas. A través de las pruebas realizadas al sistema, se pudo asegurar el correcto funcionamiento de la aplicación desarrollada.

Conclusiones Generales

Con el desarrollo de la investigación se logró cumplir los objetivos propuestos al principio de la misma: Se desarrolló un sistema capaz de gestionar los procesos que se llevan a cabo para el control de los laboratorios en el Centro de Telemática de la UCI, logrando de esta forma cumplir con las necesidades del cliente. A continuación se mencionan los resultados obtenidos durante la investigación:

- Se investigaron los principales sistemas de gestión y control de información, así como posibles soluciones al problema planteado, aunque se llegó a la conclusión de que ninguno cumplía las condiciones necesarias para solucionarlo.
- Se justificaron las herramientas, metodologías y tecnologías que se utilizaron en el desarrollo del trabajo, las cuales posibilitaron la implementación de un sistema que cumpla con los requisitos necesarios para el control de los laboratorios.
- Se realizó la implementación del sistema atendiendo a la necesidad del cliente y se realizaron las pruebas pertinentes para la validación de la solución propuesta arrojando resultados positivos.

Por todo lo antes mencionado se evidencia el cumplimiento de los objetivos propuestos en el presente trabajo de diploma, lo cual conlleva al cumplimiento del objetivo general.

Recomendaciones

Con el resultado del presente trabajo se obtuvo un sistema para la gestión de los procesos relacionados con el control de laboratorios del Centro TLM; que aunque resuelve la problemática planteada inicialmente, puede ser refinado con el objetivo de lograr una mejora en su calidad. A continuación se exponen algunas recomendaciones a tener en cuenta para futuras versiones del producto:

- Extender el sistema con la realización de nuevas funcionalidades.
- Analizar después de un período de tiempo el impacto que ha tenido la aplicación.

Referencias Bibliográficas

1. Universidad de las Ciencias Informáticas. [En línea] UCI, 2012. [Citado el: 16 de 04 de 2013.] <http://www.uci.cu/mision>.
2. DATYS. DATYS Tecnologías y Sistemas. *Control de asistencia con elementos biométricos*. [En línea] 2012. [Citado el: 12 de 06 de 2013.] <http://www.datys.cu/wpinfo/producto.aspx?42>.
3. Adisley Flores Coronado, Imiry's Rovira Prieto. *SISTEMA CONTROL DE ACCESO A COMEDORES EN LA UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS*. Ciudad De La Habana, Cuba : s.n., 2007.
4. Informática(UCI), Universidad de las Ciencias. *PAQUETE DE HERRAMIENTAS PARA LA GESTIÓN DE PROYECTOS*. Cuba : s.n., 2012.
5. Universidad de Murcia. *Ingeniería de Software y Metodología de Desarrollo*. [En línea] 2012. [Citado el: 15 de 06 de 2013.] www.um.es/docencia/barzana/IAGP/IAGP2-Metodologias-de-desarrollo.html.
6. Acuña. *Selección de metodologías de desarrollo para aplicaciones web en la facultad de informática de la Universidad de Cienfuegos*. Cienfuegos, Cuba : s.n., 2009.
7. Letelier, Patricio. *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. 2010.
8. Sub-Jefatura de Informática, Dirección Técnica de Desarrollo Informático. *Herramientas Case*. s.l. : 300 Ejemplares, 1999.
9. Instituto Politécnico de Celaya. [En línea] <http://www.iqcelaya.itc.mx/~vicente/Programacion/Lenguajes.pdf>.
10. Gonzalo Álvarez Marañón. *Características del Lenguaje Java*. [En línea] 1999. [Citado el: 25 de 06 de 2013.] <http://www.iec.csic.es/criptonomicon/java/quesjava.html>.
11. NetBeans. *NetBeans IDE Features*. [En línea] Oracle Corporation, 2013. [Citado el: 16 de 06 de 2013.] <https://netbeans.org/features/index.html>.
12. DesarrolloWeb.com. [En línea] [Citado el: 16 de 06 de 2013.] <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>.
13. PostgreSQL-es . *Portal en Español sobre PostgreSQL*. [En línea] 2009-2012. [Citado el: 17 de 06 de 2013.] http://www.postgresql.org.es/sobre_postgresql.
14. Spring. [En línea] 2013. [Citado el: 15 de 06 de 2013.] <http://www.springsource.org/spring-framework>.
15. Manual de Hibernate. [En línea] 2012. [Citado el: 13 de 06 de 2013.] <http://www.javaHispano.org...>

16. jQuery. [En línea] The jQuery Foundation, 2013. [Citado el: 13 de 06 de 2013.] <http://jquery.com/>.
17. Apache Tomcat. [En línea] 1999-2013. [Citado el: 15 de 06 de 2013.] <http://tomcat.apache.org/>.
18. Carlos Reynoso, Nicolás Kicillof. *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*. 2004.
19. Larman, Craig. *UML y Patrones. Una introducción al análisis y el diseño orientado a objetos y al proceso unificado.*. 2009.
20. Pressman, Roger S. *ingeniería de software. Un enfoque práctico. Capítulo 8 Modelado de Análisis*. 2005.
21. Rodríguez, Jorge. *Spring y el principio de Hollywood*. 2005.
22. SpringHispano.org. [En línea] 2013. [Citado el: 2013 de 06 de 19.] <http://www.springhispano.org/?q=node/46>.
23. César de la Torre LLorente, Unai Zorrilla Castro, Miguel Angel Barros, Javier Calvarro Nelson. *Guía de Arquitectura N-Capas orientadas al dominio con .NET 4.0*. Impreso en España : s.n., 2010. ISBN:978-84-936696-3-8.
24. Escribano, Gerardo Fernández. *Introducción a Extreme Programming*. 2002.
25. Michael, Marca Huallpara Hugo. *Análisis y Diseño de Sistemas II*.