

**Universidad de las Ciencias Informáticas
Facultad 1**



Título: Sistema para la gestión de inventario en los laboratorios de la
Facultad 1.

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

Autores: José Rafael Aguilar Quintana

Yunier Vega Pérez

Tutores: Ing. Arturo Luis Lara Fernández

Ing. Orestes Martínez Castellanos

Ciudad de La Habana, junio, 2010

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores del trabajo titulado:

Sistema para la gestión de inventario en los laboratorios de la Facultad 1 y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

José Rafael Aguilar Quintana

Yunier Vega Pérez

Ing. Arturo Luis Lara Fernández

Ing. Orestes Martínez Castellanos

José Rafael:

Estas son líneas muy difíciles de escribir, porque son muchas las personas que han influido en mí y me han brindado su mano en cualquier momento para hacer realidad este sueño. Quiero agradecer de todo corazón a todos.

A mi mamá por ser un espejo para mí y de la cual estoy muy orgulloso de ser su hijo, la que me ha guiado y enseñado todo en la vida, la que ha estado en cada paso que he dado, no sería nada sin ella, mami no tengo palabras....

A mi papá por nunca dejarme a un lado aunque no viva conmigo y apoyarme en lo que ha podido siempre, es uno de los que más orgulloso se sentirá a partir de este momento, ya que es lo que me ha inculcado y ha querido siempre.

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

A mis amigos que son muchos y ellos saben quienes son, en especial a los que han vivido muchos momentos conmigo, Elisandry, Yuny, Danilo, Jose, Alexey, Sander, Daniel, David, Pedri, Rene, Yunieski, Yunier, a todo el piquete de guisa ellos saben quienes son, a todos los que he conocido en la UCI, a todos.

A Ivett que ha pasado muchos momentos conmigo y me ha ayudado mucho, la cual estoy seguro de que seguirá mi camino y llegara a este hermoso momento.

A mis totures por brindarme su ayuda durante el tiempo que trabajamos y a todos los profesores que han influido en mi formación.

A la revolución por darme la oportunidad de estudiar y llegar hasta aquí.

Yunier:

A mis tutores por la confianza y el tiempo empleado en este trabajo de diploma.

A los amigos que contribuyeron con su esfuerzo al logro de este trabajo de diploma.

A mi familia por el apoyo incondicional que me han brindado.

A mi novia por todas las atenciones y ayuda prestadas.

A todas aquellas personas anónimas que han contribuido a mi formación como revolucionario.

José Rafael:

A mi mamá, a mi papá y a toda mi familia, en especial a mi hermana y mis abuelos que ya no están conmigo...

Yunier:

Dedico este trabajo de diploma especialmente a mis abuelos por haber sido los responsables de mi formación, a mi madre por haberme apoyado en los momentos difíciles y confiar en mí, a mi novia por haberme inspirado confianza durante todo este tiempo y brindarme su apoyo incondicional.

Entre los procesos que se llevan a cabo en la dirección de laboratorios de la Universidad de las Ciencias Informáticas se encuentra el Proceso de Gestión de Inventario, que tiene como objetivo inventariar los medios con los que se cuenta para la capacitación del personal y mediante la misma obtener mejores resultados docentes y productivos. Actualmente este proceso se realiza de forma manual y es considerado una tarea complicada debido a la gran cantidad de información que se maneja y la importancia de esta.

Este trabajo consiste en la creación de un sistema informatizado para la gestión de los medios y bienes con que cuentan los laboratorios de la Facultad 1 , en respuesta a la necesidad de mejorar y perfeccionar la manera en que actualmente se lleva a cabo esta tarea. La utilización de un sistema informatizado para la gestión de estos medios significaría una mejora considerable de este proceso en cuanto a eficiencia, organización y rapidez para obtener información, contribuyendo favorablemente a la correcta asignación y planificación de los medios y mediante la misma obtener mejoras en la capacitación del personal lo que traería consigo un avance en el desarrollo docente y en la producción de software en la universidad.

El desarrollo del trabajo proporcionará una aplicación con tecnologías novedosas que aporten un mejor control de los medios y bienes en los laboratorios, una alta seguridad en la entrada al sistema y garantizará los reportes correspondientes a la existencia y movimientos de medios.

Palabras claves: Sistema, Inventario, Gestión.

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	5
1.1 Introducción.....	5
1.2 Los Sistemas de Gestión de Inventario.....	5
1.2.1 Los Sistemas de Gestión de Inventario en el Mundo.....	6
1.2.2 Los Sistemas de Gestión de Inventario en Cuba.....	7
1.2.3 Los Sistemas de Gestión de Inventario en la Universidad de las Ciencias Informáticas.....	9
1.2.4 Marco Teórico-Conceptual de la Investigación.....	10
1.3 Tendencias actuales en el desarrollo de Software.....	10
1.4 Metodología, lenguaje y herramientas de modelado.....	12
1.4.1 Metodología de desarrollo de software.....	12
1.4.2 Lenguaje de modelado.....	14
1.4.3 Herramienta CASE para el modelado.....	15
1.4.4 Fundamentación de la herramienta, lenguaje y metodología a utilizar para el modelado.....	16
1.5 Arquitectura.....	16
1.5.1 Fundamentación de la arquitectura a utilizar.....	17
1.6 Herramientas, tecnologías y lenguajes para el desarrollo.....	17
1.6.1 Marco de trabajo.....	17
1.6.2 Lenguaje para el desarrollo Web.....	19
1.6.3 Servidor Web Apache.....	20
1.6.4 Object Relational Mapper.....	21
1.7 Propuestas de herramientas seleccionadas.....	24
1.8 Conclusiones.....	24
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	25
2.1. Introducción.....	25
2.2. Breve exposición de la situación actual.....	25
2.4 Personas relacionadas con el sistema.....	26
2.5 RF (Requerimientos funcionales del sistema).....	27
2.6 Requerimientos no funcionales del sistema.....	29
2.7 Fase de exploración. Definición.....	31
2.7.1 Historias de Usuario.....	31
2.8 Fase de Planificación. Definición.....	37
2.8.1 Estimación del esfuerzo por historia de usuario.....	37
2.8.2 Plan de Iteraciones.....	38
2.8.3 Plan de duración de las Iteraciones.....	39
2.8.4 Plan de entregas.....	40
2.9 Conclusiones.....	40
CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN	41
3.1 Introducción.....	41
3.2 Diseño.....	41
3.2.1 Tarjeta CRC.....	41
3.2.2 Diseño de base de datos.....	44
3.3 Implementación.....	45
3.3.1 Iteración 1.....	45
3.3.2 Iteración 2.....	46
3.3.3 Iteración 3.....	46
3.4 Tareas generales de la Implementación.....	47
3.5 Módulos Utilizados en la Implementación.....	48

3.5.1 Módulo Autenticator	49
3.5.2 Módulo LDAP	49
3.5.3 Módulo Laboratorio	50
3.5.4 Módulo Proyecto	50
3.5.5 Módulo Medio	50
3.5.6 Módulo Reporte	50
3.5.7 Módulo Parte Diario	51
3.6 Diagrama de despliegue	51
3.7 Conclusiones	52
CAPÍTULO 4: PRUEBAS Y ESTUDIO DE FACTIBILIDAD.....	53
4.1 Introducción.....	53
4.2 Pruebas.....	53
4.2.1 Pruebas unitarias	53
4.2.2 Pruebas de Aceptación.	54
4.2.3 Validación del sistema.....	65
4.3 Estudio de Factibilidad.....	66
4.3.1 Características del proyecto	66
4.3.2 Cálculo de instrucciones fuentes, esfuerzo, tiempo de desarrollo, cantidad de hombres y costo.....	68
4.4 Beneficios tangibles e intangibles.....	71
4.5 Análisis de costo.	71
4.6 Conclusiones del Capítulo.	71
CONCLUSIONES GENERALES	72
RECOMENDACIONES.....	73
GLOSARIO DE TÉRMINOS	74
REFERENCIAS BIBLIOGRÁFICAS.....	77
BIBLIOGRAFÍA CONSULTADA	78
Anexos.....	79

ÍNDICE DE TABLAS

Tabla 1. Personas relacionadas con el sistema.....	27
Tabla 2. Historia de usuario “Autenticar usuario”.....	32
Tabla 3. Historia de usuario “Gestionar usuario”.....	33
Tabla 4. Historia de usuario “Gestionar laboratorio”.....	33
Tabla 5. Historia de usuario “Gestionar nomenclador”.....	34
Tabla 6. Historia de usuario “Gestionar Medio”.....	34
Tabla 7. Historia de usuario “Gestionar proyecto”.....	34
Tabla 8. Historia de usuario “Distribuir proyecto”.....	35
Tabla 9. Historia de usuario “Realizar movimiento de medio”.....	35
Tabla 10. Historia de usuario “Registrar parte diario”.....	36
Tabla 11. Historia de usuario “Realizar búsquedas en el sistema”.....	36
Tabla 12. Historia de usuario “Obtener reportes”.....	36
Tabla 13. Plan de estimación de esfuerzo por historias de usuario.....	38
Tabla 14. Plan de iteraciones.....	39
Tabla 15. Plan de entregas.....	40
Tabla 16. Tarjeta CRC correspondiente a la clase CControladora.....	42
Tabla 17. Tarjeta CRC correspondiente a la clase CUsuario.....	42
Tabla 18. Tarjeta CRC correspondiente a la clase CLaboratorio.....	43
Tabla 19. Tarjeta CRC correspondiente a la clase CNomenclador.....	43
Tabla 20. Tarjeta CRC correspondiente a la clase CMedio.....	43
Tabla 21. Tarjeta CRC correspondiente a la clase CProyecto.....	43
Tabla 22. Tarjeta CRC correspondiente a la clase CDistribución.....	44
Tabla 23. Tarjeta CRC correspondiente a la clase CBúsqueda.....	44
Tabla 24. Tarjeta CRC correspondiente a la clase CReporte.....	44
Tabla 25. Historias de Usuario correspondientes a la primera iteración.....	46
Tabla 26. Historias de Usuario correspondientes a la segunda iteración.....	46
Tabla 27. Historias de Usuario correspondientes a la tercera iteración.....	47
Tabla 28. Tarea # 1 perteneciente a las tareas generales de implementación.....	47
Tabla 29. Tarea # 2 perteneciente a las tareas generales de implementación.....	48
Tabla 30. Tarea # 3 perteneciente a las tareas generales de implementación.....	48
Tabla 31. Tarea # 4 perteneciente a las tareas generales de implementación.....	48
Tabla 32. Caso de Prueba de Aceptación HU1_P1 “Autenticar Usuario”.....	56
Tabla 33. Caso de Prueba de Aceptación HU2_P1 “Gestionar Usuario”.....	56
Tabla 34. Caso de Prueba de Aceptación HU2_P2 “Gestionar Usuario”.....	57
Tabla 35. Caso de Prueba de Aceptación HU2_P3 “Gestionar Usuario”.....	57
Tabla 36. Caso de Prueba de Aceptación HU2_P4 “Gestionar Usuario”.....	57
Tabla 37. Caso de Prueba de Aceptación HU3_P1 “Gestionar Laboratorio”.....	58
Tabla 38. Caso de Prueba de Aceptación HU3_P2 “Gestionar Laboratorio”.....	58
Tabla 39. Caso de Prueba de Aceptación HU3_P3 “Gestionar Laboratorio”.....	58

Tabla 40. Caso de Prueba de Aceptación HU3_P4 “Gestionar Laboratorio”	59
Tabla 41. Caso de Prueba de Aceptación HU4_P1 “Gestionar Nomenclador”	59
Tabla 42. Caso de Prueba de Aceptación HU4_P2 “Gestionar Nomenclador”	59
Tabla 43. Caso de Prueba de Aceptación HU4_P3 “Gestionar Nomenclador”	60
Tabla 44. Caso de Prueba de Aceptación HU5_P1 “Gestionar Medios”	60
Tabla 45. Caso de Prueba de Aceptación HU5_P2 “Gestionar Medios”	60
Tabla 46. Caso de Prueba de Aceptación HU5_P3 “Gestionar Medios”	61
Tabla 47. Caso de Prueba de Aceptación HU5_P4 “Gestionar Medios”	61
Tabla 48. Caso de Prueba de Aceptación HU6_P1 “Gestionar Proyecto”	61
Tabla 49. Caso de Prueba de Aceptación HU6_P2 “Gestionar Proyecto”	61
Tabla 50. Caso de Prueba de Aceptación HU6_P3 “Gestionar Proyecto”	62
Tabla 51. Caso de Prueba de Aceptación HU6_P4 “Gestionar Proyecto”	62
Tabla 52. Caso de Prueba de Aceptación HU7_P1 “Distribuir Proyecto”	62
Tabla 53. Caso de Prueba de Aceptación HU8_P1 “Realizar movimientos de medios”	63
Tabla 54. Caso de Prueba de Aceptación HU8_P2 “Realizar movimientos de medios”	63
Tabla 55. Caso de Prueba de Aceptación HU8_P3 “Realizar movimientos de medios”	63
Tabla 56. Caso de Prueba de Aceptación HU9_P1 “Registrar parte diario”	64
Tabla 57. Caso de Prueba de Aceptación HU10_P1 “Realizar búsquedas en el sistema”	64
Tabla 58. Caso de Prueba de Aceptación HU10_P2 “Realizar búsquedas en el sistema”	64
Tabla 59. Caso de Prueba de Aceptación HU11_P1 “Obtener Reporte”	65
Tabla 60: Resultado de la Prueba Piloto.	66
Tabla 61: Entradas externas.	67
Tabla 62: Salidas externas.	67
Tabla 63. Consultas Externas.	67
Tabla 64. Archivos lógicos internos.	68
Tabla 65. Puntos de función desajustados.	68
Tabla 66. Características.	68
Tabla 67. Factores de escala.	69
Tabla 68. Multiplicadores de esfuerzo.	69
Tabla 69. Resultados.	70

INTRODUCCIÓN

A lo largo de la historia la sociedad ha estado en constante desarrollo y estrechamente relacionada con los nuevos avances tecnológicos, por lo cual se hace necesario el uso de las computadoras en la informatización de muchas actividades. Todo esto con el objetivo de lograr mayor calidad y seguridad en los datos. Cuba está igualmente ligada a este desarrollo que existe en todas partes del mundo, lo que permitiría un crecimiento en el desarrollo de la nación. Como parte de un programa de la revolución cubana surge en el año 2002 la Universidad de las Ciencias Informáticas (UCI) con el objetivo de desarrollar la producción de software y expandir los conocimientos en la rama de la informática y comunicaciones en todo el país. La institución cuenta con diez facultades en su sede central y tres facultades regionales, cada una posee laboratorios dedicados a desarrollar la docencia y la producción, los cuales cuentan con una serie de medios a su disposición. La Facultad 1 en la cual se centra la investigación requiere realizar un control de los medios y bienes que posee cada laboratorio.

En estos momentos este proceso de inventario se realiza de forma manual por parte de la dirección de laboratorios en el área encargados de manejar esta información, almacenando los datos en documentos Word o Excel y enviando los mismos a través de correo electrónico, lo que resulta muy engorroso a la hora de manipular los datos, ya sea para realizar una búsqueda, modificar algún dato o cualquier actividad, afectando así la calidad y rapidez del proceso. Todo esto puede traer consigo:

- La búsqueda y recuperación de datos a la hora de hacer reportes es lenta.
- Exista mala manipulación de los datos manejados en el inventario.
- Exceso en el pedido de medios o bienes y la pérdida de los mismos.
- Mala planificación y asignación de los medios y bienes.

Después de analizar la problemática expuesta se tiene el siguiente **problema científico**:

¿Cómo mejorar el control de los medios y bienes en los laboratorios de la Facultad 1?

Para darle solución al problema científico se tiene como **objeto de estudio**: gestión de inventario y como **campo de acción**: gestión de inventario en los laboratorios de la Facultad 1.

El **objetivo general** es: Desarrollar un sistema informático para la gestión de inventario en los laboratorios de la Facultad 1.

Del objetivo general se derivan los siguientes **objetivos específicos**:

- Identificar los procesos relacionados con la gestión de inventario en los laboratorios de la Facultad 1.
- Identificar y describir los requerimientos del sistema.
- Realizar el diseño e implementación del sistema.
- Garantizar la calidad y puesta en funcionamiento del sistema.

Para garantizar el cumplimiento de los objetivos trazados se tienen las siguientes **tareas de investigación**:

- Análisis bibliográfico de los procesos de gestión de inventario.
- Entrevistas a los clientes para entender los procesos de gestión de inventario en los laboratorios de la Facultad 1.
- Fundamentación de la metodología de desarrollo, herramientas y lenguajes a utilizar en el desarrollo del sistema propuesto.
- Descripción y elaboración de los artefactos que genera la metodología de desarrollo seleccionada.
- Modelación de la base de datos del sistema propuesto.
- Diseño de la interfaz del sistema propuesto.
- Implementación del sistema propuesto.
- Realización de pruebas al sistema propuesto.
- Análisis de factibilidad del sistema propuesto.

Para guiar la investigación se plantea la siguiente **hipótesis**:

El desarrollo de un sistema informático para la gestión de inventario en los laboratorios de la Facultad 1 permitirá un mejor control de los medios y bienes.

Variables:

Variable independiente: Desarrollo de un sistema informático para la gestión de inventario.

Variable dependiente: Control de los medios y bienes.

[Anexo 1: Operacionalización de las variables.](#)

Métodos de Investigación Científica empleados:

Métodos teóricos:

1-Histórico y Lógico: mediante este método se analiza la trayectoria real de los fenómenos, su evolución y desarrollo. Se utiliza este método ya que entre los objetivos propuestos está identificar los procesos relacionados con la gestión de inventario, además se utiliza para investigar sobre sistemas de gestión que tengan similitud con el que se va a implementar.

2-Análisis y síntesis: Facilita el entendimiento del fenómeno en que se trabaja, es más útil la división de estas en diferentes fases y de esta forma descubrir sus características generales, lo que ayuda a seguir una correcta investigación. A partir de un estudio detallado de la información obtenida se hace necesario organizarla y sintetizarla para lograr una estructura adecuada.

3-Modelación: El modelo científico es un instrumento de la investigación de carácter material o teórico, creado para reproducir el fenómeno que se está estudiando. Se modelan todos los procesos relacionados con la gestión de inventario en los laboratorios de la Facultad 1 de la Universidad de las Ciencias Informáticas.

Métodos empíricos:

1-Entrevista: Es una conversación planificada entre el investigador y el entrevistado para obtener información. Se realizan entrevistas a los clientes para obtener información y entender mejor los procesos relacionados con la gestión de inventario en los laboratorios de la Facultad 1 de la Universidad de las Ciencias Informáticas.

Al concluir la investigación se espera obtener como posibles resultados: Un sistema informático que mediante la gestión de inventario en los laboratorios de la Facultad 1 contribuirá a la organización y un mejor control de los bienes y medios.

Beneficios:

- El control de los medios y bienes de los laboratorios de la Facultad 1.
- Evitar pérdidas de medios y bienes en los laboratorios de la Facultad 1.

- Se agilizará el proceso de búsqueda y recuperación de los datos manejados en el inventario.
- Mejorará el trabajo con los datos manejados en el inventario.

El trabajo de diploma está estructurado en los siguientes capítulos:

Capítulo 1: Fundamentación teórica

En este capítulo se muestra el resultado de la investigación bibliográfica realizada sobre el objeto de estudio. Se describen además las características del lenguaje de programación, sistema gestor de bases de datos, metodología de desarrollo y las diferentes herramientas propuestas para el desarrollo del sistema.

Capítulo 2: Características del sistema

En este capítulo se presenta un estudio del funcionamiento y la información que se maneja para desarrollar el sistema, se muestran los requerimientos de la propuesta de solución mediante las historias de usuario que describirán en detalle sus características así como los requerimientos no funcionales que debe cumplir.

Capítulo 3: Diseño e Implementación.

En este capítulo se hace alusión a las fases de Diseño e Implementación propias de la metodología de desarrollo utilizada para guiar la elaboración del sistema que se propone. Se exponen además los artefactos generados durante el transcurso de las mismas.

Capítulo 4: Pruebas y Estudio de Factibilidad.

En este capítulo se hace referencia a la fase de Pruebas propia de la metodología de desarrollo utilizada para guiar el proceso de elaboración del sistema que se propone, obteniendo como resultado un conjunto de casos de pruebas mediante las cuales el cliente podrá validar la correcta implementación de cada uno de los requerimientos funcionales. Además, se realiza un análisis del costo, el esfuerzo y los beneficios tangibles e intangibles del sistema que se propone, actividad de suma importancia a la hora de definir la rentabilidad del sistema, se detalla el método de estimación aplicado.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En este capítulo se muestra el resultado de la investigación bibliográfica realizada sobre la gestión de inventario en el ámbito internacional, nacional y la Universidad, así como las tendencias actuales que ha adquirido la misma, se detallan los lenguajes, tecnologías, metodologías y herramientas utilizadas en el desarrollo del sistema, analizando sus características, ventajas y desventajas.

1.2 Los Sistemas de Gestión de Inventario

El **inventario** no es más que una relación escrita, hecha en un momento dado, de los bienes, créditos y deudas de toda índole de una persona o entidad, o de los efectos contenidos en un local, establecimiento o propiedad. En términos generales, se aplica también al cálculo de recursos o patrimonio de un país, al examen o análisis de una situación y a los medios de que se dispone para emprender una acción determinada. En la actualidad la **gestión del inventario** es una medida vital en casi todos los negocios, esfuerzo de las empresas con lo que pueden mantener un seguimiento adecuado de las pérdidas que se reclamen y el control de los recursos; puede mantener más de un balance, en virtud de las existencias y el robo a un mínimo.

Los sistemas de gestión han llegado a ser una fuerte arma para planificar, distribuir y controlar los recursos, ya sean tangibles o intangibles, en los que representa un papel fundamental el desarrollo y aplicación de la información a los procesos de administración. Un **sistema de gestión**, en pocas palabras, se define como: la automatización de los procesos generados en una organización, empresa o institución como: producción, contabilidad, distribución, recursos humanos, transporte, entre otras. Los sistemas de gestión tienen una triple dimensión: facilitar, simplificar y realizar de forma automática los procesos que tradicionalmente se realizaban de forma manual. Así pues, sustituyen ventajosamente al personal encargado, evitando los errores y mejorando la velocidad media; establecen un imperceptible control de presencias en las entidades financieras, con ventajas incuestionables en cuanto a fiabilidad y seguridad; realizan los reiterativos procesos contables sin errores en las operaciones, con rapidez y agilidad.

Después de haber realizado el análisis de lo que es un Sistema de Gestión, lo que es un Inventario y de que es la Gestión de Inventario, se puede decir que un Sistema de Gestión de Inventario no es más que un sistema informático que permite automatizar los procesos relacionados con la gestión del inventario, para un mejor control del mismo y la correcta toma de decisiones. [1]

1.2.1 Los Sistemas de Gestión de Inventario en el Mundo

En el transcurso de la investigación se estudiaron varios sistemas dedicados a la gestión de inventarios en el ámbito internacional. A continuación se caracterizan algunos de estos sistemas.

GIB (Gestión de Inventarios de Bienes)

La herramienta GIB – Gestión de Inventarios de Bienes – para sistemas operativos Windows ha sido desarrollada para gestionar de forma eficaz la elaboración y mantenimiento del Inventario de Bienes y derechos de las Administraciones Locales, automatizando las amortizaciones y simplificando el trabajo del personal encargado de su mantenimiento.

Las principales características del software de GIB son:

- Ajustado a la legislación de las administraciones locales.
- Control de usuarios, permisos de acceso y auditoria, donde cada usuario se identifica en la entrada a la aplicación para poder tener un control de sus privilegios.
- Parametrización del programa.
- Procedimientos avanzados y restringidos como la amortización masiva del inventario, explotación de datos y comunicación masiva con otras aplicaciones.
- Posibilidad de almacenamiento de fotos y planos de los bienes en la ficha de cada bien.
- Registro de las sucesivas mejoras de cada bien. Se puede realizar un informe anual con las altas, bajas y modificaciones realizadas sobre los bienes.
- Consultas simples y avanzadas al inventario.
- Emisión de listados y fichas de cualquier tipo de bien. Impresión de etiquetas con código de barras para identificar cada bien.

- Cuenta con una interfaz sencilla, cómoda e intuitiva.

OPEN/SIDE (Sistema Integrado de Desarrollo Empresarial)

El Open/SIDE es un sistema diseñado para la administración de las empresas. Consta de sistemas integrados para las áreas Financiero/Contable, Comercial y Recursos Humanos. Los sistemas Financieros Contables permiten a la empresa operar con información relativa a los movimientos de contabilidad, caja y bancos, cuentas por pagar, activos fijos, verificación presupuestaria, centros de costos, permitiéndoles realizar su gestión funcional. Los sistemas Comerciales integran la información de la empresa referente a sus actividades de venta y facturación, compra con ordenes, inventarios, cuentas por cobrar y control de ordenes de venta. Los sistemas de Recursos Humanos brindan y utilizan información relativa al recurso humano de la entidad en sus diferentes campos: nomina, administración de personal, reclutamiento y selección, salud ocupacional y capacitación. Open/SIDE está construido sobre la más poderosa, sólida y consolidada plataforma de base de datos del mercado: ORACLE, lo que asegura la consistencia, integridad y administración eficiente de la información corporativa.

Es un sistema multicompañía que permite que la información de cada empresa se opere de manera independiente a las otras. Todas las transacciones o movimientos pueden ser realizadas en cualquier moneda. Permite operar bajo un modelo tanto de contabilidad financiera como de contabilidad de costos. Cada uno de los módulos puede funcionar independiente a los otros, requiriendo únicamente los sistemas primarios. Además, el Open/SIDE tiene la capacidad de integrarse a otras importantes soluciones con las que se cuenta, de acuerdo con las actividades de la empresa. El sistema ofrece dispositivos de seguridad para controlar el acceso a los diferentes sistemas por medio de grupos de usuarios.

1.2.2 Los Sistemas de Gestión de Inventario en Cuba

En Cuba existen Sistemas de Gestión de Inventarios, algunos de ellos serán abordados a continuación:

SIP (El Sistema de Inventario del Patrimonio Cultural y Natural)

Como su nombre lo indica fue creado con fines patrimoniales como: el registro, el control y la conservación de los mismos. El sistema que se desarrolla desde 1988 tiene entre sus fines:

- Facilitar el cumplimiento de la legislación nacional e internacional en lo que respecta al registro

de los bienes patrimoniales.

- Contribuir a ejercer el control y a priorizar la conservación de los bienes patrimoniales más valiosos del país.
- Facilitar el intercambio de información relacionado con el patrimonio cultural y natural.
- Responder a las necesidades de diferentes usuarios: museólogos, museógrafos, conservadores, investigadores y otros con intereses más generales.

El SIP contempla bases de datos conformadas con los bienes de relevancia que integran el patrimonio cultural del país. Los que se encuentran en las instituciones correspondientes al museo cubano y los que poseen y custodian organismos e instituciones estatales de connotada significación. Está formado por 20 bases de datos que tienen un formato común y en cada manifestación se han elaborado listados de términos y Tesoros que facilitan la entrada, búsqueda y recuperación de la información. En su diseño participaron grupos de especialistas de alta calificación en cada materia, informáticos y especialistas en computación.

INVENFOR (El Sistema Informatizado para el Procesamiento del Inventario Forestal)

Es un sistema que permite procesar con mayor rapidez y eficiencia la información obtenida en el Inventario Forestal (diámetro medio, altura media) en función de estimar los parámetros, de forma tal que tribute a una planificación del manejo forestal más eficiente. La aplicación de este software supone la disminución del tiempo de procesamiento de los datos y errores de cálculo. Con la introducción de esta herramienta a las empresas, se reduce considerablemente el tiempo en que se realizan los cálculos y con esto la eficiencia en la obtención de reportes. El usuario se encarga de entrar la información obtenida del inventario forestal al sistema mediante una interfaz interactiva y amigable, este a su vez es el responsable de procesarla y presentar los informes que sean solicitados. Este sistema es de gran utilidad en los centros donde se estudie la carrera de Ingeniería Forestal como es el caso de la Universidad de Pinar del Río. Con el uso de INVENFOR se obtienen los resultados rápidamente y los profesores junto con sus alumnos pueden dedicar más tiempo a la interpretación de los mismos.

1.2.3 Los Sistemas de Gestión de Inventario en la Universidad de las Ciencias Informáticas

Sistema Assets (Sistema de Gestión Integral)

Actualmente en la UCI se trabaja con el Sistema ASSETS, que es un sistema multiusuario que se monta en una plataforma de servidores SQL, dividido en módulos económicos que trabajan en conjuntos para el control de las actividades económicas, financieras y contables sobre los medios materiales y financieros. ASSETS es una aplicación cliente-servidor programada en Visual Basic 6.0 y Microsoft SQL Server 2000, utilizando adicionalmente Crystal Reports 7.0 para la generación de reportes de salidas. Al estar en plataforma SQL, garantiza mayor seguridad y consistencia en los datos, se obliga que sea ilimitado el número de usuarios conectados y hace posible la utilización de servidores remotos. Es uno de los mejores ejemplos de la cubanización de un software extranjero, que cada día se adapta más a la realidad económica cubana. Fue Introducido hace 8 años en Cuba por el Ingeniero en Informática Marco de Lucas, italiano que comercializa el producto en la Isla por medio de la empresa cubana Infomáster, donde sus clientes corporativos suman más de 400 entre los que sobresalen, además del MES (Ministerio de educación superior), el Consejo de Estado, la Aduana General de la República, el Ministerio de Auditoría y Control, el de Justicia y el de Finanzas y Precios, entre otros. El Módulo Inventario del Assets divide los medios en dos tipos de cuentas: Activos Fijos y Útiles Herramientas.

Los Activos Fijos son bienes adquiridos por la Universidad con carácter permanente, es decir, que duran un plazo largo y pueden o no tener presencia física, por ejemplo: computadoras, mesas, programas computacionales, terrenos, equipos de transporte, entre otros. Los Útiles Herramientas son aquellos que se utilizan para realizar las actividades de mantenimiento, talleres, almacenes, así como los equipos de protección física. Comprende entre otros, herramientas manuales, artículos de protección personal, utensilios de laboratorios, minis calculadoras, utensilios menores de cocina. Los medios que pertenecen a esta cuenta tienen poco tiempo de duración.

El Assets es un Sistema de Gestión Integral estándar y parametrizado que permite controlar, realizar y contabilizar todas las transacciones comunes relacionadas con el proceso de compraventa e incluye así mismo, los procedimientos necesarios para registrar los movimientos de los activos fijos y de los útiles y herramientas, permite además administrar todos los medios de la empresa de los cuales

preocupa tener una información detallada, así como su ubicación física por centros de costos, áreas de Responsabilidad y empleados; para evitar extravío o facilitar su ubicación cuando se necesite. El sistema además facilita el ingreso al inventario y también la administración de los códigos asignados y todo el manejo que internamente se requiere, es capaz de registrar en la contabilidad todas las transacciones en dos monedas y en el mismo momento de su ocurrencia. Por ser un sistema contable proporciona opciones de seguridad que le permiten limitar el acceso a los diferentes procesos del sistema de acuerdo con el perfil de cada usuario y no cuenta con un catálogo de reportes que muestre parte de su información a otros usuarios para la toma de decisiones.

1.2.4 Marco Teórico-Conceptual de la Investigación

Después de realizar un estudio de los sistemas de gestión de inventario utilizados a nivel internacional y nacional, se puede apreciar que a pesar de las amplias posibilidades que brindan en cuanto a gestión de inventario, en general fueron diseñados para gestionar de forma eficaz la elaboración y mantenimiento del Inventario de los medios y bienes.

En el caso de los sistemas GIB y OPEN/SIDE se puede señalar que están desarrollados fundamentalmente para la gestión empresarial manteniendo un estricto control de los usuarios que ingresan al sistema, poseen módulos integrados para las áreas Financiero/Contable, Comercial y Recursos Humanos.

El sistema Assets que actualmente se utiliza en la universidad trabaja basado en módulos económicos, el módulo inventario divide los medios en dos tipos de cuenta, activos fijos y útiles herramientas. El sistema facilita el ingreso al inventario y la administración a los códigos asignados, así como el manejo de todos los datos que internamente se almacenan.

1.3 Tendencias actuales en el desarrollo de Software

En la actualidad la sociedad ha alcanzado altos niveles en el desarrollo de software, como parte de estos logros se encuentra el software libre, un movimiento tecnológico que ha revolucionado en la comunidad informática, por sus características especiales ha permitido la experimentación de nuevas

formas de desarrollo y mantenimiento de programas, nuevas normas legales y nuevos modelos económicos. Para entender este concepto se debe pensar en “libre” como libertad de expresión. Software libre está referido a la libertad de los usuarios para realizar cambios, copiar, distribuir, estudiar, ejecutar y mejorar el software, en se refiere a cuatros libertades de los usuarios del software:[2]

La libertad de usar el programa, para cualquier propósito. La libertad de estudiar como funciona el programa, y adaptarlo a las necesidades reales. La libertad de distribuir copias, con las que se puede ayudar a otros. La libertad de hacerle mejoras al programa y hacer publicas estas mejoras para que sean de beneficio en toda la comunidad. La posibilidad de tener acceso al código fuente del programa es una condición necesaria para el software libre, vital para realizar modificaciones y publicar versiones mejoradas. Esto nos da la viabilidad económica de estos sistemas libres, no necesariamente gratuitos, sino que da la posibilidad de comercializarlo, regalarlo, prestarlo con total libertad y protegerlo legalmente. Proporcionando una buena alternativa para los países más pobres o subdesarrollados.

Ventajas:

- Innovación tecnológica: el desarrollo en comunidad de este sistema y el conocimiento del código fuente, propicia que a cada instante el desarrollador necesite nuevas actualizaciones y las realice el mismo, proponiendo nueva funcionalidad al programa.
- Escrutinio público: el proceso de revisión pública al que está sometido el desarrollo del software libre imprime un gran dinamismo al proceso de corrección de errores. Cada mejora es socializada libremente, la comunidad puede cambiar la realidad de las innovaciones.
- Costo: El costo total de propiedad del sistema operativo Linux es menos de la mitad que el de Windows. Gran parte del ahorro proviene de no tener que pagar licencia y de sus menores costos de administración.
- Independencia del proveedor: El software libre garantiza una independencia con respecto al proveedor gracias a la disponibilidad del código fuente.
- Desarrollo de la industria local: En el software libre al disponer del código fuente de la aplicación, es posible desarrollar internamente las mejoras o las modificaciones que sean necesarias. De este modo se contribuye a la formación de profesionales en nuevas tecnologías, al desarrollo local y de la industria nacional de software.

- Privacidad y seguridad: El software, libre por su carácter abierto, dificulta la introducción de código malicioso, espía o de control remoto, debido a que el código lo revisan muchos usuarios y desarrolladores que pueden detectar posibles puertas traseras. En el mundo del software libre, cualquier programador puede realizar una auditoria para comprobar que no se ha introducido código malicioso, y, a su vez, cualquier entidad puede encriptar la aplicación que utilice para proteger sus datos. [3]

Desventajas:

- La curva de aprendizaje es mayor: Resulta engorroso trabajar con el software libre si ya se ha trabajado con un software propietario. Las personas que nunca hayan usado una computadora pueden tardar lo mismo en aprender a usar el software libre, que el software propietario, pero si ya han usado software propietario, generalmente tardan mas en usar el software libre.
- Se necesita dedicar recursos a la reparación de errores: El software libre se adquiere sin garantías explícitas, se vende tal cual, aunque pueden existir garantías específicas para determinadas situaciones, por lo que hay que dedicar recursos para repara cualquier daño del software, si embargo en el software propietario es imposible repara errores, hay que esperar a obtener una nueva versión.

1.4 Metodología, lenguaje y herramientas de modelado

1.4.1 Metodología de desarrollo de software

La Ingeniería de Software surge con el objetivo de crear y mantener las aplicaciones de software, aplicando las tecnologías y practicas computacionales. El desarrollo y evolución constante experimentada por los procesos de Ingeniería de software, ha traído consigo la realización de varias tareas en este campo, tales como: análisis de requisitos, especificación, diseño y arquitectura, programación, prueba, documentación y mantenimiento. El proceso de desarrollo de software, define el conjunto de actividades precisas para convertir los requisitos de los usuarios en el conjunto seguro y resistente de artefactos que componen un producto de software. Las tendencias actuales luego de un perfeccionamiento del software durante años, han llevado a cabo dos corrientes significativas: los llamados métodos ligeros y métodos pesados. Aunque ambos están encaminados a beneficiar la labor de aquellas personas que intervienen en el proceso de desarrollo. Los métodos ligeros o ágiles

proponen mejorar la calidad del software teniendo como premisa la comunicación inmediata y directa, mientras que los métodos pesados obtienen sus resultados a través de orden y documentación. Se hace necesario definir metodologías para guiar el proceso de desarrollo de un producto de software. Las metodologías se definen por pasos a seguir para el cumplimiento de un objetivo. El objetivo dentro del desarrollo del software es producir un producto de alta calidad que cumpla con los requerimientos del cliente.[4]

Entre las metodologías de desarrollo de software ágiles se encuentran:

- XP
- SCRUM
- SCRUMXP(SXP)

Entre las características generales que presentan dichas metodologías están:

- No pueden aplicarse a todo tipo de Proyectos
- Están orientadas en función de los nuevos principios del desarrollo del software.
- Pueden ser ajustables a las características del proyecto.

Extreme Programming (XP)

Basándose en la simplicidad, la comunicación y la reutilización del código, surge la Programación Extrema (XP), siendo así una metodología ágil de desarrollo de software. Como requisito para alcanzar el éxito del proyecto se obtiene el usuario final como parte del equipo. Es una metodología con reconocido éxito y se utiliza en proyectos con entregas a cortos plazos.[5]

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por la superación de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en la realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios.[6]

Las características fundamentales de XP son:

- Desarrollo iterativo e incremental: permite pequeñas mejoras, unas tras otras,

consecutivamente.

- Pruebas unitarias: estas son continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión.
- Programación en parejas: recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Permitiendo que el código sea revisado al mismo tiempo que se escribe.
- Frecuente interacción del equipo de programación con el cliente o usuario: recomendando que un representante del cliente trabaje junto al equipo de desarrollo.
- Corrección de todos los errores antes de añadir una nueva funcionalidad: haciendo entregas frecuentes.
- Refactorización del código: reescribir cierta parte del código para aumentar su legibilidad y mantenimiento.
- Propiedad del código compartida: no se dividen las responsabilidades en el desarrollo de cada modulo. Este método promueve que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los errores serán detectados.
- Simplicidad en el código: es mas sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.

1.4.2 Lenguaje de modelado

UML

El lenguaje unificado de modelado (UML) prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos. Es capaz de describir la semántica general de los diagramas y los significados de los símbolos utilizados, actualmente es el más usado. UML desarrolla un lenguaje grafico para visualizar, especificar y documentar las variadas partes que incluye el desarrollo de software.

Sus características fundamentales son:

- Lenguaje de modelado orientado a objetos

- Viabilidad en la corrección de errores

1.4.3 Herramienta CASE para el modelado

Las herramientas CASE (Ingeniería de Software Asistida por Computadora) son aplicaciones informáticas que facilitan el trabajo dentro del ciclo de desarrollo del software, se emplean para aumentar la productividad del software disminuyendo los tiempos de construcción y el costo de los mismos. La tecnología CASE supone la automatización del desarrollo del software, contribuyendo a mejorar la calidad en el desarrollo de sistemas de información.

Visual Paradigm

Es un software privativo, para modelado en UML. Posee características gráficas muy cómodas, que facilitan la realización de los diagramas de modelado siguiendo el estándar de UML, los mismos son: diagramas de clases, casos de uso, secuencia, estado, componentes, actividad, comunicación.

Ventajas:

- Entorno de creación de diagramas para UML 2.0.
- Diseño centrado en casos de uso y enfocados al negocio generando un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones.
- Disponibilidad en múltiples plataformas.
- Producto de calidad.
- Soporta aplicaciones Web.
- Varios idiomas.

- Fácil de instalar y actualizar.
- Compatibilidad entre ediciones.

1.4.4 Fundamentación de la herramienta, lenguaje y metodología a utilizar para el modelado

Dentro de las metodologías de desarrollo de software se encuentran dos grandes grupos, las metodologías ágiles o ligeras y las pesadas, aunque ambas están encaminadas a obtener como resultado un producto de software con calidad poseen diferencias notables pues las primeras proponen el mejoramiento del software mediante la comunicación inmediata y directa y las segundas obtienen sus resultados a través de orden y documentación, dado que el sistema propuesto será implementado por un equipo de desarrollo conformado por dos personas y el mismo se realizara mediante iteraciones para obtener como resultado un incremento ejecutable donde predominan reuniones a lo largo de todo el proyecto y una interacción del cliente con los desarrolladores evitando documentaciones extensas, se propone para la implementación del sistema una metodología ágil de desarrollo dentro de las cuales se encuentran SCRUM,XP, SXP , dada las necesidades reales del problema en cuestión se ha decidido utilizar a XP como metodología de desarrollo, pues sus características responden a la solucionar el problema. Como herramienta a utilizar para el modelado fue seleccionado Visual Paradigm que aunque es un software propietario la Universidad posee la licencia para su utilización. Esta herramienta tiene características gráficas muy cómodas, que facilitan la realización de los diagramas de modelado que sigue el estándar UML. También posee cualidades como la generación automática de diagramas a partir de descripciones de casos de uso, permitiendo la agilidad en el trabajo del analista. Usa un lenguaje estándar común para todo el equipo de desarrollo que facilita la comunicación. Como lenguaje para utilizar en el modelado se escogió UML, ya que este permite visualizar, construir y documentar los artefactos de un proyecto de software. Además, es el lenguaje que utiliza la herramienta escogida para realizar el modelado.

1.5 Arquitectura

Modelo Vista Controlador (MVC):

El modelo Vista Controlador es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se

ve frecuentemente en aplicaciones web, donde la vista son las páginas HTML y el código que provee de datos dinámicos a la página; el modelo es la representación de los datos con que trabaja la aplicación y la lógica de negocio; y el controlador es el responsable de recibir los eventos de entrada desde la vista.

Modelo: es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El modelo no tiene conocimiento específico de los controladores o de las vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el modelo y sus vistas, y notificar a las vistas cuando cambia el modelo.

Vista: es el objeto que maneja la presentación visual de los datos representados por el modelo. Genera una representación visual del modelo y muestra los datos al usuario. Interactúa con el modelo a través de una referencia al propio modelo.

Controlador: es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del modelo o por alteraciones de la vista. Interactúa con el modelo a través de una referencia al propio modelo.

1.5.1 Fundamentación de la arquitectura a utilizar

Se utilizara un patrón clásico del diseño web conocido como arquitectura Modelo Vista Controlador (MVC), ya que el marco de trabajo Symfony que será utilizado en el desarrollo del sistema está basado arquitectónicamente en este patrón.

1.6 Herramientas, tecnologías y lenguajes para el desarrollo

1.6.1 Marco de trabajo

Un marco de trabajo no es más que una estructura de soporte definida mediante la cual otro proyecto de software puede ser desarrollado y organizado. Puede incluir soporte de programas, biblioteca y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Representa una arquitectura de software que modela las relaciones generales de las

entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utilizan las aplicaciones del dominio.

Son diseñados para facilitar el desarrollo de software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software que tratando con los tediosos detalles de bajo nivel de proveer un sistema funcional. Es el esqueleto sobre el cual varios objetos son integrados para una solución dada. No es más que una base de programación que atiende a sus descendientes posibilitando cualquier respuesta ante las necesidades de sus miembros, o secciones de una aplicación web.

Symfony

Symfony es un completo marco de trabajo diseñado para optimizar el desarrollo de aplicaciones web mediante algunas de sus características. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de estas ventajas es que no se debe reinventar la rueda cada vez que se crea una aplicación web.[7]

Symfony está desarrollado completamente con PHP 5. Ha sido probado en diferentes proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Symfony es compatible con la mayoría de los gestores de bases de datos, como: MySQL, PostgreSQL, Oracle y Microsoft SQL Server. Se puede ejecutar tanto en plataformas Unix, Linux, como en plataformas Windows. Posee una serie de características enunciadas a continuación: [7]

Características:

- Fácil de instalar y configurar en la mayoría de las plataformas.
- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de los casos, pero lo suficientemente flexible para adaptarse en los casos mas complejos.
- Basado en la premias de “convenir en vez de configurar”, en la que el desarrollador solo debe configurar aquello que no es convencional.

- Sigue la mayoría de las mejores prácticas y patrones de diseño para la web.
- Preparado para aplicaciones empresariales y adaptables a las políticas y arquitectura propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento sencillo.
- Fácil de extender, lo que permite su integración con las librerías de otros fabricantes.

1.6.2 Lenguaje para el desarrollo Web

Las aplicaciones sobre la web mantienen un auge en todo el mundo gracias a las ventajas que las mismas ofrecen a las empresas e instituciones. Para desarrollar estas aplicaciones existen múltiples lenguajes informáticos divididos en dos grupos, el primer grupo abarca los lenguajes que corren del lado del cliente y el segundo lo conforman aquellos lenguajes que corren del lado del servidor. Las técnicas de desarrollo web más utilizadas del lado del cliente son HTML y Java Script, mientras que los lenguajes de programación del lado del servidor más usados en software libre son Java, PHP y PERL.

PHP

PHP es un lenguaje de programación estructurado, fue diseñado originalmente para la creación de páginas web dinámicas aunque últimamente ha intervenido en la creación de aplicaciones con interfaces gráficas usando bibliotecas específicas.

PHP fue creado originalmente por Rasmus Lerdorf en 1994; sin embargo su implementación principal es producida ahora por el grupo de desarrollo de PHP y sirve como el estándar por defecto para PHP al no existir una especificación formal. Publicado bajo la Licencia PHP, la fundación de software libre considera esta licencia como software libre.[8]

Es capaz de combinarse con servidores de bases de datos como MySQL, PostgreSQL, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite; lo cual permite la creación de aplicaciones web robustas y arroja resultados muy interesantes y prometedores para aquellas paginas que desean ser

activas y dinámicas. La ejecución e interpretación de PHP es completamente del lado del servidor, en este se encuentra almacenado el script por lo que el cliente solo puede recibir el resultado de la ejecución. PHP puede ser ejecutado en la mayoría de los sistemas operativos, tales como UNIX, Linux, Mac OSX y Windows, a la vez es capaz de interactuar con servidores web muy populares como ISAPI y Apache por existir también en versión CGI. [8]

Ventajas de PHP:

- Es capaz de leer y manipular datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML.
- Cuenta con una biblioteca sumamente amplia por defecto de funciones.
- No requiere definición de tipos de variables ni manejo detallado de bajo nivel.
- Tiene capacidad de conexión con la mayoría de los manejadores de bases de datos utilizados en la actualidad.
- Posee una amplia documentación en su web oficial de Internet. En la misma se encuentran muy bien explicadas todas las funciones del sistema, contando con ejemplos detallados.
- Pertenece a la alternativa de código abierto por lo que se presenta como una elección de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Permite crear formularios para la web.

1.6.3 Servidor Web Apache

El servidor Web Apache es uno de los mayores triunfos del software libre. Entre sus características fundamentales se pueden destacar:

- Es flexible, rápido y eficiente.
- Continuamente actualizado y evoluciona a mayor velocidad.
- Multiplataforma.
- Se desarrolla de forma abierta.

- Modular: puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos.
- Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos. [9]

1.6.4 Object Relational Mapper

Doctrine es un potente y completo sistema Object Relational Mapper (ORM) para PHP 5.2 o superior, con una capa de abstracción de datos incorporado. Entre muchas otras cosas tiene la posibilidad de exportar una base de datos existente a sus clases correspondientes y también a la inversa, es decir, convertir clases (convenientemente creadas siguiendo las pautas del ORM) a tablas de una base de datos.

Su principal ventaja radica en poder acceder a la base de datos utilizando la programación Orientada a Objetos (POO) debido a que doctrine utiliza el patrón Active Record para manejar la base de datos, tiene su propio lenguaje de consultas y trabaja de manera rápida y eficiente. Es fácilmente integrable a los principales marcos de trabajo de desarrollo utilizados actualmente.

1.6.5 NetBeans

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, tiene una comunidad que está en constante crecimiento, y con cerca de 100 socios en todo el mundo. La plataforma NetBeans es una base modular y extensible usada como una estructura de integración que permite para crear aplicaciones web, de escritorio, para móviles y más, a partir de distintos lenguajes de programación como son PHP, Java, Java Script, Phyton entre otros y funciona sobre distintos sistemas operativos como OpenSolaris, Linux, Windows y Mac.

Entre las características de la plataforma están:

- Administración de las interfaces de usuario (ej. menús y barras de herramientas)

- Administración de las configuraciones del usuario
- Administración del almacenamiento (guardando y cargando cualquier tipo de dato)
- Administración de ventanas
- Marco de trabajo basado en asistentes (diálogos paso a paso)

NetBeans 6.8

La nueva versión que trae NetBeans como entorno de desarrollo es la 6.8, la primera que soporta Java EE 6. Pero lo más interesante es la integración total que trae con el lenguaje de programación PHP y más específicamente con el marco de trabajo Symfony. Esta característica permite el desarrollo ágil debido a un gran auto completamiento de código por parte del entorno de trabajo, de forma que ahorra tiempo a la hora de programar y evita errores comunes de tipografía a los desarrolladores. Por otra parte, esta nueva versión de NetBeans permite integrarse con el servidor web Apache durante el tiempo de ejecución en una aplicación web alojada en este servidor.[10]

1.6.6 Sistema Gestor de base de datos

Los sistemas de gestión de bases de datos son un software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones utilizadas. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y un lenguaje de consulta. El propósito general de un sistema de gestión de bases de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para un buen manejo de datos. Entre los más utilizados están: PostgreSQL y MySQL. [11]

PostgreSQL:

Es un servidor de base de datos relacional orientado a objetos de software libre. Posee las características de los más potentes sistemas comerciales como Oracle o SQL Server, con la ventaja de que su licencia es gratuita. Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y comercializadores. Dicha comunidad es denominada grupo de desarrollo mundial de PostgreSQL, tiene más de 15 años de desarrollo activo y se ha ganado la reputación de ser confiable y mantener la integridad de los datos.[11]

Ventajas:

- Alta concurrencia
- Amplia variedad de tipos nativos
- Multiplataforma
- Extensible
- Estabilidad y confiabilidad
- Instalación ilimitada
- Diseñado para ambientes de alto volumen

1.6.7 Fundamentación de las herramientas, tecnología y lenguaje a utilizar para el desarrollo del sistema

Para el desarrollo se utilizara como lenguaje de programación PHP, pues está diseñado principalmente para el desarrollo web, controla el manejo de excepciones, posee una biblioteca de funciones sumamente amplia y es uno de los lenguajes más utilizados actualmente. Además, posee una amplia comunidad de desarrollo, la cual implementa constantemente mejoras en su código.

Para el desarrollo del sistema se utilizara como marco de trabajo: Symfony ya que posee herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web, permite la automatización de las tareas más comunes lo que facilita al desarrollador dedicarse por completo a los aspectos específicos de la aplicación, tiene un código fácil de leer y puede dársele mantenimiento de forma sencilla por lo que se considera optimo para darle solución al problema propuesto.

Como sistema gestor de base de datos se utilizara PostgreSQL, ya que es uno de los gestores de bases de datos relacionales de software libre más utilizado en el mundo, posee las características de los más potentes sistemas comerciales como Oracle o SQL Server, es multiplataforma y permite un gran volumen de datos

.

1.7 Propuestas de herramientas seleccionadas

Con lo explicado y analizado en este capítulo se ha decidido que es necesario realizar el análisis, diseño, implementación de una aplicación web que permita mejorar la gestión de los medios y bienes en los laboratorios de la facultad 1, la aplicación debe lograr agilidad en el manejo de los datos y una interfaz amigable que responda a las necesidades del cliente, la decisión de utilizar las siguientes herramientas fue tomada siguiendo fundamentalmente la política de uso con soporte multiplataforma y licencias de utilización libre.

Por lo que se propone utilizar PHP como lenguaje del lado del servidor, por todas las características que presenta, además de ser el lenguaje de programación en el que está desarrollado Symfony, que será utilizado como herramienta de desarrollo para apoyar el proceso de implementación del sistema. Se propone además XP como metodología de desarrollo de software, como sistema gestor de base de datos PostgreSQL, por sus características antes expuestas. Como herramienta de modelado en UML el Visual Paradigm.

1.8 Conclusiones

El estudio de las diferentes metodologías de desarrollo de software, lenguajes de modelado, herramientas CASE, lenguajes de programación web y sistemas gestores de bases de datos permitió conocer cuales son las que se ajustan para el desarrollo de la solución informática para la Gestión de medios y bienes. A través del análisis de las diferentes herramientas se conocieron las más adecuadas a utilizar teniendo en cuenta las características y condiciones del cliente y del equipo de desarrollo. El análisis de algunas soluciones informáticas existentes, que gestionan recursos demuestra la necesidad de la solución informática para la gestión de medios y bienes en los laboratorios de la facultad 1.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1. Introducción

En este capítulo se presenta parte de la propuesta de solución de la investigación. Apoyándose en la metodología usada, se desarrollan varias actividades y artefactos que permiten darle cumplimiento al problema planteado, y de esta forma, dominar el negocio como comienzo de un proceso que continuará con la especificación de los requisitos y con el diseño del sistema.

2.2. Breve exposición de la situación actual.

El área de los laboratorios actualmente cuenta con una gran parte de los medios y bienes existentes en la facultad, para el desarrollo docente y productivo. Resultando difícil llevar un control eficiente de los mismos. Debido a que:

- El registro de los medios se realiza de forma manual, empleando modelos de papel que pueden deteriorarse o perderse.
- Los movimientos no se desarrollan con la eficiencia requerida, propiciando el descontrol y la difícil actualización de sus datos en el área correspondiente.
- La ubicación de los medios se controla tanto a nivel central como de área, provocando la existencia de información duplicada, con errores y difícil de actualizar.
- Los reportes de entrega son archivados en documentos Excel, dificultando la obtención de datos estadísticos sobre el estado de los medios que se encuentran en los laboratorios.

Teniendo en cuenta todas las dificultades antes expresadas, se llegó a la conclusión de que el área de los laboratorios de la Facultad 1 necesita una aplicación que facilite el control de los medios y bienes con que cuenta.

2.3 Propuesta de Sistema

Con el objetivo de obtener un mejor control de los medios y bienes con que cuenta el área de los laboratorios de la Facultad 1, se ha decidido desarrollar un sistema que automatice los procesos de

gestión de inventario que se realizan de forma manual en estos momentos. El mismo brindará facilidades para almacenar y manejar la información referente a los medios y bienes de los laboratorios. A continuación se expondrán los diferentes servicios que ofrecerá dicho sistema. Para el trabajo con el sistema se han definido cuatro roles: un usuario que se puede comportar como **administrador**, el cual tendrá acceso a todas las funcionalidades del sistema, un **responsable de inventario en el área** encargado de gestionar todas las entradas de datos al sistema, un grupo de **usuarios** que se comportará como **líder de proyecto**, este tendrá acceso a determinadas funcionalidades, aplicadas solo a los **proyectos** de los cuales ellos sean líderes, las cuales serán asignadas por el administrador en su debido momento y por último el resto de los usuarios que pueden acceder al sistema en cualquier momento para obtener información brindada en este o hacer uso de otros servicios brindados en el mismo, tales como foro de discusión y encuesta. La información de todos los usuarios que interactúen con el sistema se registrará y controlará en la base de datos de este. El sistema permitirá insertar los datos de cada laboratorio, ya sea de proyecto o docencia, acción que realizará el responsable de inventario en el área, y una vez hecha la inserción el administrador podrá modificarlos y consultarlos de ser necesario, en el caso del líder de proyecto solo podrá realizar acciones de control y distribución de medios en su proyecto. Cuando el proyecto se culmine y no sea preciso tenerlo registrado el administrador podrá eliminarlo.

El sistema deberá también debe gestionar todo lo referente a los medios con que cuentan los laboratorios y las características de los mismos, el administrador podrá modificarlas así como eliminarlas. Con el objetivo de brindar una mayor información acerca de los medios y bienes, tanto el líder de proyecto como el administrador tendrán la posibilidad de acceder a los **reportes** arrojados por el inventario, el líder de proyecto tendrá derecho en todo momento de visualizar el **historial de los reportes** correspondientes a su proyecto, el administrador, podrá obtener dicho historial de cada uno de los laboratorios existentes en el sistema en todo momento. Por último, y no menos importante permitirá gestionar usuarios. El administrador podrá insertar, modificar, consultar y eliminar los datos de los usuarios, así como restringir su acceso a cada funcionalidad y su autenticación.

2.4 Personas relacionadas con el sistema

Se define como persona relacionada al sistema toda aquella que obtiene un resultado del valor de uno o varios procesos que se ejecutan en el mismo. Además de todas esas que se encuentran involucradas en dichos procesos, pues participan en ellos pero no obtienen ningún resultado de valor.

Personas relacionadas con el sistema	Justificación
Usuario	Cualquier persona que posea una cuenta en el dominio uci.cu que interactúe con el sistema, fluye dentro de este como un invitado, sin privilegios. Tiene la posibilidad de navegar sobre las diferentes opciones para que se vaya relacionando con el mismo, así como indagar en las diferentes bibliografías que se muestran de manera general.
Líder de proyecto	Usuario que está al frente de un proyecto y puede realizar acciones de consulta sobre los recursos con que cuenta.
Administrador	Es la persona facultada para la gestión del sistema. Es el encargado de administrar las diferentes cuentas de los usuarios autenticados en la aplicación. Además, tendrá la responsabilidad de realizar los movimientos de medios, distribuir proyectos y generar los partes diarios, manteniendo un control total de las acciones llevadas a cabo en la aplicación.
Responsable de inventario en el área.	Es la persona encargada de la entrada de datos al sistema

Tabla 1. Personas relacionadas con el sistema

2.5 RF (Requerimientos funcionales del sistema)

RF 1. Autenticar usuario.

RF 2. Gestionar usuario

2.1 Insertar usuario

2.2 Eliminar usuario

2.3 Buscar usuario

2.4 Modificar usuario

RF 3. Gestionar laboratorios

3.1 Insertar laboratorios

- 3. 2 Eliminar laboratorios
- 3. 3 Modificar laboratorios
- 3. 4 Mostrar laboratorio
- RF 4. Gestionar nomenclador
 - 4. 1 Insertar nomenclador
 - 4. 2 Eliminar nomenclador
 - 4. 3 Modificar nomenclador
- RF 5. Gestionar medios
 - 5. 1 Insertar medios
 - 5. 2 Eliminar medios
 - 5. 3 Modificar medios
 - 5. 4 Mostrar medios
- RF 6. Gestionar proyecto
 - 6. 1 Insertar proyecto
 - 6. 2 Eliminar proyecto
 - 6. 3 Modificar proyecto
 - 6. 4 Mostrar proyecto
- RF 7. Distribuir proyecto
- RF 8. Realizar movimientos de medios
- RF 9. Registrar parte diario
- RF 10. Realizar búsquedas en el sistema
 - 10. 1 Dado un laboratorio buscar los datos de los medios con que cuenta el laboratorio.
 - 10. 2 Dado un laboratorio y un proyecto buscar los datos de los medios del proyecto
- RF 11. Obtener reportes

2.6 Requerimientos no funcionales del sistema

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

Los requisitos no funcionales pueden ser más críticos que los funcionales, puesto que si un requisito funcional no se cumple, el sistema se degrada, pierde eficacia, y puede no responder a la totalidad de los requerimientos del usuario, pero en cambio si un requisito no funcional no se cumple, el sistema puede inutilizarse.

Entre los requerimientos no funcionales del sistema propuesto se encuentran:

Usabilidad:

- Para utilizar el sistema es necesario poseer conocimientos elementales de computación y sobre el ambiente Web en sentido general.
- Tener buena visibilidad en los principales navegadores.
- El sistema debe estar disponible las veinticuatro horas del día, sin ninguna interrupción.
- El sistema debe ser accesible desde todos los puntos donde exista una máquina conectada a la red.
- El sistema debe tener una interfaz que le sea familiar al usuario para aprovechar sus conocimientos en el manejo de herramientas de software.
- El sistema debe ser una interfaz de fácil aprendizaje para que usuarios inexpertos puedan familiarizarse lo más pronto posible y le sea cómodo el manejo del software.

Confiabilidad y Seguridad:

- Chequear que el usuario esté autenticado antes de que pueda realizar alguna acción sobre el sistema.
- Garantizar que las funcionalidades del sistema se muestren de acuerdo con el tipo de usuario que esté activo.

- El sistema debe tener la capacidad de identificar con certeza a los diversos usuarios o entidades que interactúan con él.
- El sistema debe tener la capacidad que tiene el sistema de darle seguridad al usuario, de que las informaciones solo serán vistas por quien este capacitado para esto.
- El servicio del sistema debe tener una disponibilidad aceptable (99%).
- El tiempo entre fallos debe ser breve o cero, haciendo lo posible para que esto no ocurra.
- La base de datos debe estar fraccionada sobre varios esquemas, dividiendo así de una forma lógica las funcionalidades, evitando así la pérdida total de la información en caso de algún accidente o ataque.
- Permitir autenticarse a través de un Servidor de Dominio.

Rendimiento:

- El sistema debe ser capaz de mantener el mismo rendimiento y estabilidad a medida que aumenta la cantidad de datos a gestionar.
- El sistema debe ser capaz de correr al montar en una misma PC todos sus componentes tales como Gestor de base de dato, Aplicación Web, etc.

Funcionalidad:

- Capacidades de búsqueda con velocidad apropiada. El sistema debe ser Multiplataforma.

Desempeño y escalabilidad:

- El tiempo de respuestas debe ser corto con respuestas rápidas y eficientes.
- El sistema podrá soportar un gran número de clientes online.
- El sistema debe tener un rendimiento óptimo debido a que presta servicios a un gran número de usuarios.

Requerimientos de software del Sistema:

- Las computadoras clientes deben contar con Mozilla Firefox o Internet-Explorer.
- La computadora servidora debe tener instalado el SGBD MySQL.

- La computadora servidora debe tener instalado el servidor Web Apache.
- La computadora servidora debe tener instalado el Marco de trabajo Symfony y PHP5.

Requerimientos de hardware del sistema:

- La computadora servidor debe tener como mínimo 512 MB de memoria RAM como mínimo.
- Las computadoras clientes deben estar conectadas en red.
- Las computadoras clientes deben ser Pentium III o superior.
- El servidor debe ser Pentium III o superior.
- El servidor debe tener Intel(R) PRO/1000 PM Network Connection.

2.7 Fase de exploración. Definición

El ciclo de vida de XP enfatiza en el carácter interactivo e incremental del desarrollo. Una iteración de desarrollo es un período de tiempo en el que se realiza un conjunto de funcionalidades determinadas, que en el caso de XP corresponden a un conjunto de historias de usuarios. [4]

La metodología de desarrollo XP comienza con la fase de exploración, en esta fase los clientes plantean a grandes rasgos las historias de usuario mediante un proceso de identificación que son de interés para la primera entrega del producto. Al mismo tiempo, el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto.

Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

2.7.1 Historias de Usuario

Uno de los artefactos más importantes que genera la metodología XP son las historias de usuario (HU). Estas tienen el mismo propósito que los casos de uso y son escritas por el propio cliente, tal y como

ven ellos las necesidades del sistema, por tanto, son descripciones cortas y escritas en el lenguaje del usuario sin terminología técnica.

Las historias de usuarios conducen al proceso de creación de las pruebas de aceptación, los cuales servirán para verificar que estas historias se han implementado correctamente. Otra de sus características es que solamente proporcionan los detalles sobre la estimación del riesgo y cuánto tiempo conllevará su implementación.[12]

Su nivel de detalle debe ser el mínimo posible, de manera que permita hacerse una ligera idea de cuánto costará implementar el sistema.

Durante el análisis en la fase de exploración se identificaron once historias de usuarios, cada una de ellas respondiendo a las diferentes funcionalidades solicitadas por el cliente y dando una idea al resto del equipo de desarrollo de cómo debe ser su posterior implementación. Estas se describen a continuación:

Historia de Usuario	
Número: 1	Nombre Historia de Usuario: Autenticar usuario
Usuario: Usuario	
Prioridad en Negocio: Alta	Iteración Asignada: 1
Riesgo en Desarrollo: Medio	Puntos Estimados: 1
Descripción: Se brinda la posibilidad de que la persona que accede al sistema introduzca sus datos (usuario y contraseña) con la finalidad de verificar y otorgarle los permisos según el rol que cumpla dentro de la aplicación. Los datos asociados a cada cuenta de usuario van a coincidir con los utilizados por este para acceder al dominio UCI. CU.	
Observaciones: RF1	

Tabla 2. Historia de usuario "Autenticar usuario".

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

Usuario: Administrador, Responsable de inventario en el área	
Prioridad en Negocio: Alta	Iteración Asignada: 1
Riesgo en Desarrollo: Medio	Puntos Estimados: 1
Descripción: El administrador del sistema puede insertar, modificar, eliminar y buscar usuarios dentro del mismo, además de asignarle los permisos correspondientes.	
Observaciones: RF2	

Tabla 3. Historia de usuario “Gestionar usuario”.

Historia de Usuario	
Número: 3	Nombre Historia de Usuario: Gestionar Laboratorio
Usuario: Administrador, Responsable de inventario en el área	
Prioridad en Negocio: Alta	Iteración Asignada: 1
Riesgo en Desarrollo: Medio	Puntos Estimados: 1
Descripción: El administrador puede acceder a las opciones para insertar, modificar, buscar o eliminar del menú de operaciones todos los datos de un laboratorio y acceder a los mismos cuando desee con el objetivo de modificarlos. El responsable de inventario en el área solo se encargará de insertar los datos de los laboratorios al sistema.	
Observaciones: RF3	

Tabla 4. Historia de usuario “Gestionar laboratorio”.

Historia de Usuario	
Número: 4	Nombre Historia de Usuario: Gestionar nomenclador
Usuario: Administrador	
Prioridad en Negocio: Alta	Iteración Asignada: 1
Riesgo en Desarrollo: Bajo	Puntos Estimados: 1
Descripción: Se realizan las acciones de insertar, eliminar y modificar nomenclador. El administrador tiene la	

posibilidad de realizar estas acciones cuando sea necesario para mantener valores constantes dentro de la aplicación.

Observaciones: RF4

Tabla 5. Historia de usuario “Gestionar nomenclador”.

Historia de Usuario	
Número: 5	Nombre Historia de Usuario: Gestionar Medio
Usuario: Administrador, Responsable de inventario en el área	
Prioridad en Negocio: Alta	Iteración Asignada: 2
Riesgo en Desarrollo: Medio	Puntos Estimados: 1
Descripción: Se realizan las acciones de insertar, eliminar así como buscar medios en los laboratorios. El administrador tiene la posibilidad de insertar todos los datos relacionados con un medio, así como acceder a estos para modificarlos o eliminarlos en caso de ser necesario. En el caso del responsable de área solo se encargará de insertar datos.	
Observaciones: RF5	

Tabla 6. Historia de usuario “Gestionar Medio”.

Historia de Usuario	
Número: 6	Nombre Historia de Usuario: Gestionar proyecto
Usuario: Administrador	
Prioridad en Negocio: Alta	Iteración Asignada: 2
Riesgo en Desarrollo: Bajo	Puntos Estimados: 1
Descripción: Se realizan las acciones de crear, modificar y eliminar proyectos productivos. El administrador tiene la posibilidad de insertar todos los datos relacionados al proyecto, así como acceder a estos cada vez que desee con el objetivo de modificarlos en caso que sea necesario o eliminar estos del sistema después de culminado el trabajo de dicho proyecto.	
Observaciones: RF6	

Tabla 7. Historia de usuario “Gestionar proyecto”.

Historia de Usuario	
Número: 7	Nombre Historia de Usuario: Distribuir proyecto
Usuario: Administrador	
Prioridad en Negocio: Alta	Iteración Asignada: 2
Riesgo en Desarrollo: Bajo	Puntos Estimados: 1
Descripción: Se realiza la acción de distribuir los proyectos por laboratorios, en un mismo laboratorio puede existir más de un proyecto.	
Observaciones: RF7	

Tabla 8. Historia de usuario “Distribuir proyecto”.

Historia de Usuario	
Número: 8	Nombre Historia de Usuario: Realizar movimientos de medio
Usuario: Administrador	
Prioridad en Negocio: Alta	Iteración Asignada: 2
Riesgo en Desarrollo: Bajo	Puntos Estimados: 1
Descripción: El administrador accede a las opciones del menú para eliminar medios de un laboratorio e insertarlos en otro o darles baja por deterioro especificándose el código del movimiento, el tipo de medio a mover y su cantidad física.	
Observaciones: RF8	

Tabla 9. Historia de usuario “Realizar movimiento de medio”.

Historia de Usuario	
Número: 9	Nombre Historia de Usuario: Registrar parte diario
Usuario: Administrador	
Prioridad en Negocio: Alta	Iteración Asignada: 3

Riesgo en Desarrollo: Medio	Puntos Estimados: 1
Descripción: Se realiza un control de los medios del laboratorio a través de un modelo de parte donde se especifican los medios existentes y su estado actual.	
Observaciones: RF9	

Tabla 10. Historia de usuario “Registrar parte diario”.

Historia de Usuario	
Número: 10	Nombre Historia de Usuario: Realizar búsquedas en el sistema
Usuario: Usuario	
Prioridad en Negocio: Media	Iteración Asignada: 3
Riesgo en Desarrollo: Bajo	Puntos Estimados: 1
Descripción: El usuario podrá realizar búsquedas de información en el sistema cuando lo desee, siempre que tenga permiso para acceder a esta. El administrador podrá buscar los datos relacionados con los medios de los laboratorios, en tanto el líder de proyecto podrá hacer lo mismo pero con los medios de su proyecto.	
Observaciones: RF10	

Tabla 11. Historia de usuario “Realizar búsquedas en el sistema”.

Historia de Usuario	
Número: 11	Nombre Historia de Usuario: Obtener reportes
Usuario: Administrador, Líder de proyecto	
Prioridad en Negocio: Media	Iteración Asignada: 3
Riesgo en Desarrollo: Bajo	Puntos Estimados: 1
Descripción: El administrador podrá obtener cualquier reporte que desee relacionado con el sistema, mientras el líder de proyecto solo podrá obtener reportes relacionados con su proyecto.	
Observaciones: RF11	

Tabla 12. Historia de usuario “Obtener reportes”.

2.8 Fase de Planificación. Definición

En esta fase el cliente establece la prioridad de cada historia de usuario, los programadores realizan una estimación del esfuerzo necesario para la implementación de cada una de ellas. Las estimaciones de esfuerzo asociado a la implementación de las historias las establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos. Por otra parte, el equipo de desarrollo mantiene un registro de la “velocidad” de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración. La planificación se puede realizar basándose en el tiempo o el alcance. La velocidad del proyecto es utilizada para establecer cuántas historias se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de historias.

2.8.1 Estimación del esfuerzo por historia de usuario

Se ha realizado una estimación para cada una de las historias de usuario previamente identificadas, con el objetivo de llegar a realizar un buen desarrollo del sistema propuesto, los resultados de dicha estimación se muestran a continuación:

Historias de Usuario	Puntos estimados
Autenticar usuario	1
Gestionar usuario	1
Gestionar laboratorio	1
Gestionar nomenclador	1
Gestionar medio	1
Gestionar proyecto	1
Distribuir proyecto	1
Realizar movimientos de medios	1
Registrar parte diario	1
Realizar búsquedas en el sistema	1

Obtener reportes	1
Total	11

Tabla 13. Plan de estimación de esfuerzo por historias de usuario.

2.8.2 Plan de Iteraciones

Todo proyecto que siga la metodología XP se ha de dividir en iteraciones de aproximadamente 3 semanas de duración, por lo tanto, luego de la previa identificación y descripción de las historias de usuario y realizar la estimación de esfuerzo de cada una de ellas se procede a la planificación de la etapa de implementación del sistema, donde se definen las historias de usuarios que serán implementadas en cada iteración del proyecto así como las posibles fechas de culminación. Se decidió realizar un total de tres iteraciones que abarcaran la implementación de la totalidad de historias de usuario, cada una de estas se define a continuación:

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, laboratorios y nomenclador.

Iteración 2

Durante esta iteración se implementan las restantes funcionalidades con prioridad alta, al terminar esta se habrá implementado lo relacionado con las historias de usuario 5,6,7 y 8 en las cuales se relacionan con la gestión de medios, proyecto, distribuir proyectos además de realizar movimientos de medios. La versión del producto referente a esta iteración junto a las implementaciones anteriores, serán mostradas al cliente con el objetivo de realizar cambios en base a la opinión del mismo.

Iteración 3

En esta iteración serán implementadas el resto de las funcionalidades, estas hacen alusión a todo lo relacionado con la distribución de los medios del proyecto, los reportes asociados a los medios y

bienes y las búsquedas realizadas en el sistema. Estas funciones están descritas en las historias de usuario 9,10 y 11. 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.

2.8.3 Plan de duración de las 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 cuales 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.[5]

Iteración	Orden de las HU a implementar	Duración total de la Iteración
Iteración # 1	<ol style="list-style-type: none">1. Autenticar usuario.2. Gestionar usuario.3. Gestionar laboratorio.4. Gestionar nomenclador	4 semanas
Iteración # 2	<ol style="list-style-type: none">1. Gestionar medios.2. Gestionar proyectos.3. Distribuir proyecto4. Realizar movimientos de medios.	4 semanas
Iteración # 3	<ol style="list-style-type: none">1. Registrar parte diario.2. Realizar búsquedas en el sistema.3. Obtener reporte.	3 semanas

Tabla 14. Plan de iteraciones.

2.8.4 Plan de entregas

En el plan de entrega que se plantea a continuación se hace una propuesta de la fecha aproximada en que se harán las versiones (releases) del sistema. Al finalizar cada iteración en la fase de implementación.

Entregable	Final 1ra Iteración 3ra semana de marzo	Final 2da Iteración 3ra semana de abril	Final 3ra Iteración 2da semana de mayo
Sistema para la gestión de inventario en los laboratorios de la Facultad1.	0.1	0.2	1.0

Tabla 15. Plan de entregas.

2.9 Conclusiones

En este capítulo han sido analizados los procesos fundamentales que están relacionados con la solución informática a desarrollar; la gestión de inventario en los laboratorios de la Facultad 1. Se elaboró además una propuesta que cumple con las funcionalidades acordes a las necesidades del cliente. Fueron definidas once historias de usuarios, facilitando una mejor comprensión de las funcionalidades que debe cumplir el sistema, así como el orden en las que deben ser implementadas las mismas para su desarrollo.

CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN

3.1 Introducción

En el presente capítulo será abordado todo lo referente al diseño e Implementación del sistema, obteniendo en la primera etapa una correcta modelación de las clases del diseño a través de las tarjetas CRC(Clase, Responsabilidad y Colaboración) y los modelos lógicos y físicos de la base de datos, en la segunda etapa serán implementadas las historias de usuarios comprendidas en cada iteración mediante la obtención de tareas de implementación generadas por cada una de ellas, se obtendrá también el diagrama de despliegue y un grupo de tareas generales imprescindibles para el correcto funcionamiento del sistema.

3.2 Diseño

Durante el diseño de la solución, y de cada historia de usuario que lo requiera, la máxima simplicidad posible es la clave para el éxito de XP. Se debe tener en cuenta que un diseño complejo siempre tarda más en desarrollarse que uno simple, y que siempre es más fácil añadir complejidad a un diseño simple que quitarla de uno complejo, siempre se deben hacer las cosas lo más sencillas posible, evitando añadir una funcionalidad no contemplada a una iteración.

3.2.1 Tarjeta CRC

Para el diseño del sistema XP sugiere llevar a cabo reuniones entre todos los desarrolladores implicados, donde se toman las decisiones mediante el uso de tarjetas CRC. Cada tarjeta representa un objeto del sistema, en la que su nombre aparece escrito en la parte superior, sus responsabilidades en la parte izquierda y los objetos con los que colabora en la parte derecha. [13]

A continuación se muestran las tarjetas CRC correspondientes a las clases utilizadas en la implementación del sistema.

Nombre de la clase: Ccontroladora	Superclase:	Subclase:
Responsabilidad Principal: Gestión de las tareas del sistema		
Responsabilidades	Colaboraciones	
Autenticar usuario	CUsuario, Modelo	
Gestionar usuario	CUsuario, Modelo	
Gestionar laboratorio	CUsuario, CLaboratorio, Modelo	
Gestionar nomenclador	Modelo	
Gestionar medios	CUsuario, Modelo	
Gestionar proyecto	CUsuario, CLaboratorio, CProyecto, Modelo	
Distribuir proyecto	CUsuario, CLaboratorio, CProyecto, Modelo	
Realizar movimientos de medios	CUsuario, CLaboratorio, CProyecto, CMedio, Modelo	
Registrar parte diario	CUsuario, CLaboratorio, CProyecto, CMedio, Modelo	
Realizar búsquedas en el sistema	CUsuario, CLaboratorio, CProyecto, CMedio, Modelo	
Obtener reportes	CUsuario, CLaboratorio, CProyecto, CMedio, Modelo	

Tabla 16. Tarjeta CRC correspondiente a la clase CControladora.

Nombre de la clase: Cusuario	Superclase:	Subclase:
Responsabilidad Principal: Gestión de usuarios		
Responsabilidades	Colaboraciones	
Autenticar usuario	Modelo	
Insertar usuario	Modelo	
Eliminar usuario	Modelo	
Buscar usuario	Modelo	
Modificar usuario	Modelo	

Tabla 17. Tarjeta CRC correspondiente a la clase CUsuario.

Nombre de la clase: Claboratorio	Superclase:	Subclase:
Responsabilidad Principal: Gestión de laboratorios		

Responsabilidades	Colaboraciones
Insertar laboratorio	CUusuario, Modelo
Eliminar laboratorio	CUusuario, Modelo
Mostrar laboratorio	CUusuario, Modelo
Modificar laboratorio	CUusuario, Modelo

Tabla 18. Tarjeta CRC correspondiente a la clase CLaboratorio.

Nombre de la clase:	Superclase:	Subclase:
Cnomenclador		
Responsabilidad Principal: Gestión de nomencladores		
Responsabilidades	Colaboraciones	
Insertar nomenclador	Modelo	
Eliminar nomenclador	Modelo	
Modificar nomenclador	Modelo	

Tabla 19. Tarjeta CRC correspondiente a la clase CNomenclador.

Nombre de la clase:	Superclase:	Subclase:
Cmedio		
Responsabilidad Principal: Gestión de medios		
Responsabilidades	Colaboraciones	
Insertar medio	CUusuario, Modelo	
Eliminar medio	CUusuario, Modelo	
Modificar medio	CUusuario, Modelo	
Mostrar medio	CUusuario, Modelo	

Tabla 20. Tarjeta CRC correspondiente a la clase CMedio.

Nombre de la clase:	Superclase:	Subclase:
Cproyecto		
Responsabilidad Principal: Gestión de proyectos		
Responsabilidades	Colaboraciones	
Insertar proyecto	CUusuario, CLaboratorio, CProyecto, Modelo	
Eliminar proyecto	CUusuario, CLaboratorio, CProyecto, Modelo	
Modificar proyecto	CUusuario, CLaboratorio, CProyecto, Modelo	
Mostrar proyecto	CUusuario, CLaboratorio, CProyecto, Modelo	

Tabla 21. Tarjeta CRC correspondiente a la clase CProyecto.

Nombre de la clase: Cdistribución	Superclase:	Subclase:
Responsabilidad Principal: Distribuir los proyectos en los laboratorios		
Responsabilidades	Colaboraciones	
Asignar proyecto	CUsuario, CLaboratorio, CProyecto, Modelo	

Tabla 22. Tarjeta CRC correspondiente a la clase CDistribución.

Nombre de la clase: CBúsqueda	Superclase:	Subclase:
Responsabilidad Principal: Realzar búsquedas de información en el sistema		
Responsabilidades	Colaboraciones	
Mostrar búsqueda	CUsuario, CLaboratorio, CProyecto, CMedio, Modelo	

Tabla 23. Tarjeta CRC correspondiente a la clase CBúsqueda.

Nombre de la clase: Creporte	Superclase:	Subclase:
Responsabilidad Principal: Obtener reportes en el sistema		
Responsabilidades	Colaboraciones	
Generar reportes	CUsuario, CLaboratorio, CProyecto, CMedio, Modelo	

Tabla 24. Tarjeta CRC correspondiente a la clase CReporte.

3.2.2 Diseño de base de datos.

En esta sección se muestran los modelos de datos lógicos y físicos correspondientes al diseño general de la base de datos, para ello es esencial dominar los siguientes conceptos:

Datos: El Dato es una representación simbólica (numérica, alfabética, algorítmica etc.), un atributo o una característica de una entidad.

Modelo de datos: es un lenguaje orientado a describir una base de datos.

Modelos de Datos Lógicos: Son orientados a las operaciones más que a la descripción de una realidad.

Modelos de Datos Físicos: Son estructuras de datos a bajo nivel implementadas dentro del propio manejador.

[Anexo 2: Modelo de datos Lógico y Físico de la base de datos.](#)

3.3 Implementación.

La implementación según plantea la metodología de desarrollo XP, debe realizarse de forma iterativa teniendo como fruto de cada iteración un producto funcional que pueda ser probado y mostrado al cliente. El equipo de desarrollo se retroalimentará con las sugerencias y críticas que haga el cliente sobre la versión culminada al final de cada iteración, en la misma se detallan las iteraciones que serán llevadas a cabo durante la implementación del sistema y se describen las tareas generadas por cada una de las historias de usuario.

A lo largo del desarrollo de las iteraciones se realizará la implementación de las historias de usuario que componen cada una de estas. Se llevará a cabo una revisión del plan de iteraciones y se modificará de ser necesario. En correspondencia con el plan de desarrollo se descomponen las historias de usuario en tareas, asignándolas después a un equipo (o una persona) responsable de su implementación. Estas tareas serán usadas por los programadores, pueden ser descritas utilizando un lenguaje técnico y no necesariamente deben ser entendibles para el cliente. Se llevarán a cabo tres iteraciones de desarrollo sobre el sistema, obteniéndose al finalizar un producto listo para su aplicación.

A continuación se muestran cada una de las iteraciones:

3.3.1 Iteración 1.

En esta iteración se implementan las historias de usuario Autenticar Usuario, Gestionar Usuario, Gestionar Laboratorio y Gestionar Nomenclador, algunas necesarias para la asignación de permisos y otras imprescindibles para la posterior implementación de otras funcionalidades necesarias para el sistema, con la culminación de ésta se desea que cualquier usuario pueda acceder a la aplicación haciendo uso de su cuenta en el dominio UCI.CU y entregar al cliente una primera versión del producto.

Historias de Usuarios	Estimación	Real
Autenticar Usuario	1	1
Gestionar Usuario	1	1
Gestionar Laboratorio	1	1
Gestionar Nomenclador	1	1
Total	4	4

Tabla 25. Historias de Usuario correspondientes a la primera iteración.

Anexo 3: Tareas de implementación generadas en la iteración 1.

3.3.2 Iteración 2.

En la segunda iteración se llevara a cabo la implementación de las historias de usuarios Gestionar medio, Gestionar proyecto, Distribuir proyecto y Realizar movimiento de medios, de nivel de prioridad alto y con una gran responsabilidad en el correcto funcionamiento del sistema, obteniéndose como resultado una segunda versión del producto la cual podrá ser mostrada al cliente.

Historia de Usuario	Estimación	Real
Gestionar Medio	1	1
Gestionar Proyecto	1	1
Distribuir Proyecto	1	1
Realizar movimiento de medio	1	1
Total	4	4

Tabla 26. Historias de Usuario correspondientes a la segunda iteración.

Anexo 4: Tareas de implementación generadas en la iteración 2.

3.3.3 Iteración 3.

En la tercera iteración se llevara a cabo la implementación de las historias de usuario Registrar parte diario, Realizar búsquedas y Obtener reportes, con el objetivo de llevar un control estricto del manejo de la información almacenada en el sistema. Como resultado de esta iteración se obtendrá una versión final del producto que será mostrada y finalmente entregada al cliente.

Historia de Usuario	Estimación	Real
Registrar Parte Diario	1	1
Realizar Búsquedas	1	1
Obtener Reportes	1	1
Total	3	3

Tabla 27. Historias de Usuario correspondientes a la tercera iteración.

Anexo 5: Tareas de implementación generadas en la iteración 3.

3.4 Tareas generales de la Implementación.

Para la implementación del sistema se llevan a cabo una serie de tareas que no se encuentran comprendidas en las historias de usuarios descritas por el cliente y que se han definido como tareas generales a realizar. Con la ejecución de las tareas generales se cumplen requisitos indispensables en el correcto funcionamiento del sistema, garantizando el control de la autenticación en la aplicación, la navegabilidad y la interacción de los usuarios con el mismo.

Tarea	
Número de tarea: 1	Número de HU:
Nombre de la tarea: Creación de la base de datos.	
Tipo de tarea: Desarrollo	Puntos estimados:
Fecha inicio: 05/03/2010	Fecha fin: 08/03/2010
Programador responsable: Yunier Vega Pérez – José Rafael Aguilar Quintana	
Descripción: Utilizando PostgreSQL como gestor, se crea la base de datos en la cual se almacenaran los datos persistentes generados durante la utilización de la aplicación.	

Tabla 28. Tarea # 1 perteneciente a las tareas generales de implementación.

Tarea	
Número de tarea: 2	Número de HU:
Nombre de la tarea: Implementación de la clase abstracta modelo.	
Tipo de tarea: Desarrollo	Puntos estimados:
Fecha inicio: 10/03/2010	Fecha fin: 12/03/2010
Programador responsable: Yunier Vega Pérez – José Rafael Aguilar Quintana	

Descripción: Se creara una clase abstracta para la construcción de consultas a la base de datos. Esta clase permitirá migrar en caso necesario a otros gestores de base de datos realizando el mínimo de cambios posible en el código de la aplicación.

Tabla 29. Tarea # 2 perteneciente a las tareas generales de implementación.

Tarea	
Número de tarea: 3	Número de HU:
Nombre de la tarea: Implementación de la clase controlador.	
Tipo de tarea: Desarrollo	Puntos estimados:
Fecha inicio: 14/03/2010	Fecha fin: 16/03/2010
Programador responsable: Yunier Vega Pérez – José Rafael Aguilar Quintana	
Descripción: Esta clase se encargara del control general de la secuencia de actividades durante la interacción de un usuario con el sistema, con ella colaboran todas las demás clases existentes, esta colaboración permite la gestión de los datos mostrados al usuario e introducido por este. Además, muestra las vistas correspondientes en cada petición.	

Tabla 30. Tarea # 3 perteneciente a las tareas generales de implementación.

Tarea	
Número de tarea: 4	Número de HU:
Nombre de la tarea: Diseño de la interfaz Web.	
Tipo de tarea: Desarrollo	Puntos estimados:
Fecha inicio: 15/04/2010	Fecha fin: 20/04/2010
Programador responsable: Yunier Vega Pérez – José Rafael Aguilar Quintana	
Descripción: Se diseñara una interfaz web funcional, creando las vistas necesarias para que los usuarios interactúen con el sistema pudiendo mostrar y entrar datos.	

Tabla 31. Tarea # 4 perteneciente a las tareas generales de implementación.

3.5 Módulos Utilizados en la Implementación.

La reutilización de código es el comportamiento y las técnicas que garantizan que una parte o la totalidad de un programa informático existente se puedan emplear en la construcción de otro programa. De esta forma, se aprovecha el trabajo anterior, se economiza tiempo, y se reduce la redundancia. La

manera más fácil de reutilizar código es copiarlo total o parcialmente desde el programa antiguo al programa en desarrollo. Pero es trabajoso mantener múltiples copias del mismo código, por lo que en general se elimina la redundancia dejando el código reusable en un único lugar, y llamándolo desde los diferentes programas. Este proceso se conoce como abstracción. La abstracción puede verse claramente en las librerías, módulos y bibliotecas de software, en las que se agrupan varias operaciones comunes a cierto dominio para facilitar el desarrollo de programas nuevos.

En el epígrafe actual se describen los módulos utilizados para la implementación del sistema propuesto:

3.5.1 Módulo Autenticator

El Módulo Autenticator es un script para la gestión de páginas restringidas sólo a usuarios registrados, con nivel de acceso y gestión de errores en el momento de autenticarse. Es ideal para zonas de administración o zonas "sólo usuarios" que requieran autenticación para acceder a las mismas. Incluye administración de usuarios (altas/bajas/modificaciones) de ejemplo. Este gestor de usuarios está programado en lenguaje PHP, usa motor de base de dato y se basa en el uso de sesiones; los cuales deben estar habilitados en el servidor que se instale.

3.5.2 Módulo LDAP

LDAP (Protocolo Ligero de Acceso a Directorios) es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red. LDAP también es considerado una base de datos (aunque su sistema de almacenamiento puede ser diferente) a la que pueden realizarse consultas. Un directorio es un conjunto de objetos con atributos organizados en una manera lógica y jerárquica. El ejemplo más común es el directorio telefónico, que consiste en una serie de nombres (personas u organizaciones) que están ordenados alfabéticamente, con cada nombre teniendo una dirección y un número de teléfono adjunto. Un árbol de directorio LDAP a veces refleja varios límites políticos, geográficos y/u organizacionales, dependiendo del modelo elegido. Los despliegues actuales de LDAP tienden a usar nombres de Sistema de Nombres de Dominio (DNS) para estructurar los niveles más altos de la jerarquía. Conforme se descende en el directorio pueden aparecer entradas que representan personas, unidades

organizaciones, impresoras, documentos, grupos de personas o cualquier cosa que representa una entrada dada en el árbol (o múltiples entradas).

Habitualmente, almacena la información de autenticación de un dominio (usuario y contraseña) y es utilizado para autenticarse aunque es posible almacenar otra información (datos de contacto del usuario, ubicación de diversos recursos de la red, permisos, certificados, etc.). De forma sintetizada se puede decir que LDAP es un protocolo de acceso unificado a un conjunto de información sobre una red.

3.5.3 Módulo Laboratorio

Este módulo garantizará las funciones correspondientes a la tercera y cuarta historia de usuario del presente trabajo. Con la implementación del mismo podrán insertarse, eliminarse, buscar y modificar datos de los laboratorios en el sistema.

3.5.4 Módulo Proyecto

Este módulo permitirá llevar a cabo las funciones comprendidas en la historia de usuario número seis referentes a la inserción de proyectos en el sistema, búsquedas de los mismos, eliminarlos cuando termine su trabajo, modificar sus datos, siempre de cada proyecto se almacena el nombre de su líder, este líder cuenta con acceso en el sistema para ver los datos relacionados a su proyecto.

3.5.5 Módulo Medio

Este módulo garantizará llevar a cabo las funciones comprendidas en las historias de usuario número cinco y ocho referente al trabajo con los medios que posee cada laboratorio, ya sea, insertar nuevos medios, dar de baja a los mismos cuando se decida, modificar sus datos, consultarlos cada vez que sea necesario para verificar sus series o comprobar el estado actual del medio, mantener el control de los movimientos de medios que se realicen y actualizar los nuevos datos en el sistema.

3.5.6 Módulo Reporte

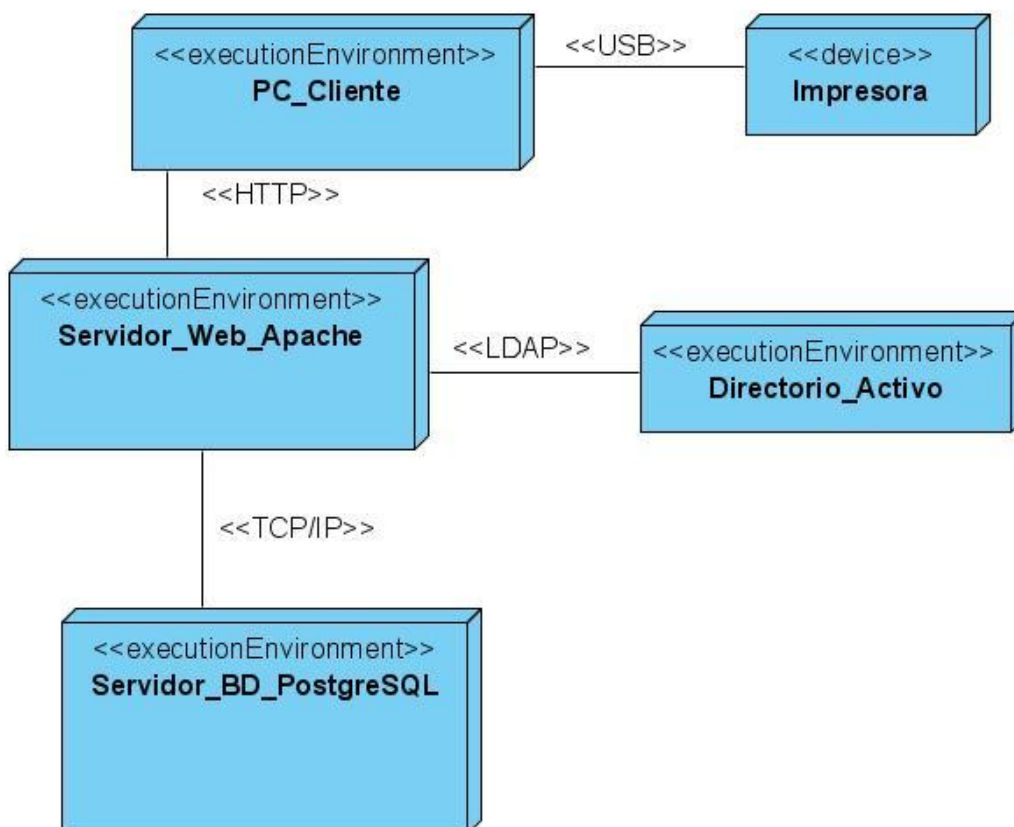
Este módulo permitirá llevar a cabo las funciones correspondientes en las historias de usuario número diez y once referente a la obtención de reportes en el sistema y la realización de búsquedas de

información, el administrador podrá mostrar e imprimir en todo momento los reportes que considere necesario para mantener el orden, cuidado y control del área.

3.5.7 Módulo Parte Diario

Este módulo garantiza llevar a cabo las funciones correspondientes a la historia de usuario número nueve referentes al registro del parte diario, este parte es un documento emitido por el jefe de área en el cual se detallan los medios reales que posee cada laboratorio y su estado, el mismo es recibido por cada técnico durante su turno de guardia el cual tiene la responsabilidad de informar cualquier anomalía acontecida en el laboratorio que cuida.

3.6 Diagrama de despliegue



3.7 Conclusiones.

En el presente capítulo se hizo referencia a las etapas de diseño e implementación del software en desarrollo exponiendo todos los artefactos generados y sus descripciones. Se hizo énfasis en la simplicidad y la refabricación, un factor saludable en la práctica de programación, para evitar el exceso de código con una lógica deficiente, un diseño innecesariamente complejo, problemas para el mantenimiento del sistema y defectos que van en contra de la calidad del producto.

CAPÍTULO 4: PRUEBAS Y ESTUDIO DE FACTIBILIDAD

4.1 Introducción

La metodología XP pone la comprobación como el fundamento del desarrollo, con cada programador escribiendo pruebas cuando escriben su código de producción. Las pruebas se integran en el proceso de integración continua y construcción lo que rinde una plataforma altamente estable para el desarrollo futuro. La programación extrema define dos tipos de pruebas las cuales son: pruebas unitarias y pruebas de aceptación, acerca de esto se abordara en este capítulo. Además de esto se desarrollara un proceso de vital de importancia dentro del proceso de desarrollo de software como lo es la estimación, la cual consiste en determinar con cierto grado de certeza los recursos necesarios para el desarrollo del mismo, ya sean recursos de hardware, software, esfuerzo, tiempo y costo. En el presente capítulo también se realizará un estudio de factibilidad para la realización del sistema propuesto mediante una estimación de tamaño, esfuerzo y planificación necesaria para llevar a cabo el mismo.

4.2 Pruebas

El proceso de desarrollo de las pruebas ayuda al cliente a clarificar y concretar la funcionalidad de la historia de usuario y favorece la comunicación entre el cliente y el equipo de desarrollo. Además, ayuda a identificar y corregir fallos u omisiones en las historias de usuario. También permite corregir errores en las ideas del cliente, por ejemplo encontrar resultados que el cliente espera encontrar en la implementación pero para los que no existe ningún camino de ejecución que conduzca a ello. Permite identificar historias adicionales que no fueran obvias para el cliente o en las que el cliente no hubiese pensado de no enfrentarse a dicha situación.

4.2.1 Pruebas unitarias

La producción de código está dirigida por las pruebas unitarias. Las pruebas unitarias son establecidas antes de escribir el código y son ejecutadas constantemente ante cada modificación del sistema. Los clientes escriben las pruebas funcionales para cada historia de usuario que deba validarse. En este

contexto de desarrollo evolutivo y de énfasis en pruebas constantes, la automatización para apoyar esta actividad es crucial.

El marco de trabajo Symfony tiene incluido su propio marco de trabajo de prueba llamado Lime, el cual proporciona soporte para las pruebas unitarias, es más eficiente que otros marcos de trabajo de pruebas de PHP y tiene las siguientes ventajas:

- Ejecuta los archivos de prueba en un entorno independiente para evitar interferencias entre las diferentes pruebas. No todos los marcos de trabajo de pruebas garantizan un entorno de ejecución “limpio” para cada prueba.
- Las pruebas de Lime son fáciles de leer y sus resultados también lo son. En los sistemas operativos que lo soportan, los resultados de Lime utilizan diferentes colores para mostrar de forma clara la información más importante.
- Symfony utiliza Lime para sus propias pruebas, por lo que el código fuente de Symfony incluye muchos ejemplos reales de pruebas unitarias y funcionales.
- El núcleo de Lime se valida mediante pruebas unitarias.
- Está escrito con PHP, es muy rápido y está bien diseñado internamente. Consta únicamente de un archivo, llamado lime.php, y no tiene ninguna dependencia.[7]

4.2.2 Pruebas de Aceptación.

Las pruebas de aceptación son las realizadas por el cliente y usuarios finales de la aplicación. En estas serán probadas las funcionalidades exigidas por el cliente y descritas en las historias de usuario, además de los aspectos de seguridad requeridos. Luego de haber superado las pruebas de aceptación podrá considerarse que la aplicación es apta para su uso y despliegue. 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: 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 llevarán a cabo siguiendo los pasos que se muestran a continuación:

- Se redactarán los casos de prueba teniendo en cuenta el orden de las historias de usuario y los niveles de prioridad dados a las funcionalidades.

- Se hará la planificación con el cliente de cuándo y cuáles pruebas serán llevadas a cabo.
- Se reunirán los miembros del proyecto seleccionados para realizar las pruebas.
- Se completarán cada uno de los campos de las tablas de las pruebas de aceptación con el resultado de la prueba.

A continuación se muestran los artefactos generados como fruto de las pruebas de aceptación:

Caso de Prueba de Aceptación	
Código: HU1_P1	Historia de Usuario: Autenticar Usuario
Nombre del responsable: Yunier Vega Pérez	
Descripción: Autenticarse en el sistema.	
Condiciones de ejecución: El módulo LDAP debe estar previamente activado. Los usuarios del sistema deben estar creados y configurados con sus respectivos permisos.	
Entrada/Pasos de ejecución: Se ingresan nombre de usuario y contraseña del dominio UCI.CU y se ejecuta la acción autenticar usuario.	
Resultado esperado: El usuario es autenticado correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 32. Caso de Prueba de Aceptación HU1_P1 “Autenticar Usuario”.

Caso de Prueba de Aceptación	
Código: HU2_P1	Historia de Usuario: Gestionar Usuario.
Nombre del responsable: Yunier Vega Pérez	
Descripción: Insertar un usuario en el sistema.	
Condiciones de ejecución: El usuario debe estar autenticado y tener los permisos correspondientes.	
Entrada/Pasos de ejecución: Se deben llenar todos los campos vacíos para agregar un usuario.	
Resultado esperado: El usuario es insertado correctamente en el sistema.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 33. Caso de Prueba de Aceptación HU2_P1 “Gestionar Usuario”.

Caso de Prueba de Aceptación	
Código: HU2_P2	Historia de Usuario: Gestionar Usuario.
Nombre del responsable: Yunier Vega Pérez	
Descripción: Eliminar usuario del sistema.	

Condiciones de ejecución: El usuario debe estar autenticado y tener los permisos correspondientes. El usuario a eliminar debe existir.
Entrada/Pasos de ejecución: Se busca el usuario y se procede a la eliminación.
Resultado esperado: El usuario es eliminado.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 34. Caso de Prueba de Aceptación HU2_P2 “Gestionar Usuario”.

Caso de Prueba de Aceptación	
Código: HU2_P3	Historia de Usuario: Gestionar Usuario.
Nombre del responsable: Yunier Vega Pérez	
Descripción: Buscar usuario en el sistema.	
Condiciones de ejecución: El usuario debe estar autenticado y tener los permisos correspondientes. El usuario a buscar debe existir.	
Entrada/Pasos de ejecución: Se busca el usuario y se muestran sus datos.	
Resultado esperado: El usuario es mostrado.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 35. Caso de Prueba de Aceptación HU2_P3 “Gestionar Usuario”.

Caso de Prueba de Aceptación	
Código: HU2_P4	Historia de Usuario: Gestionar Usuario.
Nombre del responsable: Yunier Vega Pérez	
Descripción: Modificar datos de usuario en el sistema.	
Condiciones de ejecución: El usuario debe estar autenticado y tener los permisos correspondientes. El usuario a modificar debe existir.	
Entrada/Pasos de ejecución: Se busca el usuario y se muestran sus datos para modificarlos.	
Resultado esperado: El usuario es modificado.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 36. Caso de Prueba de Aceptación HU2_P4 “Gestionar Usuario”.

Caso de Prueba de Aceptación	
Código: HU3_P1	Historia de Usuario: Gestionar Laboratorio.
Nombre del responsable: Yunier Vega Pérez	
Descripción: Insertar laboratorios en el sistema.	

Condiciones de ejecución: El usuario debe estar autenticado y tener los permisos correspondientes.
Entrada/Pasos de ejecución: Se deben llenar todos los campos vacíos para agregar un nuevo laboratorio.
Resultado esperado: El laboratorio es agregado al sistema.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 37. Caso de Prueba de Aceptación HU3_P1 “Gestionar Laboratorio”.

Caso de Prueba de Aceptación	
Código: HU3_P2	Historia de Usuario: Gestionar Laboratorio.
Nombre del responsable: Yunier Vega Pérez	
Descripción: Eliminar laboratorios en el sistema.	
Condiciones de ejecución: El usuario debe estar autenticado y tener los permisos correspondientes. EL laboratorio a eliminar debe existir.	
Entrada/Pasos de ejecución: Se busca el laboratorio y se procede a su eliminación.	
Resultado esperado: El laboratorio es eliminado del sistema.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 38. Caso de Prueba de Aceptación HU3_P2 “Gestionar Laboratorio”.

Caso de Prueba de Aceptación	
Código: HU3_P3	Historia de Usuario: Gestionar Laboratorio.
Nombre del responsable: Yunier Vega Pérez	
Descripción: Modificar laboratorios en el sistema.	
Condiciones de ejecución: El usuario debe estar autenticado y tener los permisos correspondientes. EL laboratorio a modificar debe existir.	
Entrada/Pasos de ejecución: Se busca el laboratorio y se muestran sus campos para modificarlos.	
Resultado esperado: El laboratorio es modificado.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 39. Caso de Prueba de Aceptación HU3_P3 “Gestionar Laboratorio”.

Caso de Prueba de Aceptación	
Código: HU3_P4	Historia de Usuario: Gestionar Laboratorio.
Nombre del responsable: Yunier Vega Pérez	
Descripción: Mostrar datos del laboratorio.	
Condiciones de ejecución: El usuario debe estar autenticado y tener los permisos correspondientes. EL	

laboratorio del cual se mostraran sus datos debe existir.
Entrada/Pasos de ejecución: Se busca el laboratorio y se muestran sus campos para ser consultados.
Resultado esperado: El laboratorio es mostrado con todos sus campos.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 40. Caso de Prueba de Aceptación HU3_P4 “Gestionar Laboratorio”.

Caso de Prueba de Aceptación	
Código: HU4_P1	Historia de Usuario: Gestionar Nomenclador.
Nombre del responsable: José Rafael Aguilar Quintana.	
Descripción: Insertar nomenclador.	
Condiciones de ejecución: El usuario debe estar autenticado y tener los permisos correspondientes.	
Entrada/Pasos de ejecución: Se deben llenar todos los campos vacíos para agregar un nuevo nomenclador.	
Resultado esperado: El nomenclador es agregado al sistema.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 41. Caso de Prueba de Aceptación HU4_P1 “Gestionar Nomenclador”.

Caso de Prueba de Aceptación	
Código: HU4_P2	Historia de Usuario: Gestionar Nomenclador.
Nombre del responsable: José Rafael Aguilar Quintana.	
Descripción: Eliminar nomenclador.	
Condiciones de ejecución: El usuario debe estar autenticado y tener los permisos correspondientes. El nomenclador a eliminar debe existir.	
Entrada/Pasos de ejecución: Se busca el nomenclador y se procede a su eliminación.	
Resultado esperado: El nomenclador es eliminado del sistema.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 42. Caso de Prueba de Aceptación HU4_P2 “Gestionar Nomenclador”.

Caso de Prueba de Aceptación	
Código: HU4_P3	Historia de Usuario: Gestionar Nomenclador.
Nombre del responsable: José Rafael Aguilar Quintana.	
Descripción: Modificar datos del nomenclador.	
Condiciones de ejecución: El usuario debe estar autenticado y tener los permisos correspondientes. El nomenclador a modificar debe existir.	

Entrada/Pasos de ejecución: Se busca el nomenclador y se muestran sus campos para ser modificados.
Resultado esperado: El nomenclador es modificado en el sistema.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 43. Caso de Prueba de Aceptación HU4_P3 “Gestionar Nomenclador”.

Caso de Prueba de Aceptación	
Código: HU5_P1	Historia de Usuario: Gestionar Medios.
Nombre del responsable: José Rafael Aguilar Quintana.	
Descripción: Insertar medios.	
Condiciones de ejecución: El usuario debe estar autenticado y tener los permisos correspondientes.	
Entrada/Pasos de ejecución: Se llenen todos los campos vacíos para agregar un nuevo medio al sistema.	
Resultado esperado: El medio es agregado.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 44. Caso de Prueba de Aceptación HU5_P1 “Gestionar Medios”.

Caso de Prueba de Aceptación	
Código: HU5_P2	Historia de Usuario: Gestionar Medios.
Nombre del responsable: José Rafael Aguilar Quintana.	
Descripción: Eliminar medios.	
Condiciones de ejecución: El usuario debe estar autenticado y tener los permisos correspondientes. El medio a eliminar debe existir.	
Entrada/Pasos de ejecución: Se busca el medio para proceder a su eliminación.	
Resultado esperado: El medio es eliminado.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 45. Caso de Prueba de Aceptación HU5_P2 “Gestionar Medios”.

Caso de Prueba de Aceptación	
Código: HU5_P3	Historia de Usuario: Gestionar Medios.
Nombre del responsable: José Rafael Aguilar Quintana.	
Descripción: Modificar datos de medios.	
Condiciones de ejecución: El usuario debe estar autenticado y tener los permisos correspondientes. El medio a modificar debe existir.	
Entrada/Pasos de ejecución: Se busca el medio para mostrar sus campos y modificarlos.	
Resultado esperado: El medio es modificado.	

Evaluación de la prueba: Prueba satisfactoria.

Tabla 46. Caso de Prueba de Aceptación HU5_P3 “Gestionar Medios”.

Caso de Prueba de Aceptación	
Código: HU5_P4	Historia de Usuario: Gestionar Medios.
Nombre del responsable: José Rafael Aguilar Quintana.	
Descripción: Mostrar datos de medios.	
Condiciones de ejecución: El usuario debe estar autenticado y tener los permisos correspondientes. El medio del cual se mostraran sus datos debe existir.	
Entrada/Pasos de ejecución: Se busca el medio para mostrar sus campos.	
Resultado esperado: El medio es mostrado.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 47. Caso de Prueba de Aceptación HU5_P4 “Gestionar Medios”.

Caso de Prueba de Aceptación	
Código: HU6_P1	Historia de Usuario: Gestionar Proyecto.
Nombre del responsable: José Rafael Aguilar Quintana.	
Descripción: Insertar Proyectos.	
Condiciones de ejecución: El usuario debe estar autenticado y tener los permisos correspondientes.	
Entrada/Pasos de ejecución: Se llenan los campos vacíos para agregar un nuevo proyecto al sistema.	
Resultado esperado: El Proyecto es agregado.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 48. Caso de Prueba de Aceptación HU6_P1 “Gestionar Proyecto”.

Caso de Prueba de Aceptación	
Código: HU6_P2	Historia de Usuario: Gestionar Proyecto.
Nombre del responsable: José Rafael Aguilar Quintana.	
Descripción: Eliminar Proyectos.	
Condiciones de ejecución: El usuario debe estar autenticado y tener los permisos correspondientes. El proyecto a eliminar debe existir.	
Entrada/Pasos de ejecución: Se busca el proyecto y se procede a su eliminación.	
Resultado esperado: El proyecto es eliminado del sistema.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 49. Caso de Prueba de Aceptación HU6_P2 “Gestionar Proyecto”.

Caso de Prueba de Aceptación	
Código: HU6_P3	Historia de Usuario: Gestionar Proyecto.
Nombre del responsable: José Rafael Aguilar Quintana.	
Descripción: Modificar datos del Proyecto.	
Condiciones de ejecución: El usuario debe estar autenticado y tener los permisos correspondientes. El proyecto del cual se modificaran sus datos debe existir.	
Entrada/Pasos de ejecución: Se busca el proyecto y se muestran sus campos para ser modificados.	
Resultado esperado: El proyecto es modificado.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 50. Caso de Prueba de Aceptación HU6_P3 “Gestionar Proyecto”.

Caso de Prueba de Aceptación	
Código: HU6_P4	Historia de Usuario: Gestionar Proyecto.
Nombre del responsable: José Rafael Aguilar Quintana.	
Descripción: Mostrar proyectos.	
Condiciones de ejecución: El usuario debe estar autenticado y tener los permisos correspondientes. El proyecto a mostrar debe existir.	
Entrada/Pasos de ejecución: Se busca el proyecto y se muestran sus campos para ser consultados.	
Resultado esperado: El proyecto es mostrado.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 51. Caso de Prueba de Aceptación HU6_P4 “Gestionar Proyecto”.

Caso de Prueba de Aceptación	
Código: HU7_P1	Historia de Usuario: Distribuir Proyecto.
Nombre del responsable: Yunier Vega Pérez	
Descripción: Asignar proyectos a un laboratorio.	
Condiciones de ejecución: El usuario debe estar autenticado y tener los permisos correspondientes. El proyecto para ser asignado y el laboratorio deben existir.	
Entrada/Pasos de ejecución: Se busca el proyecto y se asigna a un laboratorio.	
Resultado esperado: El proyecto es asignado.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 52. Caso de Prueba de Aceptación HU7_P1 “Distribuir Proyecto”.

Caso de Prueba de Aceptación	
Código: HU8_P1	Historia de Usuario: Realizar movimientos de medios.
Nombre del responsable: José Rafael Aguilar Quintana.	
Descripción: Selección de los medios a mover.	
Condiciones de ejecución: El usuario debe estar autenticado y tener los permisos correspondientes. El laboratorio y los medios seleccionados deben existir.	
Entrada/Pasos de ejecución: Se buscan los medios y se seleccionan.	
Resultado esperado: Los medios son seleccionados correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 53. Caso de Prueba de Aceptación HU8_P1 “Realizar movimientos de medios”.

Caso de Prueba de Aceptación	
Código: HU8_P2	Historia de Usuario: Realizar movimientos de medios.
Nombre del responsable: José Rafael Aguilar Quintana.	
Descripción: Eliminar medios seleccionados.	
Condiciones de ejecución: El usuario debe estar autenticado y tener los permisos correspondientes. Los medios a eliminar deben existir.	
Entrada/Pasos de ejecución: Se buscan los medios y se procede a su eliminación.	
Resultado esperado: Los medios son eliminados.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 54. Caso de Prueba de Aceptación HU8_P2 “Realizar movimientos de medios”.

Caso de Prueba de Aceptación	
Código: HU8_P3	Historia de Usuario: Realizar movimientos de medios.
Nombre del responsable: José Rafael Aguilar Quintana.	
Descripción: Insertar medios seleccionados en un nuevo local.	
Condiciones de ejecución: El usuario debe estar autenticado y tener los permisos correspondientes. Los medios a insertar deben existir.	
Entrada/Pasos de ejecución: Se buscan los medios y se agregan al nuevo local	
Resultado esperado: Los medios son insertados.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 55. Caso de Prueba de Aceptación HU8_P3 “Realizar movimientos de medios”.

Caso de Prueba de Aceptación	
Código: HU9_P1	Historia de Usuario: Registrar parte diario.
Nombre del responsable: Yunier Vega Pérez.	
Descripción: Mostrar parte diario.	
Condiciones de ejecución: El usuario debe estar autenticado y tener los permisos correspondientes. Los medios a mostrar en el parte deben existir.	
Entrada/Pasos de ejecución: Se buscan los medios, se verifica su estado y existencia física.	
Resultado esperado: Se muestra un parte actualizado.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 56. Caso de Prueba de Aceptación HU9_P1 “Registrar parte diario”.

Caso de Prueba de Aceptación	
Código: HU10_P1	Historia de Usuario: Realizar búsquedas en el sistema.
Nombre del responsable: José Rafael Aguilar Quintana.	
Descripción: Buscar contenido.	
Condiciones de ejecución: El usuario debe poseer los permisos correspondientes para realizar la búsqueda.	
Entrada/Pasos de ejecución: El usuario procede a ingresar la palabra clave relacionada con el contenido y se ejecuta la acción buscar.	
Resultado esperado: La búsqueda es realizada correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 57. Caso de Prueba de Aceptación HU10_P1 “Realizar búsquedas en el sistema”.

Caso de Prueba de Aceptación	
Código: HU10_P2	Historia de Usuario: Realizar búsquedas en el sistema.
Nombre del responsable: José Rafael Aguilar Quintana.	
Descripción: Mostrar contenidos encontrados.	
Condiciones de ejecución: El usuario debe ingresar la palabra clave previamente.	
Entrada/Pasos de ejecución: El usuario visualiza el resultado de la búsqueda realizada.	
Resultado esperado: Los contenidos encontrados se muestran correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 58. Caso de Prueba de Aceptación HU10_P2 “Realizar búsquedas en el sistema”.

Caso de Prueba de Aceptación	
Código: HU11_P1	Historia de Usuario: Obtener reportes.

Nombre del responsable: Yunier Vega Pérez.
Descripción: Mostrar reportes.
Condiciones de ejecución: El usuario debe estar autenticado y tener los permisos correspondientes a los reportes que solicita. La información mostrada en el reporte debe existir en el sistema.
Entrada/Pasos de ejecución: El usuario visualiza el resultado del reporte solicitado.
Resultado esperado: El reporte es mostrado correctamente.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 59. Caso de Prueba de Aceptación HU11_P1 “Obtener Reporte”.

4.2.3 Validación del sistema.

Con motivo de probar el correcto funcionamiento de la aplicación “Sistema para la gestión de inventario en los laboratorios de facultad 1 ” se realizó una prueba piloto entre el 19 y 21 de mayo del 2010 . Esta prueba se llevó a cabo en coordinación con la dirección de laboratorios correspondiente en el área, la misma arrojó resultados satisfactorios los cuales fueron avalados por el consejo de dirección.

La siguiente tabla muestra los resultados obtenidos:

NL: Número de Laboratorio.

CPc: Cantidad de computadoras.

PC: Computadoras.

Fecha	Funcionalidad	Dato Introducido	Resultado Esperado	Resultado Obtenido	Evaluación
19/05/2010	Autenticar Usuario	Usuario: lorenzom	Lorenzom	lorenzom	bien
		Usuario: yuniervp	Yuniervp	yuniervp	bien
		Usuario:jraguilar	Jraguilar	jraguilar	bien
	Gestionar Medio				
	- Insertar Pc	Serie:815250	Serie: 815250	Serie:815250	bien
	- Modificar Pc	Serie:815264	Serie: 815266	Serie:815266	bien
	- Eliminar Pc	Serie: 815250	Pc eliminada	Pc eliminada	bien
	Gestionar Laboratorio				
	- Insertar Lab	NL: 201 A:doc	NL:201 A:doc	NL:201 A:doc	bien
	- Modifica Lab	NL:202 A:doc	NL:202 A:doc	NL:202 A:doc	bien

	- Eliminar Lab	NL:201 A:doc	Lab eliminado	Lab eliminado	bien
20/05/2010	Gestionar Proyecto - Insertar proy - Modificar proy - Eliminar proy	Nombre: Akad Nombre: SICI Nombre: Akad	Proy: Akad Proy: Ident Proy eliminado	Proy: Akad Proy: Ident Proy eliminado	bien bien bien
	Distribuir Proyecto	NL:101 proy: Akad NL:101 proy: SICI	NL:101 proy: Akad NL:101 proy: SICI	NL:101 proy: Akad NL:101 proy: SICI	bien bien
21/05/2010	Realizar Movimientos	traslado NL: 202 CPc:3 - pc1:324455 - pc2:324506 - pc3:324475 NL:201 CPc:2 -pc1:324828 -pc2:324527 NL:203 CPc :2 pc1:324821 pc2:324826	NL:203 CPc:2 -pc1:324455 -pc1:324455 ----- NL:202 CPc:2 -pc1:324828 -pc2:324527 NL:201 CPc:2 -pc1:324821 -pc2:324826	NL:203 CPc:2 -pc1:324455 -pc1:324455 ----- NL:202 CPc:2 -pc1:324828 -pc2:324527 NL:201 CPc:2 -pc1:324821 -pc2:32482	bien bien error bien bien bien bien
21/05/2010	Obtener Reportes - Roturas PC - Roturas Otros de Medios	NL:201 -pc1: estado:rota -pc2: estado:rota sillas: 3 mesa: 2	NL:201 -pc1: stado:rota -pc2: etado:rota sillas: 3 mesa: 2	NL:201 -pc1: estado:rota -pc2: estado:rota sillas: 3 mesa: 2	bien bien bien bien

Tabla 60: Resultado de la Prueba Piloto.

4.3 Estudio de Factibilidad

4.3.1 Características del proyecto

Nombre de la entrada externa.	Cantidad de ficheros.	Cantidad de elementos de datos.	Clasificación (Simple, Media o Compleja).
Gestionar laboratorios	1	4	Simple

Capítulo 4: Pruebas y Estudio de Factibilidad

Gestionar nomenclador	1	3	Simple
Gestionar medios	1	4	Simple
Gestionar proyecto	1	4	Simple
Total		15	

Tabla 61: Entradas externas.

Nombre de la salida externa.	Cantidad de ficheros.	Cantidad de elementos de datos.	Clasificación (Simple, Media o Compleja).
Registrar parte diario.	1	1	Simple
Realizar búsquedas en el sistema.	1	3	Simple
Obtener reporte	1	3	Simple
Total		7	

Tabla 62: Salidas externas.

Nombre de la consulta externa.	Cantidad de ficheros.	Cantidad de elementos de datos.	Clasificación (Simple, Media o Compleja).
Distribuir proyecto	1	1	Simple
Realizar movimiento de medios	1	1	Simple
Gestionar laboratorios	1	4	Simple
Gestionar nomenclador	1	4	Simple
Gestionar medios	1	4	Simple
Gestionar proyecto	1	4	Simple
Total		18	

Tabla 63. Consultas Externas.

Nombre del fichero interno	Cantidad de ficheros.	Cantidad de elementos de datos.	Clasificación (Simple, Media o Compleja).
Laboratorio	1	1	Simple

Nomenclador	1	1	Simple
Medio	1	1	Simple
Proyecto	1	1	Simple
Reporte	1	1	Simple
Total	5		

Tabla 64. Archivos lógicos internos.

Elementos	Simple		Medio		Complejo		Subtotal
	No.	Peso	No.	Peso	No.	Peso	
Entradas externas	15	3	0	4	0	6	45
Salidas externas	7	4	0	5	0	7	28
Consultas Externas	18	3	0	4	0	6	51
Archivos lógicos internos	5	7	0	10	0	15	35
Total							159

Tabla 65. Puntos de función desajustados.

4.3.2 Cálculo de instrucciones fuentes, esfuerzo, tiempo de desarrollo, cantidad de hombres y costo.

El modelo COCOMO II usa Puntos Función y/o Líneas de Código Fuente (SLOC) como base para medir tamaño en los modelos de estimación de Diseño Temprano y Post-Arquitectura. Los Puntos de Función procuran cuantificar la funcionalidad de un sistema de software. La meta es obtener un número que caracterice completamente al sistema. Son útiles estimadores ya que están basados en información que está disponible en las etapas tempranas del ciclo de vida del desarrollo de software. COCOMO II considera solamente Puntos Función Desajustados.

Características	Valor
Puntos de función desajustados	159
Lenguaje (php)	53
Instrucciones fuentes por puntos de función	8427
Instrucciones fuentes	8.427 ksloc

Tabla 66. Características.

Nombre	Valor	Justificación
PREC	1.24	Existen varias aplicaciones similares a nivel mundial, pero no a nivel nacional.
FLEX	2.03	Cuenta con una alta flexibilidad.
TEAM	1.10	El equipo de desarrollo presenta muy alta cohesión y madurez.
RESL	2.83	El equipo de desarrollo presenta muy alta cohesión y madurez.
PMAT	3.12	Existe una experiencia previa en el desarrollo de aplicaciones web.
Total(FE)	10.32	

Tabla 67. Factores de escala.

Nombre	Valor	Justificación
DOCU	1.00	Documentación necesaria para el desarrollo del proyecto
RELY	1.00	Una falla en el software tendría un impacto medio.
RUSE	1.07	El nivel de reusabilidad del código es alto.
ACAP	0.85	La capacidad de análisis y diseño, eficiencia, habilidad para comunicarse y trabajar en equipo es alta.
AEXP	1.00	Nivel de experiencia del grupo de desarrollo en aplicaciones equivalentes es normal.
PEXP	0.91	Experiencia del grupo de desarrollo (principalmente programadores) en el uso de herramientas de software y hardware utilizado como plataforma.
LTEX	0.91	Experiencia del grupo de desarrollo en el lenguaje de programación y las herramientas de desarrollo utilizadas.
PCAP	0.88	La capacidad de trabajo en equipo, eficiencia y habilidad para comunicarse es alta.
Total(ME)	0.66	

Tabla 68. Multiplicadores de esfuerzo.

PREC: Precedencia.

FLEX: Flexibilidad en el desarrollo.

TEAM: Cohesión del equipo.

RESL: Arquitectura/Resolución de riesgo.

PMAT: Madurez del proceso.

DOCU: Documentación acorde a las diferentes etapas del ciclo de vida.

RELY: Confiabilidad requerida.

RUSE: Requerimientos de reusabilidad.

ACAP: Capacidad del analista.

AEXP: Experiencia en la aplicación.

PEXP: Experiencia en la plataforma.

LTEX: Experiencia en el lenguaje y las herramientas.

PCAP: Capacidad del programador.

Cálculos:

A (constante de calibración) = 2.94 Σ (FE) = 10.32 Π (ME) = 0.66 Sal= \$100

B (factor de escala) = 0.91 + 0.01 * Σ FE = 0.91 + 0.01 * 10.32 = 1.0132

C = 3.67 D = 0.24

Size = 53 x 159 = 8427 SLOC = 8.427 KSLOC

PM nominal = A x Size ^B = 2.94 x 8.427^1.0132 = 25.84

PM ajustado = PM nominal x Π (ME) = 25.84 x 0.66 = 17.05

F = D + 0.2 x 0.01 x Σ (FE) = 0.24 + 0.2 x 0.01 x 10.32 = 0.26

Tiempo calendario del desarrollo (TDEV):

TDEV = C x PM ajustado^ F = 3.67 x 17.05^0.26 = 7.20

Costo humano (CH):

CH = PM ajustado/ TDEV = 17.05 / 7.20 = 2.36 \approx 2

Cp. = CH x Sal x PM ajustado = 2 x 100 x 17.05 = \$ 3410 Cálculo de:	Valor
Esfuerzo	17.05 hombre/mes
Tiempo de desarrollo	7.20 meses
Cantidad de hombres	2 hombres
Salario medio	\$ 100
Costo	\$ 3410

Tabla 69. Resultados.

4.4 Beneficios tangibles e intangibles

El desarrollo del sistema propuesto en este trabajo brinda beneficio principalmente a la dirección de laboratorios en el área correspondiente a la Facultad¹ de la Universidad de Ciencias Informáticas, pues le permite al jefe de área y personal autorizado tener un mejor control de los medios disponibles en los laboratorios así como obtener reportes con la información necesaria, además cada uno de los líderes de proyecto también obtienen reportes de su proyecto ,permitiendo economizar gran parte del tiempo que se dedicaba al trabajo realizado manualmente, además, permite a la dirección de laboratorios llevar un control sistemático de todos los medios y bienes de estos. El Sistema de Gestión de Inventario para la Facultad 1 no está concebido para ser un producto comercial, aunque en un futuro pueda reportar beneficios monetarios a la entidad adquiriendo este carácter.

4.5 Análisis de costo.

Cada producto que se desarrolle tiene un costo de producción asociado, el cual debe ser justificado en base a los beneficios reportados por el mismo. El sistema propuesto en este trabajo de diploma no lleva consigo grandes gastos, debido a que solo contribuye a este el salario de los desarrolladores, debido a esto se llega a la conclusión de que su implementación es factible, a esto está muy vinculado la utilización de plataformas y herramientas libres que no requieren la compra de licencia ni ningún otro tipo de pago asociado a ello.

4.6 Conclusiones del Capítulo.

Se ha mostrado en este trabajo que una historia de usuario es solo un resumen de una funcionalidad que debe tener el sistema. Aunque los desarrolladores van ampliando el conocimiento sobre ese requisito funcional a medida que se implemente, en última instancia es el cliente el que tiene el conocimiento sobre qué es lo que quiere y es el que va a comprobarlo mediante una prueba de aceptación para darle el visto bueno. Por ello, es mucho más eficaz que sea el cliente, el cual tiene la información completa quien elabore las pruebas, además se realizó un análisis de factibilidad de la solución propuesta, llegando a la conclusión de que es viable su desarrollo comparando los costos de producción con los beneficios reportados por su puesta en funcionamiento.

CONCLUSIONES GENERALES

Se dio cumplimiento al objetivo general planteado al principio de la investigación, en el cual se propuso desarrollar un sistema informático para la gestión de inventario en los laboratorios de la Facultad 1, por lo que se concluye que:

- El sistema mejora el control de los medios y bienes.
- Se abarataron los costos para desarrollar el sistema usando herramientas y tecnologías libres.
- Se desarrolló el sistema con siete módulos, los cuales garantizan la automatización de los procesos de gestión de inventario que se realizaban de forma manual.
- La realización de pruebas al sistema garantizó la factibilidad del mismo, pues garantizó su funcionalidad.

RECOMENDACIONES

Una vez concluido el desarrollo de este trabajo se recomienda:

- Se recomienda a la universidad de las ciencias informáticas que este trabajo de diploma sea considerado como material de consulta por las personas que deban realizar un sistema similar y de esta forma el mismo pueda contribuir al desarrollo de futuros proyectos.
- Se recomienda a la dirección de laboratorios en área mantener una actualización periódica sobre el sistema para que se mantenga la fiabilidad y funcionamiento óptimo del mismo así como la integridad de la información que se gestiona a través de él.
- Se recomienda a la dirección de laboratorios en el área continuar trabajando en el perfeccionamiento del sistema con el propósito de adicionar otras funcionalidades para abarcar la totalidad de las problemáticas existentes.
- Se recomienda a los futuros desarrolladores que trabajen en el mejoramiento del sistema la elaboración de un manual de usuarios para facilitar el trabajo con la aplicación.
- Se recomienda a los futuros desarrolladores la realización de una ayuda online para brindar apoyo al personal que trabaje en la aplicación.

GLOSARIO DE TÉRMINOS

Apache: Servidor de páginas web de código abierto para diferentes plataformas (UNIX, Windows, etc.)

Aplicación: Programa que resuelve un problema determinado para un cliente final.

Cliente: Persona, organización o grupo de personas que encarga la construcción de un sistema, ya sea empezando desde cero o mediante el refinamiento de versiones sucesivas.

Código: Texto escrito en un lenguaje de programación que ha de ser compilado o interpretado para ejecutarse en una computadora.

Cocomo II: Es un modelo de estimación de proyectos de software que permite realizar estimaciones en función del tamaño del software, y de un conjunto de factores de costo y de escala. Los factores de costo describen aspectos relacionados con la naturaleza del producto, hardware utilizado, personal involucrado, y características propias del proyecto. Está compuesto por tres modelos denominados: Composición de Aplicación, Diseño Temprano y Post-Arquitectura.

Excel: Es un programa informático desarrollado y distribuido por Microsoft. Se trata de un software que permite realizar tareas contables y financieras gracias a sus aplicaciones para crear y trabajar con hojas de cálculo.

Hipertexto: Datos que contienen enlaces (links) a otros datos.

HTML: Lenguaje de marcas hipertextuales, diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web.

Herramientas CASE: Herramientas utilizadas para el desarrollo de proyectos de Ingeniería de Software.

Hardware: Se refiere a toda la infraestructura tecnológica, componentes físicos, computadoras, servidores y componentes periféricos (Impresoras, unidades de almacenamiento externo, scanner, entre otros).

Interfaz de Usuario: Es la parte de una aplicación que se encarga de interactuar con el usuario.

LDAP: (Protocolo Ligerero de Acceso a Directorios) es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.

Linux: Es una implementación independiente con "espíritu" POSIX (especificación para sistemas operativos). Tiene extensiones System V y BSD, y ha sido escrito completamente a base de aportaciones. Linux no tiene código propietario. Linux está distribuido libremente bajo "GNU Public License". Actualmente solo trabaja en IBM PC (o compatibles) y con arquitecturas ISA e EISA, y requiere un procesador 386 o superior. El kernel de Linux está escrito por Linux Torvalds (Torvalds@kruuna.helsinki.fi), desde Finlandia y otros voluntarios de otras partes del mundo.

Marco de trabajo: Es una estructura de soporte definida, en la cual otro proyecto de software puede ser organizado y desarrollado. Es un patrón de la arquitectura que proporciona una plantilla extensible para aplicaciones dentro de un dominio específico.

Mozilla Firefox: Navegador de Internet libre y de código abierto.

Microsoft SQL Server 2000: Es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional.

Open Source: Código abierto o código libre. Software que distribuye de forma libre su código fuente, de forma que los desarrolladores pueden hacer variaciones, mejoras o reutilizarlo en otras aplicaciones. También conocido como free software.

ORACLE: Oracle es básicamente una herramienta cliente/servidor para la gestión de bases de datos. Es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hace que sólo se vea en empresas muy grandes y multinacionales, por norma general.

Proyecto: Acciones y recursos que permiten en un tiempo dado el logro de un resultado para el cual fue concebido, un Proyecto definido puede durar más de un año o período fiscal.

PHP: Acrónimo de Hypertext Preprocessor es un lenguaje interpretado de alto nivel, embebido en páginas HTML

PostgreSQL: Es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) libre.

Recursos: Son los elementos del ordenador que utilizan los dispositivos para poder funcionar correctamente.

Sistema Informático: Un sistema informático como todo sistema, es el conjunto de partes interrelacionadas, hardware, software y de Recurso Humano.

Software: Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora.

Servidor: Es una aplicación informática o programa que realiza algunas tareas en beneficio de otras aplicaciones llamadas clientes. Algunos servicios habituales son los servicios de archivos, que permiten a los usuarios almacenar y acceder a los archivos de una computadora y los servicios de aplicaciones, que realizan tareas en beneficio directo del usuario final.

SQL: Es un lenguaje declarativo de acceso a bases de datos que permite especificar diversos tipos de operaciones sobre las mismas. Aúna características del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar información de interés de una base de datos.

SGBD: Sistema de Gestión de bases de datos. Es el software que permite la utilización y/o la actualización de los datos almacenados en una o varias bases de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez.

UML: “Unified Modeling Language” Lenguaje gráfico que brinda un vocabulario y reglas para especificar, construir, visualizar y documentar los artefactos de un sistema utilizando el enfoque orientado a objetos.

Word: Es un procesador de texto que permite la creación de documentos. Es parte del paquete de software Office de la empresa Microsoft.

REFERENCIAS BIBLIOGRÁFICAS

1. *Gestión y Manejo de inventario*. 2010; Disponible en: <http://mexico.smetoolkit.org/mexico/es/content/es/587/Gesti-oacute-n-de-inventario>.
2. Autores, C.D. *La definición de software libre*. 2010; Disponible en: <http://www.gnu.org/philosophy/free-sw.es.html>.
3. Rodríguez, C.M. *Las esferas socioculturales del Software Libre*. 2004; Disponible en: <http://www.derecho-internet.org/node/293>.
4. Canós José H., L.P., Penadés Carmen, *Metodologías Ágiles en el Desarrollo de Software*.
5. Gerardo, F.E., *Introducción a Extreme Programming*. 2002: Departamento de Sistemas Informáticos.
6. BECK, K., *Extreme Programming Explained*. 1999: First Edition September 29, 1999.
7. François Zaninotto, F.P., *Symfony 1.2, la guía definitiva*: Apress.
8. Autores, C.D. *Manual de PHP*. 2008; Disponible en: <http://es.php.net/my.php>.
9. FÉLIX, A.D.C.S. *El servidor de web Apache*. 2000; Disponible en: <http://acs.barrapunto.org/articulos/trunk/LinuxActual/Apache/apache.pdf>
10. Baruch, R. *NetBeans 6.8 Liberado*. 2009; Disponible en: <http://blogultura.com/java/netbeans-6-8-liberado/>.
11. *MySQL vs PostgreSQL*. 2009; Disponible en: <http://www.guatemwireless.org/articulos/mysql-vs-postgresql/>.
12. LETELIER, P., Penadés Carmen, *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. 2004.
13. *Tarjetas de CRC*. 2003; Disponible en: <http://jms32.eresmas.net/tacticos/UML/UML04/UML0402.html>.

BIBLIOGRAFÍA CONSULTADA

1. ÁLVAREZ, M.A. Qué es Java Script y las posibilidades que nos ofrece con respecto al HTML. 2009; Disponible en:
<http://www.desarrolloweb.com/articulos/490.php>.
2. AUTORES, C.D. Servicio Web. 2009; Disponible en:
http://es.wikipedia.org/wiki/Servicio_Web.
3. Rodríguez, C.M. Las esferas socioculturales del Software Libre. 2004; Disponible en:
<http://www.derecho-internet.org/node/293>.
4. AUTORES, C.D. Conceptos Básicos. ¿Qué es HTML? . Disponible en:
http://gias720.dis.ulpgc.es/Gias/Cursos/Tutorial_html/concepto.htm#html.
5. AUTORES, C.D. Script del lado del servidor. 2009; Disponible en:
http://es.wikipedia.org/wiki/Script_del_lado_del_servidor.
6. AUTORES, C.D., Manual de PHP. 2008.
7. FÉLIX, A.D.C.S. El servidor de Web Apache. 2000; Disponible en:
<http://acs.barrapunto.org/articulos/trunk/LinuxActual/Apache/apache.pdf>
8. GARCÍA, F., La Tesis y el Trabajo de Tesis: Recomendaciones metodológicas para la elaboración de los trabajos de tesis.
9. LATINA, O. ¿Por qué es importante UML? Disponible en:
<http://www.osmosislatina.com/lenguajes/uml/basico.htm>.
10. Masía, D. 2008; Disponible en:
<http://www.desarrolloweb.com/articulos/840.php>.
11. PAVÓN, E.L., Visual Paradigm, una herramienta de lo más útil.
12. Letelier, P., Metodologías Ágiles y XP: Departamento de Sistemas Informáticos y Computación.