



**Universidad de las Ciencias Informáticas
Facultad #1**

**Sistema de gestión de los Datos Primarios para la contabilización del costo de la
producción en la empresa Cuba Ron**

Trabajo de diploma para optar por el título de ingeniero en ciencias informáticas.

Autores:

Liusmy Cárdenas Medina.
Abel Savigne Suárez.

Tutor:

MSc. Rafael Rodríguez Puente

Año 50 de la Revolución
Ciudad de la Habana, Cuba
Junio 2009

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores del trabajo titulado: Sistema de gestión de los Datos Primarios para la contabilización del costo de la producción en la empresa Cuba Ron 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 __15__ días del mes de _____junio_____ del año ____2009____.

Liusmy Cardenas Medina

Abel Savigne Suárez

Rafael Rodriguez Puente

A mi madre por ser la mejor del mundo, sin ella no sería nada
A mi padre por todo su cariño y comprensión en todos estos años
A mi hermano Reinier porque esto sea un ejemplo a seguir

Liusmy

A mi madre por ser lo que más quiero en el mundo.
A mi padre que es para mí un ejemplo a seguir en la vida.
A mi suegra que es como mi segunda madre y me ha dado un poco de su cariño y
parte de su corazón.

Abel

Quiero agradecer a todas esas personas que hicieron posible este momento, a los que me inculcaron e influyeron en mi formación.

A mi tía Rafaela por ser como mi madre

A mi padrino Renato por ser alguien muy especial en mi vida

A mis primas que son como hermanas Mercedes, Marisol y Clara

A mi amigo, novio y compañero de tesis Abel por quererme y soportarme

A mi amigo entrañable Daniel por estar aquí aún no estando

A mis amistades que no podría enumerar todas porque son muchas

A los familiares de mi novio por quererme.

A las personas de Cuba Ron y la Ronera Santa Cruz por facilitarme la realización de la tesis.

Gracias a mi Revolución y a la vida por haberme dado tanto.

Liusmy

Quiero agradecerles a todas las personas que de una forma u otra influyeron en mi formación y me dieron un poco de todo lo bueno que aprendieron de la vida.

A mis hermanos Yordalis, Yami, Nelly, Angelín, Pachi, por darme apoyo incondicional estos 5 años de carrera.

A mi tía Margarita por ser otra madre para mí.

A mis primas Yanara, Leyanis por ser hermanas y amigas en estos 24 años.

A mi tío Walfrido por ser mi maestro y consejero mayor.

A toda la familia de mi novia por brindarme todo su cariño y tratarme como un hijo más.

En especial a mi amiga, novia y compañera Liusmy por ser unas de las cosas que más quiero en la vida, por entenderme y apoyarme en los momentos difíciles.

Abel

En la empresa Cuba Ron una de las actividades fundamentales es la contabilidad, la cual ha sido llevada de forma continua. Actualmente la empresa cuenta con un sistema que permite la gestión de los datos primarios del proceso de contabilización del costo de la producción, la cual se ha visto rezagada debido a que a pesar de tener un sistema automatizado tiene que realizar operaciones manualmente. Además el sistema presenta inconvenientes hacia una posible migración a software libre. El presente trabajo de diploma, “Sistema de gestión de los datos primarios para la contabilización del costo de la producción en la empresa Cuba Ron”, surge por la necesidad de tener un sistema automatizado con tecnologías de alta calidad y rendimiento, que pueda sustituir el sistema actualmente utilizado. Igualmente se quiere automatizar las operaciones que se hacen de forma manual. Con el propósito de facilitar las operaciones contables en la empresa Cuba Ron se realizará la implementación de una aplicación web que sea capaz de mejorar el trabajo en esta entidad y que les permita una mayor organización apoyándose en herramientas y tecnologías destacadas. El desarrollo del trabajo abrirá una nueva etapa en esta entidad pues permitirá una aplicación con tecnologías novedosas que aporten un excelente control de los datos primarios de la producción, una alta seguridad en la entrada al sistema y garantizará los reportes de las producciones.

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	5
1.1 Introducción.....	5
1.2 Conceptos claves.....	5
1.2.1 Contabilidad	5
1.2.2 Sistema contable	6
1.2.3 Contabilidad de costos	6
1.2.4 Datos Primarios	7
1.3 Sistemas contables utilizados en el mundo	7
1.3.1 AgroWin - Sistema de gestión total para el Agro	7
1.3.2 ContaPyme.....	7
1.4 Sistemas contables utilizados en Cuba	8
1.4.1 SABIC	8
1.4.2 Versat- Sarasola	8
1.5 Fundamentación de la tecnología a utilizar	9
1.5.1 Lenguaje del lado del servidor.....	9
1.5.2 Lenguaje del lado del cliente	11
1.5.3 Sistemas Gestores de Base de Datos (SGBD).	12
1.5.4 Proceso de desarrollo del software.....	15
1.5.5 Lenguaje de modelado.....	24
1.5.6 Herramienta Case.....	25
1.5.7 Framework.....	29
1.6 Conclusiones.....	32
CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.....	33
2.1 Introducción.....	33
2.2 Descripción de los procesos vinculados al campo de acción.....	33
2.2.1 Flujo actual del proceso.	33

2.2.2 Objeto de automatización	34
2.3 Propuesta del sistema	34
2.3.1 Personas relacionadas con el sistema	34
2.3.2 Requerimientos funcionales del sistema	35
2.3.3 Requerimientos no funcionales del sistema	37
2.4 Historias de usuario	38
2.5 Estimación de esfuerzo	43
2.6 Plan de iteraciones	43
2.6.1 Iteración 1	44
2.6.2 Iteración 2	44
2.6.3 Iteración 3	44
2.7 Plan de duración de iteraciones	44
2.8 Plan de entregas	45
2.9 Conclusiones	45
CAPÍTULO 3: CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA	47
3.1 Introducción	47
3.2 Metáforas del sistema	47
3.3 Arquitectura del sistema	47
3.3.1 Patrones de diseño en Symfony	48
3.3.2 Estructura del sistema	48
3.4 Diagramas de clases del diseño	50
3.5 Diseño de la base de datos	50
3.6 Tarjetas C.R.C	51
3.7 Tareas de la ingeniería	52
3.7.1 Iteración 1	52
3.7.2 Iteración 2	56
3.7.3 Iteración 3	58
3.8 Pruebas unitarias	60

3.9 Pruebas de aceptación.....	60
3.10 Conclusiones.....	68
Conclusiones Generales.....	69
Recomendaciones.....	70
Referencias bibliográficas.....	71
Bibliografía.....	73
Glosario de Términos.....	74
Anexo 1 Diagramas de clases del diseño.....	76
Anexo 2 Modelo de Base de datos.....	80
Anexo 3 Diagrama de despliegue.....	82
Anexo 4 Agile Manifesto.....	83
Anexo 5 Críticas a la metodología XP.....	85

Índice de tablas

Tabla 2. 1 Personas relacionadas al sistema.....	35
Tabla 2. 2 HU Autenticar	38
Tabla 2. 3 HU Gestionar usuario del sistema.....	39
Tabla 2. 4 HU Gestionar productos en el sistema	39
Tabla 2. 5 HU Gestionar normas de consumo	39
Tabla 2. 6 HU Gestionar materias primas en el sistema	40
Tabla 2. 7 HU Obtener indicadores específicos.....	40
Tabla 2. 8 HU Obtener causas de tiempo perdido.....	40
Tabla 2. 9 HU Mostrar planes de producción	41
Tabla 2. 10 HU Mostrar planes de entrega.....	41
Tabla 2. 11 HU Mostrar Tiempo Perdido Mensual.....	41
Tabla 2. 12 HU Mostrar Indicadores Específicos Mensual	42
Tabla 2. 13 HU Mostrar Causas de ajustes.....	42
Tabla 2. 14 HU Mostrar Causas de Otras Operaciones.....	42
Tabla 2. 15 Estimación de esfuerzos por HU	43
Tabla 2. 16 Plan de duración de iteraciones	45
Tabla 2. 17 Plan de duración de entrega.....	45
Tabla 3. 1 Estructura del Symfony	49
Tabla 3. 2 Estructura del sistema a desarrollar.....	50
Tabla 3. 3 Tarjeta C.R.C Productos	51
Tabla 3. 4 Tarjeta C.R.C Norma de Consumo	51
Tabla 3. 5 Tarjeta C.R.C Materias Prima.....	52
Tabla 3. 6 Tarjeta C.R.C Centro de Costo.....	52
Tabla 3. 7 Historias de usuario desarrolladas en la primera iteración	53
Tabla 3. 8 Tarea para crear la interfaz del sistema	53
Tabla 3. 9 Tarea 2 Mostrar formulario de validación de cuentas.....	53
Tabla 3. 10 Tarea 3 Diseño del formulario para la gestión de un producto	54
Tabla 3. 11 Tarea 4 Validar campos del formulario para la gestión de un producto	54
Tabla 3. 12 Tarea 5 Diseñar el formulario para la gestión de una norma de consumo.....	54
Tabla 3. 13 Tarea 6 Validar campos del formulario para la gestión de una norma de consumo	55
Tabla 3. 14 Tarea 7 Diseñar el formulario para gestionar una materia prima	55
Tabla 3. 15 Tarea 8 Validar campos del formulario para la gestión de una materia prima.....	55
Tabla 3. 16 Tarea 9 Diseñar el formulario para la gestión de usuario	56
Tabla 3. 17 Historias de usuario desarrolladas en la segunda iteración	56
Tabla 3. 18 Tarea 1 Diseñar la plantilla para obtener indicadores específicos	57

Tabla 3. 19 Tarea 2 Diseñar la plantilla par obtener las causas de tiempo perdido	57
Tabla 3. 20 Tarea 3 Diseñar la plantilla para mostrar los planes de producción	57
Tabla 3. 21 Tarea 4 Diseñar la plantilla para ver los planes de entrega.....	58
Tabla 3. 22 Historias de usuario desarrolladas en la tercera iteración	58
Tabla 3. 23 Tarea 1 Diseñar plantilla para mostrar el tiempo perdido del mes	59
Tabla 3. 24 Tarea 2 Diseñar la plantilla para ver los indicadores específicos del mes	59
Tabla 3. 25 Tarea 3 Diseñar plantilla para mostrar las causas de ajustes	59
Tabla 3. 26 Tarea 4 Diseñar plantilla para mostrar las causas de otras operaciones.....	60
Tabla 3. 27 Prueba de aceptación Autenticarse en el sistema.....	61
Tabla 3. 28 Prueba de aceptación Insertar usuario en el sistema	61
Tabla 3. 29 Prueba de aceptación Eliminar usuario del sistema.....	62
Tabla 3. 30 Prueba de aceptación Insertar producto en el sistema	62
Tabla 3. 31 Prueba de aceptación Modificar producto en el sistema.....	63
Tabla 3. 32 Prueba de aceptación Eliminar producto del sistema	63
Tabla 3. 33 Prueba de aceptación Insertar norma de consumo en el sistema	63
Tabla 3. 34 Prueba de aceptación Modificar norma de consumo en el sistema.....	64
Tabla 3. 35 Prueba de aceptación Eliminar norma de consumo en el sistema	64
Tabla 3. 36 Prueba de aceptación Insertar materia prima en el sistema.....	64
Tabla 3. 37 Prueba de aceptación Modificar materia prima.....	65
Tabla 3. 38 Prueba de aceptación Eliminar materia prima	65
Tabla 3. 39 Prueba de aceptación Ingresar indicadores específicos.....	66
Tabla 3. 40 Prueba de aceptación Ingresar tiempo perdidos diarios	66
Tabla 3. 41 Prueba de aceptación Mostrar planes de producción	66
Tabla 3. 42 Prueba de aceptación Mostrar planes de entrega	67
Tabla 3. 43 Prueba de aceptación Mostrar Tiempo perdido mensual	67
Tabla 3. 44 Prueba de aceptación Mostrar Indicadores específicos.....	67
Tabla 3. 45 Prueba de aceptación Mostrar Causas de ajustes	68

Índice de figuras

Ilustración 1 Fases de la metodología XP(9).....	16
Ilustración 2 Fases de la metodología RUP(9).....	18
Ilustración 3 El proceso de Scrum(11)	22
Ilustración 6 Plantilla global en Symfony	48
Ilustración 7 Plantillas globales de la aplicación	48
Ilustración 8 Estructura de Symfony.....	49
Ilustración 9 Estructura del sistema a	50

INTRODUCCIÓN

El desarrollo acelerado del mundo actual hace imposible que cualquier entidad o institución pueda controlar de forma manual su actividad fundamental. Es por ello que la mayoría utilizan sistemas automatizados que permitan agilizar su trabajo de manera óptima, eficiente y con el personal necesario.

Los métodos utilizados para llevar a vías de hecho la contabilidad provienen de la antigüedad y su desarrollo viene ligado al del comercio. Durante la primera mitad del siglo XX las tareas de índole económico-financieras se realizaban en su totalidad manualmente o ayudados con máquinas de cálculo o de cálculo y registro y de gran simplicidad; posteriormente éstas máquinas fueron sustituidas por otras más costosas y de gran tamaño. En la actualidad el conjunto de conocimientos y técnicas que hacen posible el procesamiento automático de la información han alcanzado a las máquinas del sector comercial y científico, al punto en que las computadoras han sustituido a las calculadoras y a los equipos de contabilidad manual.

Al igual que un sistema informático puede verse como cualquier conjunto de dispositivos que colaboran en la realización de una tarea; un sistema de contabilidad puede verse como una combinación de programas, procedimientos, datos y equipamiento, utilizados de manera coherente en el procesamiento de la información.

En particular, la empresa Cuba Ron constituye una entidad en la cual la contabilidad es una de las actividades fundamentales. Además, ha tenido que valerse de un sistema (contable) para controlar y agilizar sus operaciones.

Cuba Ron es la organización que se ha erigido como productor de grandes marcas de ron en Cuba, depositaria de las más fieles tradiciones de la cultura ronera cubana, como parte enriquecedora e inseparable de la cultura nacional. Integrada al sistema de perfeccionamiento empresarial, participa en el proceso de recuperación de la economía cubana, desplegando sus reservas de eficiencia en una estrategia empresarial, con una visión definida y dirigida a diversificar en su objeto social por lo que requiere de un sistema que contabilice el costo de la producción.

En la empresa Cuba Ron existe un sistema automatizado con el objetivo de llevar la contabilización del costo de la producción en cada una de sus Unidades Empresariales de Base (UEB), sin embargo este sistema está diseñado en Microsoft Access 97, con Visual Basic 6.0, lo cual lo hace ser un sistema

antiguo e ineficiente, careciendo de agilidad y rapidez, por lo que el trabajo se torna engorroso ya que no cumple con sus objetivos. También surgirían inconvenientes para una futura migración a software libre. Existen operaciones que se hacen manualmente y que se necesita automatizarlas para lograr rapidez en la captura de información. No tiene roles definidos por lo que el acceso al sistema no es seguro ya que la información puede modificarse por personas ajenas, además la base de datos no es centralizada provocando que la información que se envía puede extraviarse o que la misma no sea la más actualizada. Teniendo esto como **situación problemática** concluimos como **problema científico** ¿Cómo lograr una mejora en la gestión de los datos primarios en el proceso de contabilización del costo de la producción en la empresa Cuba Ron? , tomando como **objeto de estudio** el proceso de contabilización del costo de la producción en la empresa Cuba Ron y como **campo de acción** datos primarios del proceso de contabilización del costo de la producción para la empresa Cuba Ron.

Para darle solución al problema definido se ha planteado como **objetivo general** desarrollar un sistema web que permita a la presidencia de la empresa controlar los datos primarios de las unidades empresariales básicas (UEB) y como **objetivos específicos**:

- Analizar la situación actual del proceso de contabilización de costo de la empresa Cuba Ron.
- Analizar las herramientas que se usarán en la elaboración de la aplicación informática.
- Realizar el análisis y diseño del sistema informático.
- Implementar el sistema informático, haciendo uso de técnicas y herramientas anteriormente escogidas.
- Proponer el sistema desarrollado.

Para cumplir con estos objetivos se proponen las siguientes **tareas de la investigación**:

- Analizar la situación actual acerca de la empresa Cuba Ron.
- Entrevistar a los contadores de la empresa.
- Investigar sobre el proceso de contabilización de la empresa.
- Seleccionar las herramientas necesarias para el análisis y diseño del sistema.
- Diseñar el sistema.
- Describir las funcionalidades de los módulos en los que se va a dividir el sistema
- Estudiar las técnicas y herramientas necesarias para implementar el sistema.
- Seleccionar la plataforma adecuada para la realización del sistema.

- Implementar el sistema.
- Realizar pruebas a la solución.

Para guiar el desarrollo de este proceso se plantea como **hipótesis:**

Si se desarrolla un sistema web que permita a la presidencia de la empresa controlar los datos primarios de las unidades empresariales básicas, entonces se logrará una mejora en la gestión de los datos primarios en la presidencia de la empresa.

Para la realización del trabajo se tienen en cuenta algunos **métodos investigativos**. A continuación se mencionarán cada uno de ellos y de qué forma se ponen de manifiesto en la investigación.

Analítico-sintético:

Este método fue utilizado para analizar la situación problemática y determinar posibles variantes de solución.

Entrevista:

Se realizaron entrevistas a varios trabajadores de la empresa Cuba Ron con el fin de conocer en qué estado se encuentra el tema de la contabilización de los costos de la producción y de qué manera realizan la misma.

Hipotético-deductivo:

Permitió, a partir de la hipótesis, elaborar conclusiones acerca de la factibilidad de mejorar la automatización del proceso de contabilización de costo en la Empresa Cuba Ron.

El documento está estructurado por 3 capítulos:

Capítulo 1: Fundamentación Teórica, expone las principales características de las herramientas, tecnologías y la metodología empleadas en el desarrollo del sistema.

Capítulo 2 Descripción de la solución propuesta, se describe una propuesta de la investigación y los procesos que están vinculados al campo de acción, se plantean los requisitos funcionales y no funcionales que debe tener el sistema para lograr óptimos resultados, además de hacer referencia a las personas vinculadas a éste. Se detallan los artefactos generados en la fase de Exploración y Planificación de la metodología XP teniendo especial atención en las Historias de Usuario.

Capítulo 3 Construcción de la solución propuesta, aborda las fases de Diseño, Implementación y Prueba, donde se describen los artefactos de tarjetas C.R.C, tareas de la ingeniería y caso de prueba de aceptación

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

Este capítulo tiene como objetivo fundamental definir una serie de conceptos para mayor entendimiento del problema; se muestra una breve descripción de varios software existente en la actualidad, tanto a nivel nacional como internacional, para llevar la contabilidad en las empresas. Además expone las principales características de las herramientas, tecnologías y la metodología empleadas en el desarrollo del sistema.

1.2 Conceptos claves

La Contabilidad se remonta desde tiempos muy antiguos, cuando el hombre se ve obligado a llevar registros y controles de sus propiedades porque su memoria no bastaba para guardar la información requerida.

Actualmente, dentro de lo que son los sistemas de información empresarial, la contabilidad se erige como uno de los sistemas más eficaces para dar a conocer los diversos ámbitos de la información de las unidades de producción o empresas. El concepto ha evolucionado, de forma que cada vez es mayor el grado de "especialización" de ésta disciplina dentro del entorno empresarial.

Es de gran importancia porque todas las empresas tienen la necesidad de llevar un control de sus negociaciones, así obtendrá mayor productividad.

1.2.1 Contabilidad

La contabilidad, es una herramienta empresarial que permite el registro y control sistemático de todas las operaciones que se realizan en la empresa, por ende no existe una definición concreta de la contabilidad aunque todas estas definiciones tienen algo en común. (1)

A continuación se presentan varias acepciones de la contabilidad que han sido definidas por diferentes autores y cuerpos colegiados de la profesión contable:

"La contabilidad es el arte de registrar, clasificar y resumir en forma significativa y en términos de dinero, las operaciones y los hechos que son cuando menos de carácter financiero, así como el de interpretar sus resultados" (Instituto Americano de Contadores Públicos Certificados)

"La contabilidad es el sistema que mide las actividades del negocio, procesa esa información convirtiéndola en informes y comunica estos hallazgos a los encargados de tomar las decisiones" (Horngren & Harrison. 1991)

"La contabilidad es el arte de interpretar, medir y describir la actividad económica" (Meigs, Robert., 1992)

"La contabilidad es el lenguaje que utilizan los empresarios para poder medir y presentar los resultados obtenidos en el ejercicio económico, la situación financiera de las empresas, los cambios en la posición financiera y/o en el flujo de efectivo" (Catacora, Fernando, 1998)

"La contabilidad tiene diversas funciones, pero su principal objetivo es suministrar, cuando sea requerida o en fechas determinadas, información razonada, en base a registros técnicos, de las operaciones realizadas por un ente público o privado" (Redondo, A., 2001)

En fin, la contabilidad es una técnica que se ocupa de registrar, clasificar y resumir las operaciones de un negocio con el objetivo de mostrar sus resultados, para poder orientarse sobre el curso que siguen los negocios mediante datos contables; permitiendo así conocer la estabilidad, la solvencia de la compañía y la capacidad financiera de la empresa.

1.2.2 Sistema contable

Sistema de contabilidad es una estructura organizada mediante la cual se recogen las informaciones de una empresa como resultado de sus operaciones, valiéndose de recursos como formularios, reportes, libros etc. y que presentados a la gerencia le permitirán a la misma tomar decisiones financieras.

Un sistema de contabilidad no es más que normas, pautas, procedimientos etc. para controlar las operaciones y suministrar información financiera de una empresa, por medio de la organización, clasificación y cuantificación de las informaciones administrativas y financieras que se nos suministre. (2)

1.2.3 Contabilidad de costos

La Contabilidad de Costos es un sistema de información que clasifica, asigna, acumula y controla los costos de actividades, procesos y productos, para facilitar la toma de decisiones, la planeación y el control administrativo.(3)

En otras palabras la contabilidad de costos es la encargada del cálculo de los costos tomando como base los elementos de la producción, y permite determinar si hubo ganancias o pérdidas.

1.2.4 Datos Primarios

Los datos primarios, son la información inicial con la que se cuenta para desarrollar un determinado proyecto. Es la información básica de trabajo, son los datos mínimos necesarios y requeridos para la realización de un análisis determinado.

1.3 Sistemas contables utilizados en el mundo

1.3.1 AgroWin - Sistema de gestión total para el Agro

Es un sistema especialmente diseñado para proveer información al agricultor sobre su finca y sus recursos.

Una de sus principales características es que utiliza un esquema de registro de información muy sencillo, de tal forma que se convierta en un instrumento fácil de utilizar y ágil en el seguimiento y toma de decisiones basadas en la información oportuna y actualizada de la finca.

El registro sencillo de la información de la finca facilita el que muchos agricultores con escasos conocimientos técnicos en áreas como contabilidad, inventarios, presupuestos y estadísticas, queden habilitados para manejar el sistema, debido a que es el sistema el que se encarga de interpretar la información base que el usuario le suministra y automáticamente transformarla en información técnica y correctamente clasificada y codificada a la luz de las técnicas financieras y estadísticas.(4)

Este sistema a pesar de ser un sistema de gestión, no es factible pues solo abarca el área agrícola y no está generalizado a otras empresas.

1.3.2 ContaPyme

Sistema totalmente integrado:(5)

- El más sencillo, amigable y completo sistema de contabilidad y gestión.
- Desarrollo de última tecnología.
- Versátil y adaptable a cualquier empresa.
- Escalable con la empresa.

- Permite ejercer un control total de la información.
- Manejo de múltiples áreas de trabajo y múltiples empresas.
- Único con opciones especiales para el cálculo de costos de producción.

A pesar de las grandes ventajas que presenta este sistema, es muy costoso por lo que el país no lo puede adquirir.

1.4 Sistemas contables utilizados en Cuba

1.4.1 SABIC

El SABIC (Sistema automatizado para la Banca Internacional de Comercio) es un sistema diseñado y desarrollado por la Dirección de Sistemas Automatizados del Banco Central de Cuba para satisfacer las necesidades de procesamiento de datos de Bancos e instituciones no bancarias, utilizando los medios técnicos de computación disponibles en el mercado.

Este sistema ha sido adaptado a los requerimientos de las operaciones propias del Banco Central y ha sido desarrollado para que los empleados que hagan uso de él puedan tramitar sus operaciones y realizar sus consultas sin necesidad de acudir a los archivos ni a la actividad manual. De esta forma se aumenta la seguridad, la eficiencia del trabajo y la productividad de los trabajadores.

Entre sus principales características, está la contabilización en tiempo real (que permite mantener actualizados los ficheros contables); y contabilización multimoneda (que permite registrar los activos y pasivos en las monedas orígenes sin tener que realizar en el momento del registro las conversiones de monedas, lo cual aumenta la exactitud de la información sobre la posición financiera de la institución). Además, las operaciones contables se pueden realizar a través de transacciones tipificadas que generan los asientos contables de forma automática. (6)

Este sistema no resuelve la problemática pues está solamente adaptado a las operaciones del banco.

1.4.2 Versat- Sarasola

Este es un software creado por el licenciado en Economía y villaclareño Miguel Cabrera González, este es un sistema de contabilidad confiable, ofrece mayor organización, control y disciplina en cada gestión, posibilita enviar información eficaz, de forma inmediata y desde lugares apartados.

Este proyecto resulta ser un sistema integrado, constituido por 12 módulos que incluyen configuración y seguridad, contabilidad general y de gastos, costos y procesos, análisis económico empresarial y control

de activos fijos, además de intervenir finanzas y cajas, planificación y presupuestos, control de inventarios, de productos terminados, pago de salario, paquete de gestión, contratación y facturación.

Este producto le ha traído grandes beneficios al país, pues se ahorró un millón 186 mil dólares que costaban las licencias al evitarse la importación del sistema foráneo, más labor de consultaría, atenciones y otros gastos, además de que actualmente lo utilizan alrededor de 200 entidades de varias provincias y siguen aumentando los clientes.

Este sistema a pesar de ser realizado en Cuba, no es aplicable en la empresa Cuba Ron porque es una aplicación de escritorio y tiene código cerrado.

1.5 Fundamentación de la tecnología a utilizar

1.5.1 Lenguaje del lado del servidor

La programación del lado del servidor es una tecnología que consiste en el procesamiento de una petición de un usuario mediante la interpretación de un script en el servidor web para generar páginas HTML dinámicamente como respuesta.

En el dominio de la red, los lenguajes de lado servidor más ampliamente utilizados para el desarrollo de páginas dinámicas son el ASP, PHP y PERL.

1.5.1.1 PHP

El lenguaje PHP es un lenguaje de alto nivel empleado para el desarrollo de aplicaciones web. Este lenguaje permite crear prácticamente cualquier funcionalidad imaginable. La principal característica de PHP es que es un lenguaje que se ejecuta en el servidor, y por tanto, es independiente del navegador que use el visitante. Una de las características más importantes de PHP es que permite manejar bases de datos desde una interfaz web. Dado que PHP es un lenguaje de código abierto, se puede usar libremente sin necesidad de pagar licencia.(7)

Ventajas

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.

- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Biblioteca nativa de funciones sumamente amplia e incluída.
- No requiere definición de tipos de variables.
- Tiene manejo de excepciones (desde PHP5).

1.5.1.2 Perl

Perl es un lenguaje de programación interpretado, al igual que muchos otros lenguajes de Internet como Javascript o ASP. Esto quiere decir que el código de los scripts en Perl no se compila sino que cada vez que se quiere ejecutar se lee el código y se pone en marcha interpretando lo que hay escrito. Además es extensible a partir de otros lenguajes, ya que desde Perl podremos hacer llamadas a subprogramas escritos en otros lenguajes. También desde otros lenguajes podremos ejecutar código Perl. (8)

1.5.1.3 ASP

Microsoft Active Server Page (ASP) es una tecnología de script que corre del lado de servidor y puede ser usado para crear aplicaciones Web dinámicas e interactivas. Una página ASP es una página HTML que contienen scripts que corren del lado del servidor que son procesados por un servidor Web antes de ser utilizado por el navegador. Usted puede combinar ASP con XML (Extensible Markup Language) para crear sitios Web interactivos poderosos. (8)

ASP es una característica de Microsoft Internet Information Server. Debido a que los scripts que corren en servidor son construidos en una página regular de HTML, este puede ser servido en casi cualquier navegador. Un archivo ASP puede ser creado incluyendo un script escrito en VBScript o JScript en un archivo HTML. (8)

1.5.1.4 JSP

JSP es un acrónimo de Java Server Pages, que en castellano vendría a decir algo como Páginas de Servidor Java. Es, pues, una tecnología orientada a crear páginas web con programación en Java.

Con JSP podemos crear aplicaciones web que se ejecuten en variados servidores web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma. Por tanto, las JSP podremos escribirlas con nuestro editor HTML/XML habitual.(8)

1.5.1.5 Selección del lenguaje del servidor

Se escogió el lenguaje de programación PHP por ser líder entre los lenguajes existentes, por lo que es el más utilizado en el mundo. Además para la elaboración del sistema se cuenta con el framework Symfony desarrollado con PHP.

1.5.2 Lenguaje del lado del cliente

El lenguaje del lado del cliente es aquel que toma las acciones en el propio navegador del usuario.

1.5.2.1 Javascript

Es un lenguaje de programación interpretado; es decir, que no requiere compilación, utilizado principalmente en páginas Web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C.

Al contrario de Java, JavaScript no es un lenguaje orientado a objetos propiamente dicho, al no disponer de Herencia; es más bien un lenguaje basado en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad.

Todos los navegadores interpretan el código JavaScript integrado dentro de las páginas Web. Para interactuar con una página Web se provee al lenguaje JavaScript de una implementación del DOM.

1.5.2.2 AJAX (Asynchronous JavaScript And XML)

AJAX, acrónimo de Asynchronous Javascript And XML (Javascript y XML asíncronos, donde XML es un acrónimo de eXtensible Markup Language), lo que quiere decir Ajax no es una sola tecnología, sino la unión de varias, uniéndose en una nueva y poderosas forma.

AJAX incorpora:

- Presentación basada en estándares usando XHTML y CSS

- Exhibición e interacción dinámicas usando el Document Object Model
- Intercambio y manipulación de datos usando XML and XSLT
- Recuperación de datos asincrónica usando XMLHttpRequest

Se ejecuta en el cliente, es decir, en el navegador del usuario, y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma. En resumen, Ajax es un enfoque impresionante del uso de distintas tecnologías, que están conduciendo a una nueva generación de aplicaciones web, desde una muy sencilla aplicación que se introduce en este enfoque hasta las complejas aplicaciones de servicios web.

1.5.2.3 HTML

El lenguaje llamado HTML indica al navegador donde colocar cada texto, cada imagen o cada video y la forma que tendrán estos al ser colocados en la página.

El lenguaje consta de etiquetas que tienen esta forma o <P>. Cada etiqueta significa una cosa, por ejemplo significa que se escriba en negrita (bold) o <P> significa un párrafo, <A> es un enlace, etc. Casi todas las etiquetas tienen su correspondiente etiqueta de cierre, que indica que a partir de ese punto no debe de afectar la etiqueta. Por ejemplo se utiliza para indicar que se deje de escribir en negrita. Así que el HTML no es más que una serie de etiquetas que se utilizan para definir la forma o estilo que queremos aplicar a nuestro documento. Esto está en negrita. (8)

1.5.2.4 Selección del lenguaje del lado del cliente

Se definió utilizar Ajax para un mejor rendimiento y velocidad en los tiempos de respuesta de la aplicación, así como el lenguaje JavaScript para las validaciones del lado del cliente.

1.5.3 Sistemas Gestores de Base de Datos (SGBD).

Los sistemas de gestión de base de datos son un tipo de 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 de un lenguaje de consulta. El propósito general de los sistemas de gestión de base 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: Microsoft SQL Server, Oracle, PostgreSQL y MySQL.

1.5.3.1 PostgreSQL

PostgreSQL está considerado como uno de los gestores de bases de datos de código abierto más avanzados del mundo. Proporciona un gran número de características que normalmente sólo se encontraban en las bases de datos comerciales tales como Oracle.

PostgreSQL aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, optimización de consultas y herencia. Tiene soporte para lenguajes procedurales internos (son aquellos en los cuales el usuario instruye al sistema para que lleve a cabo una serie de operaciones en la base de datos con el fin de calcular el resultado deseado y están fundamentados en la utilización de variables para almacenar valores y en la realización de operaciones con los datos almacenados), incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL. Otra ventaja de PostgreSQL es su habilidad para usar Perl o Python.

1.5.3.2 Oracle

Es un sistema de administración de bases de datos fabricado por Oracle Corporation. Se considera como uno de los sistemas de bases de datos más completos, destacando su soporte de transacciones, estabilidad y escalabilidad. Además, es multiplataforma. Su mayor defecto es su enorme precio, que es de varios miles de euros

1.5.3.3 MySQL

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario. MySQL Database Server es la base de datos de código fuente abierto más usada del mundo. Su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar. La extensiva reutilización del código dentro del software y una aproximación minimalística para producir características funcionalmente ricas, ha dado lugar a un sistema de administración de la base de datos incomparable en velocidad, compactación, estabilidad y facilidad de despliegue.

Características:

- Acceso a las bases de datos de forma simultánea por varios usuarios y/o aplicaciones.

- Seguridad, en forma de permisos y privilegios, determinados usuarios tendrán permiso para consulta o modificación de determinadas tablas. Esto permite compartir datos sin que peligre la integridad de la base de datos o protegiendo determinados contenidos.
- Potencia: MySQL es un lenguaje muy potente para consulta de bases de datos, usar un motor ahorra una enorme cantidad de trabajo.
- Portabilidad: MySQL es también un lenguaje estandarizado, de modo que las consultas hechas usando MySQL son fácilmente portables a otros sistemas y plataformas.
- Conectividad: es decir, permite conexiones entre diferentes máquinas con distintos sistemas operativos. Es corriente que servidores Linux o Unix, usando MySQL, sirvan datos para ordenadores con Windows, Linux, Solaris, etc. Para ello se usa TCP/IP, tuberías, o sockets Unix.
- Sin lugar a duda, lo mejor de MySQL es su velocidad a la hora de realizar las operaciones, lo que le hace uno de los gestores que ofrecen mayor rendimiento.
- Su bajo consumo lo hacen apto para ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Las utilidades de administración de este gestor son envidiables para muchos de los gestores comerciales existentes, debido a su gran facilidad de configuración e instalación.
- Tiene una probabilidad muy reducida de corromper los datos, incluso en los casos en los que los errores no se produzcan en el propio gestor, sino en el sistema en el que está.

1.5.3.4 Microsoft SQL Server

Microsoft SQL Server es una plataforma de base de datos y análisis de datos que se utiliza en el procesamiento de transacciones en línea (OLTP) a gran escala, el almacenamiento de datos y las aplicaciones de comercio electrónico. SQL Server ofrece la tecnología y las funciones con las que pueden contar las organizaciones. Con avances significativos en áreas clave de la administración de datos empresariales, la productividad de los desarrolladores y la inteligencia empresarial, las ventajas de SQL Server son considerables. Aprovechamiento de los activos de datos. Aumento de la productividad. Reducción de la complejidad de la tecnología de la información. Disminución del costo total de propiedad. Microsoft SQL Server, al contrario de su más cercana competencia, no es multiplataforma, ya que sólo está disponible en Sistemas Operativos de Microsoft.

1.5.3.5 Selección del sistema gestor de base de datos.

Como Sistema Gestor de Base de Datos (SGBD) de software libre se decidió utilizar PostgreSQL, ya que es el más recomendado para sistemas que tengan alta concurrencia y que maneje gran cantidad de información como es el caso de la aplicación en cuestión. Otra razón por la cual se seleccionó este SGBD es que MySQL desde enero del año 2008 fue adquirida por la empresa SunMicrosystems, lo que provoca una dualidad en cuanto a las licencias de uso, algo que puede traer futuras complicaciones

1.5.4 Proceso de desarrollo del software

Con el objetivo de crear y mantener las aplicaciones de software, aplicando las tecnologías y prácticas computacionales, surge la Ingeniería de software. El desarrollo y evolución constante experimentada por los procesos de Ingeniería de software, ha traído consigo la realización de varias tareas en este campo, como son: análisis de requisitos, especificación, diseño y arquitectura, programación, prueba, documentación y mantenimiento.

El proceso de desarrollo del 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 presentes, luego del perfeccionamiento de los procesos del software durante años, han llevado a cabo dos corrientes significativas: los llamados métodos ligeros y métodos pesados. Aunque ambos están enfocados 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.

Entre las metodologías de desarrollo de software encontramos:

- XP (eXtreme Programing)
- Scrum

- MSF (Microsoft Solution Features)
- RUP (Rational Unified Process)

Entre las características generales que presentan dichas metodologías se encuentran las siguientes:

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

1.5.4.1 XP (Programación Extrema)

Basándose en la simplicidad, la comunicación y la reutilización del código, surge la Programación Extrema (XP), siendo así una metodología muy ligera de desarrollo de software. XP consiste en una programación extrema (rápida). Como requisito para alcanzar el éxito del proyecto se tiene al usuario final como parte del equipo. Es una metodología con reconocido éxito y se utiliza en proyectos con entregas a cortos plazos.

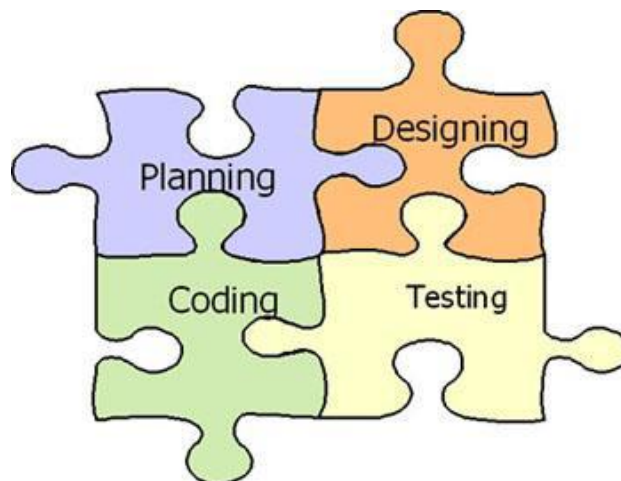


Ilustración 1 Fases de la metodología XP (9)

Las características fundamentales de XP son:

- Desarrollo iterativo e incremental: que 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 ciertas partes del código para aumentar su legibilidad y mantenibilidad.
- Propiedad del código compartida: no se dividen las responsabilidades en el desarrollo de cada módulo. Este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.
- Simplicidad en el código: es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.

Lo esencial en este proceso de desarrollo es lograr la comunicación entre desarrolladores y usuarios, la retroalimentación entre ellos y con los usuarios finales y la simplicidad en el código.

1.5.4.2 RUP

La metodología RUP es un proceso de desarrollo de software que unifica los mejores elementos de metodologías anteriores, el cual está preparado para el desarrollo de proyectos complejos y grandes, y es orientado a objetos. Asume tres características fundamentales que lo diferencian del resto de las metodologías: (10)

- **Dirigido por casos de uso:** Significa que el proceso sigue un hilo a través de flujos de trabajo que parten de los Casos de Uso. Guían el diseño, la implementación y las pruebas.
- **Centrado en la arquitectura:** Son las vistas en cada flujo de trabajo, que contiene los elementos fundamentales en cada uno de ellos.
- **Iterativo e incremental:** Es la estrategia para desarrollar un producto de software en pasos pequeños y manejables. Al iterar se decide si continuar o no con el proyecto, y se sabe si hubo incremento.

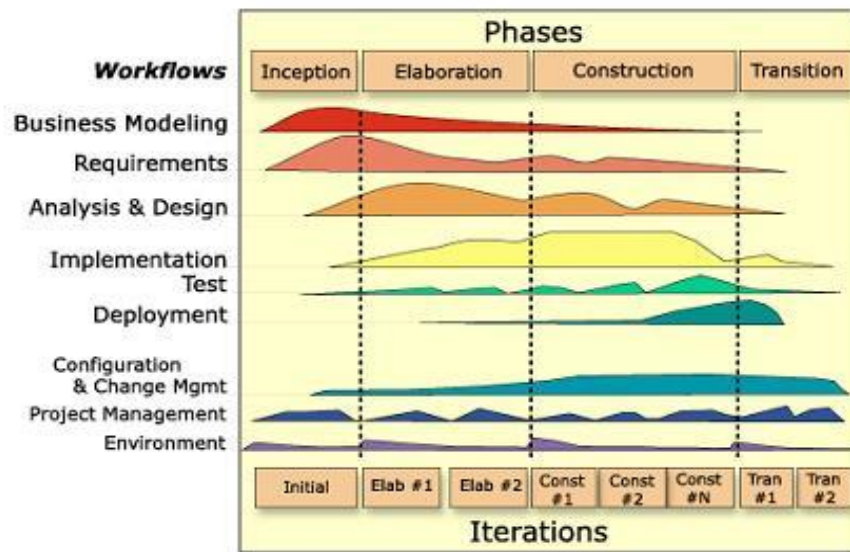


Ilustración 2 Fases de la metodología RUP(9)

RUP se divide en 4 fases de desarrollo del software:(10)

- Fase de **Inicio**: donde se trazan los objetivos del ciclo de vida.
- Fase de **Elaboración**: donde se define parte de la arquitectura.
- Fase de **Construcción**: donde se obtiene una funcionalidad operativa inicial.
- Fase de **Transición**: donde se obtiene una versión del producto.

Durante las fases de desarrollo se llevan a cabo los flujos de trabajo o disciplinas de desarrollo que RUP define como (10)

- **Modelado del Negocio (o empresarial)**: Se comprende la estructura dinámica de la organización, así como sus problemas, y se identifican las mejoras potenciales.
- **Requerimientos**: Establecer y mantener un acuerdo entre clientes sobre lo que el sistema debe hacer. Proveer a los desarrolladores de un mayor entendimiento de los requerimientos del sistema.
- **Análisis y Diseño**: Resolver problemas del negocio, refinando los requisitos, usando el lenguaje de los desarrolladores. Proporcionar una comprensión detallada de los requisitos.
- **Implementación**: Implementación de los elementos del diseño y probar componentes independientes.
- **Prueba**: Verificar los resultados de la implementación probando cada construcción.

- **Despliegue:** Describe las actividades asociadas al garantizar que el producto de software esté disponible para los usuarios.

Se utilizan además las disciplinas de soporte:

- **Gestión de cambios y configuración:** Explica cómo controlar y sincronizar la evolución del conjunto de productos de trabajo que componen un sistema de software.
- **Gestión de proyectos:** Se centra en la planificación del proyecto, la gestión del riesgo, la supervisión del progreso y las métricas.
- **Entorno:** Organiza esos elementos de método que proporcionan el entorno de desarrollo de software que da soporte al equipo de desarrollo, incluidos los procesos y las herramientas.

RUP define cuatro bases fundamentales que intervienen durante todo el desarrollo del proyecto:
(10)

Trabajadores (son el “Quién” del proceso): Definen el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Son los encargados de llevar a cabo las actividades y son propietarios de elementos.

Actividades (“Cómo” se lleva a cabo el proceso): Es una tarea que tiene un propósito definido, que es realizada por los trabajadores.

Artefactos (“Qué” se produce durante el proceso): Productos tangibles del proyecto que son producidos, modificados y usados por los trabajadores, ya sea modelos, código fuente y ejecutables.

Flujo de actividades (“Cuándo” llevar a cabo las actividades en el proceso): Secuencia de actividades realizadas por los trabajadores que produce un resultado de valor observable.

Existen aproximadamente 50 artefactos que resultan al utilizar esta metodología, lo que la convierte en una de las más importantes para alcanzar un grado de certificación en el desarrollo de software. (9)

1.5.4.3 MSF

Microsoft Solution Framework (MSF), es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de

proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas.(9)

MSF tiene las siguientes características:

- **Adaptable:** es parecido a un compás, usado en cualquier parte como un mapa, del cual su uso es limitado a un específico lugar.
- **Escalable:** puede organizar equipos tan pequeños entre 3 o 4 personas, así como también, proyectos que requieren 50 personas a más.
- **Flexible:** es utilizada en el ambiente de desarrollo de cualquier cliente.
- **Tecnología Agnóstica:** porque puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología.

MSF se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el modelo de Aplicación. (9)

1. **Modelo de Arquitectura del Proyecto:** Diseñado para acortar la planificación del ciclo de vida. Este modelo define las pautas para construir proyectos empresariales a través del lanzamiento de versiones.
2. **Modelo de Equipo:** Este modelo ha sido diseñado para mejorar el rendimiento del equipo de desarrollo. Proporciona una estructura flexible para organizar los equipos de un proyecto. Puede ser escalado dependiendo del tamaño del proyecto y del equipo de personas disponibles.
3. **Modelo de Proceso:** Diseñado para mejorar el control del proyecto, minimizando el riesgo, y aumentar la calidad acortando el tiempo de entrega. Proporciona una estructura de pautas a seguir en el ciclo de vida del proyecto, describiendo las fases, las actividades, la liberación de versiones y explicando su relación con el Modelo de equipo.
4. **Modelo de Gestión del Riesgo:** Diseñado para ayudar al equipo a identificar las prioridades, tomar las decisiones estratégicas correctas y controlar las emergencias que puedan surgir. Este modelo proporciona un entorno estructurado para la toma de decisiones y acciones valorando los riesgos que puedan provocar.
5. **Modelo de Diseño del Proceso:** Diseñado para distinguir entre los objetivos empresariales y las necesidades del usuario. Proporciona un modelo centrado en el usuario para obtener un diseño eficiente y flexible a través de un enfoque iterativo. Las fases de diseño conceptual, lógico y físico

proveen tres perspectivas diferentes para los tres tipos de roles: los usuarios, el equipo y los desarrolladores.

6. Modelo de Aplicación: Diseñado para mejorar el desarrollo, el mantenimiento y el soporte, proporciona un modelo de tres niveles para diseñar y desarrollar aplicaciones software. Los servicios utilizados en este modelo son escalables, y pueden ser usados en un solo ordenador o incluso en varios servidores. (9)

1.5.4.4 Scrum

Scrum es un proceso en el que se aplican de manera regular un conjunto de mejores prácticas para trabajar en equipo y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos. (11)

En Scrum se realizan entregas parciales del resultado final del proyecto, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad y la productividad son fundamentales. (11)

Scrum también se utiliza para resolver situaciones en que no se está entregando al cliente lo que necesita, cuando las entregas se alargan demasiado, los costes se disparan o la calidad no es aceptable, cuando se necesita capacidad de reacción ante la competencia, cuando la moral de los equipos es baja y la rotación alta, cuando es necesario identificar y solucionar ineficiencias sistemáticamente o cuando se quiere trabajar utilizando un proceso especializado en el desarrollo de producto.

En Scrum un proyecto se ejecuta en bloques temporales cortos y fijos (iteraciones de un mes natural y hasta de dos semanas, si así se necesita). Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite. (11)

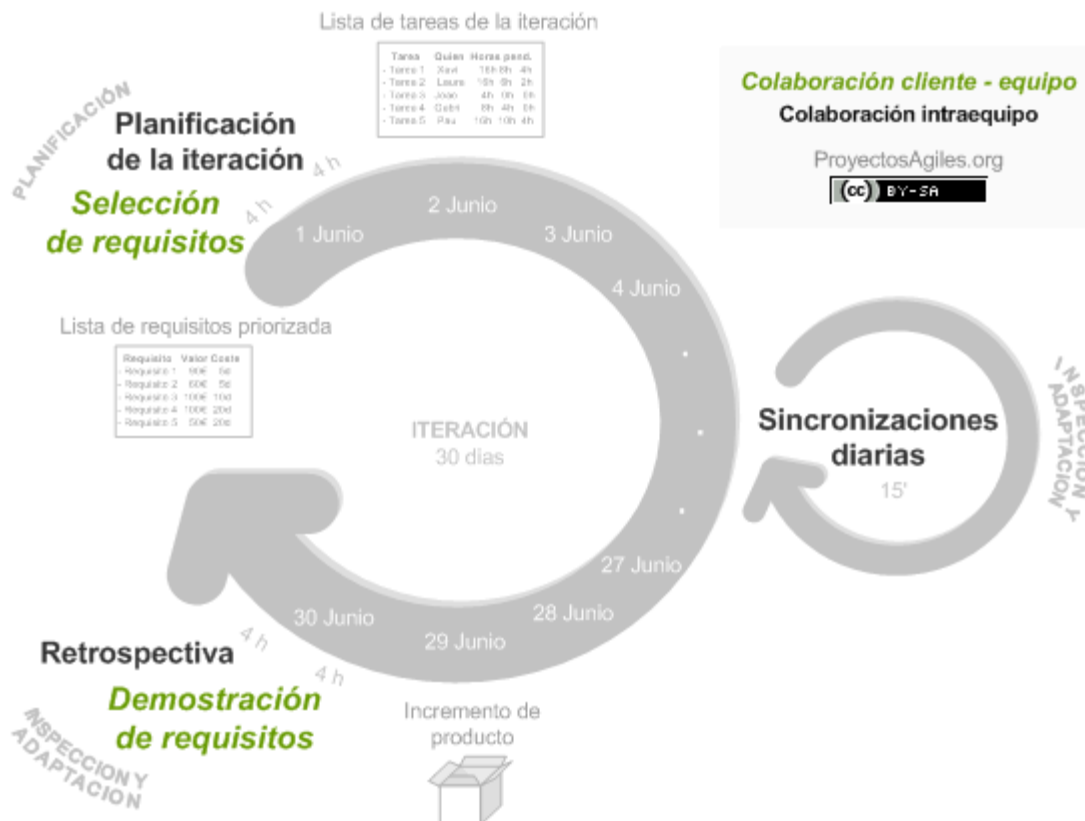


Ilustración 3 El proceso de Scrum(11)

El proceso parte de la lista de objetivos/requisitos priorizada del producto, que actúa como plan del proyecto. En esta lista el cliente prioriza los objetivos balanceando el valor que le aportan respecto a su coste y quedan repartidos en iteraciones y entregas. De manera regular el cliente puede maximizar la utilidad de lo que se desarrolla y el retorno de inversión mediante la replanificación de objetivos que realiza al inicio de cada iteración.

Las actividades que se llevan a cabo en Scrum son las siguientes:

Planificación de la iteración(11)

El primer día de la iteración se realiza la reunión de planificación de la iteración. Tiene dos partes:

1. **Selección de requisitos** (4 horas máximo). El cliente presenta al equipo la lista de requisitos priorizada del producto o proyecto. El equipo pregunta al cliente las dudas que surgen y selecciona los requisitos más prioritarios que se compromete a completar en la iteración, de manera que puedan ser entregados si el cliente lo solicita.

2. **Planificación de la iteración** (4 horas máximo). El equipo elabora la lista de tareas de la iteración necesarias para desarrollar los requisitos a que se ha comprometido. La estimación de esfuerzo se hace de manera conjunta y los miembros del equipo se autoasignan las tareas. (11)

Ejecución de la iteración(11)

Cada día el equipo realiza una reunión de sincronización (15 minutos máximos). Cada miembro del equipo inspecciona el trabajo que el resto está realizando (dependencias entre tareas, progreso hacia el objetivo de la iteración, obstáculos que pueden impedir este objetivo) para poder hacer las adaptaciones necesarias que permitan cumplir con el compromiso adquirido. En la reunión cada miembro del equipo responde a tres preguntas:

- ¿Qué he hecho desde la última reunión de sincronización?
- ¿Qué voy a hacer a partir de este momento?
- ¿Qué impedimentos tengo o voy a tener?

Durante la iteración el Facilitador se encarga de que el equipo pueda cumplir con su compromiso y de que no se merme su productividad.

- Elimina los obstáculos que el equipo no puede resolver por sí mismo.
- Protege al equipo de interrupciones externas que puedan afectar su compromiso o su productividad.

Inspección y adaptación (11)

El último día de la iteración se realiza la reunión de revisión de la iteración. Tiene dos partes:

1. **Demostración** (4 horas máximo). El equipo presenta al cliente los requisitos completados en la iteración, en forma de incremento de producto preparado para ser entregado con el mínimo esfuerzo. En función de los resultados mostrados y de los cambios que haya habido en el contexto del proyecto, el cliente realiza las adaptaciones necesarias de manera objetiva, ya desde la primera iteración, replanificando el proyecto.
2. **Retrospectiva** (4 horas máximo). El equipo analiza cómo ha sido su manera de trabajar y cuáles son los problemas que podrían impedirle progresar adecuadamente, mejorando de manera continua su productividad. El Facilitador se encargará de ir eliminando los obstáculos identificados.

1.5.4.5 Selección de la metodología a utilizar

Se decidió utilizar XP debido a que se adapta en gran medida, tanto al tipo de proyecto a desarrollar como a las condiciones de trabajo. Uno de los principios básicos de XP es que el cambio frecuente de los requerimientos es algo normal en el proceso de desarrollo y ésta metodología se adapta perfectamente a los proyectos cuyos requerimientos cambian a menudo. La planificación es mas transparente pues los clientes forman parte del equipo de trabajo y no tiene sobrecarga de artefactos, que en ocasiones resulta molesta.

1.5.5 Lenguaje de modelado

La comunidad del software precisa de una forma de comunicar sus modelos, no solo entre los miembros de un proyecto, sino a todas las personas involucradas en el, necesita un lenguaje para proporcionar un marco en el que desarrolladores individuales puedan pensar y analizar .Es por esto que surgen numerosos lenguajes de modelado. Algunos de ellos son:

1.5.5.1 Process Modeling Language (PML)

Es un lenguaje de modelado orientado a objetos diseñado para describir el comportamiento del sistema físico mediante estructuras de representación modulares (clases de modelado). Las clases PML representan conceptos físicos que son familiares al modelador. El conocimiento físico declarado por las clases se utiliza para analizar los modelos estructurados, obteniéndose de manera automatizada la representación matemática de las dinámicas de interés.

Modélica: Surge a mediados de la década del noventa debido a uno de los grandes problemas con todas las herramientas de modelado que iban surgiendo, la falta de transportabilidad de los modelos desarrollados que eran dependientes del entorno utilizado. Fue definido por un grupo de muchos expertos de diferentes dominios de la ingeniería, así como de la gran mayoría de desarrolladores de los lenguajes de modelado orientado a objetos que habían ido surgiendo, que tenían por objetivo crear un lenguaje de modelado capaz de expresar la conducta de modelos de un amplio abanico de campos de la ingeniería y sin limitar a los modelos a una herramienta comercial en particular. Se puede considerar a Modélica no solo como un lenguaje de modelado sino como una especificación de dominio público que permite el intercambio de modelos. Es un lenguaje de modelado que no tiene propietario y su nombre es una marca

registrada de la “Modelica Association” que es la responsable de la publicación de la especificación del lenguaje Modelica (12)

1.5.5.2 Lenguaje de Modelado para Realidad Virtual (VRML)

Se utiliza para describir simulaciones interactivas de participantes múltiples, esto es, mundos virtuales enlazados de manera global vía Internet e hiperenlazados con el World Wide Web. Puede utilizarse para especificar todos los aspectos del despliegue del mundo virtual, su interacción y trabajo con redes internas. La intención de sus diseñadores es hacer del VRML el lenguaje estándar para la simulación interactiva, dentro del World Wide Web. (13)

1.5.5.3 Lenguaje Unificado de Modelado (UML)

Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. Es importante resaltar que UML es un "lenguaje" para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional o RUP), pero no especifica en sí mismo qué metodología o proceso usar.

1.5.5.4 Selección del lenguaje de modelado

Este lenguaje es el más utilizado a nivel mundial para el modelar los artefactos creados durante el proceso de desarrollo de software. Además, es importante destacar que los demás lenguajes de modelado no responden a las necesidades del proyecto, es por ello que para la realización de este trabajo se adopta UML como lenguaje de modelado.

1.5.6 Herramienta Case

Hoy en día, la Ingeniería de Software cuenta con una serie de herramientas automatizadas destinadas a diferentes propósitos. La utilización de herramientas CASE (Computer Aided Software Engineering),

ingeniería de software asistida por computadora, se ha difundido ampliamente en el ámbito del desarrollo de software.

La ingeniería de sistemas asistida por ordenador es la aplicación de tecnología informática a las actividades, las técnicas y las metodologías propias de desarrollo, su objetivo es acelerar el proceso para el que han sido diseñadas, en el caso de las CASE para automatizar o apoyar una o más fases del ciclo de vida del desarrollo de sistemas (Kendall, 1997). Acerca de las herramientas automatizadas para la ingeniería de software se puede decir que:

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

Estas herramientas permiten a los desarrolladores modelar y documentar sus artefactos, cubriendo el ciclo de vida del proceso de desarrollo de software. A pesar de ello, presentan diferencias que contribuyen a seleccionar una herramienta según las características del proyecto.

1.5.6.1 Visual Paradigm

Esta herramienta tiene unas características graficas muy cómodas, que facilitan la realización de los diagramas de modelado que sigue el estándar de UML, los mismo son: Diagramas de clase, Casos de Uso, Comunicación, Secuencia, Estado, Actividad, Componentes, etc.

Se integra con las siguientes herramientas Java:

- Eclipse/IBM WebSphere
- JBuilder
- NetBeans IDE
- Oracle JDeveloper
- BEA Weblogic

Visual Paradigm ofrece:

- Entorno de creación de diagramas para UML 2.0.
- Diseño centrado en casos de uso y enfocado al negocio que genera 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, para cada necesidad.
- Disponibilidad en múltiples plataformas.

1.5.6.2 Rational Rose

Esta herramienta que domina en el mercado mundial, fue desarrollada por los creadores de UML y emplea este lenguaje para modelar. Es una de las más potentes desarrolladas hasta el momento, pero consume muchos recursos, lo que hace que si no se presenta la tecnología adecuada, resulte incómodo trabajar con ella.

1.5.6.3 Enterprise Architect (EA)

Es una herramienta CASE para el diseño y construcción de sistemas de software.

Beneficios de EA (14)

- Soporta la especificación de UML 2.0, que describe un lenguaje visual por el cual se pueden definir mapas o modelos de un proyecto, y los diagramas de UML 2.1.
- Es una herramienta progresiva que cubre todos los aspectos del ciclo de desarrollo, proporcionando una trazabilidad completa desde la fase inicial del diseño a través del despliegue y mantenimiento.
- También provee soporte para pruebas, mantenimiento y control de cambio.
- Sincroniza código y elementos del modelo.
- Diseña y genera elementos de base de datos.
- Sus modelos se pueden exportar en formato RTF para una personalización y presentación final.
- Sus plantillas soportan las características de los elementos de los modelos de EA y datos extendidos (pruebas, riesgos, recursos, cambios), además de soportar formatos como encabezados, índices, etc.
- Permite exportar un modelo completo o una rama simple del modelo a páginas web en HTML.
- Realiza ingeniería directa e inversa de código fuente en Action Script, C++, C#, Delphi, Java, Python, PHP, VB.NET, Visual Basic.
- Tiene una alta capacidad de sincronización de código.

- Posee soporte para Corba también disponible como plug-in libre.
- Posee plug-ins para vincular EA a Visual Studio.NET y Eclipse.
- Posee una interfaz de usuario intuitiva.
- Realiza ingeniería inversa para sistemas de Base de Datos muy populares como Oracle, SQL Server, MySQL, Access, PostgreSQL, etc.
- Permite generar tablas del Modelo de Base de Datos, columnas, claves, llaves foráneas, etc.
- Permite generar los scripts DLL para crear las estructuras de Base de Datos.
- Posibilita el desarrollo distribuido, es decir, es multiusuario.
- Soporta diferentes repositorios basados en DBMS, incluyendo Oracle, SQL Server, MySQL, PostgreSQL.
- Soporta importar/exportar archivos XMI para manejar la distribución y actualización de frameworks y otros paquetes basados en la estructura del modelo.
- Soporta control de versiones de repositorios.
- Soporta importar archivos .JAR en java y ensamblados .NET.
- Posee un perfil incorporado para XSD para simplificar el desarrollo de esquemas XML usando UML.
- Perfil UML para XSD disponible para descarga libre.
- Generar esquemas complejos de XML de modelos UML
- Ingeniería Reversa de esquema XML a Modelos UML.
- Ingeniería Directa de esquema XML desde Modelos UML
- Posee gran velocidad, estabilidad y buen rendimiento

1.5.6.4 Selección de la herramienta case a utilizar

Visual Paradigm es la herramienta a utilizar porque permite realizar ingeniería tanto directa como inversa ya que a partir de un modelo relacional en PostgreSQL o cualquier gestor de base de datos es capaz de desplegar todas las clases asociadas a las tablas. Cabe destacar igualmente su robustez, usabilidad y portabilidad. Esta herramienta no es indispensable para el desarrollo del sistema, pero con la realización de algunos de sus artefactos hace más entendible el diseño de la aplicación.

1.5.7 Framework

En general, con el término framework, nos estamos refiriendo a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta. (15)

Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.

Un framework Web, por tanto, podemos definirlo como un conjunto de componentes (por ejemplo clases en java y descriptores y archivos de configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas web.

Existe un gran variedad de Framework entre los cuales se encuentran: CakePHP, Zend Framework y Symfony

1.5.7.1 Symfony

Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, 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 todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web.

Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows. A continuación se muestran algunas de sus características. (16)

Symfony se diseñó para que se ajustara a los siguientes requisitos:

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y *nix estándares)

- Independiente del sistema gestor de bases de datos
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos
- Basado en la premisa de "*convenir en vez de configurar*", en la que el desarrollador solo debe configurar aquello que no es convencional
- Sigue la mayoría de *mejores prácticas* y patrones de diseño para la web
- Preparado para aplicaciones empresariales y adaptable a las políticas y arquitecturas 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 muy sencillo
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros

1.5.7.2 Zend Framework

Es un framework para desarrollo de aplicaciones Web y servicios Web con PHP, brinda soluciones para construir sitios web modernos, robustos y seguros. Además es Open Source y trabaja con PHP 5. A diferencia de CakePHP que trabaja con PHP 4 y PHP 5. (17)

Zend Framework es desarrollado por Zend que es la empresa que respalda comercialmente a PHP y entre sus principales características:

- Trabaja con MVC (Model View Controller)
- Cuenta con módulos para manejar archivos PDF, canales RSS, Web Services (Amazon, Flickr, Yahoo), etc.
- El Marco de Zend también incluye objetos de las diferentes bases de datos, por lo que es extremadamente simple para consultar su base de datos, sin tener que escribir ninguna consulta SQL.
- Una solución para el acceso a base de datos que balancea el ORM con eficiencia y simplicidad. Se puede ver esta función en el futuro.
- Completa documentación y test de alta calidad.
- Soporte avanzado para i18n (internacionalización).
- Un buscador compatible con Lucene.

- Robustas clases para autenticación y filtrado de entrada.
- Clientes para servicios web, incluidos Google Data APIs y Strikelron.
- Muchas otras clases útiles para hacerlo tan productivo como sea posible.

1.5.7.3 Cake PHP

CakePHP es un framework para php que nos permite programar más rápido evitándonos escribir código tedioso de tareas muy comunes.(18)

1. Características más destacadas:

- Arquitectura MVC (modelo, vista y controlador)
- Helpers para AJAX, Javascript, formularios HTML y más
- Validación nativa
- El código para CRUD (crear, leer, actualizar y eliminar) se llevan a cabo a través de CakePHP
- Listas de Control de Acceso
- Sanación de datos
- Componentes para el manejo de la seguridad, sesiones y peticiones.
- Caché flexible
- y más...

2. Licencia flexible - CakePHP está distribuido bajo la MIT License

3. IP limpia - Cada línea de código está escrita por el equipo de desarrollo de CakePHP

4. Extremadamente simple - Sólo ve el nombre es Cake (pastel)

5. Desarrollo rápido - Construye aplicaciones más rápido que antes.

6. Buenas prácticas - Cake es muy fácil de entender y cumple los estándares en seguridad y autenticación, manejo de sesiones y muchas otras características.

7. Orientado a Objetos - Si te gusta la programación orientada a objetos que bien y si eres principiante te sentirás cómodo.

8. Cero Configuración - Solamente pon la información de la base de datos y la magia comenzará.

1.5.7.3 Selección del framework a utilizar

Se utilizará el framework Symfony porque es uno de los más utilizados en el mundo posee gran cantidad de documentación disponible. Es recomendable para desarrollar aplicaciones web complejas para asegurar el mantenimiento y las ampliaciones futuras de la aplicación.

1.6 Conclusiones

Con el estudio teórico se ha podido observar las tendencias de los diferentes sistemas contables tanto internacionales como nacionales, arribando a la conclusión de la importancia de realizar un sistema contable para la empresa Cuba Ron, pues ninguna de los estudiados cuenta con los requisitos planteados por la empresa. El avance de los medios y tecnologías enfocados a la Web se encuentran en un desarrollo acelerado lo que permitió que se haya podido seleccionar herramientas y tecnologías que permitan realizar un sistema con los requisitos requeridos.

CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

2.1 Introducción

Este capítulo se centra en las características principales del sistema, tiene como objetivo mostrar el flujo de los datos primarios en el proceso contable, mediante la descripción de funcionalidades para la comprensión del problema a resolver. Se lleva a cabo el levantamiento de requisitos tanto funcionales como no funcionales mostrando una mejor visión del proceso en cuestión. En este capítulo también se aborda la fase de exploración y planificación de la metodología XP. Durante esta etapa se identifican las historias de usuarios, se realiza la estimación del esfuerzo que costará implementar cada historia de usuario y las iteraciones en las que se dividirá el proceso.

2.2 Descripción de los procesos vinculados al campo de acción.

Actualmente el sistema de gestión de los datos primarios en la empresa Cuba Ron requiere de una serie de funcionalidades y operaciones, que permiten el flujo de los mismos; esto posibilita que en la empresa se tenga un control sobre los datos primarios de sus producciones y el trabajo se haga más fluido.

2.2.1 Flujo actual del proceso.

El proceso de gestión de los datos primarios se realiza de la siguiente manera:

Se procesa diariamente la información de un centro de costo productivo (fábrica, planta o taller) en unidades físicas, almacenando los datos acumulados en dispositivos de almacenamiento extraíbles, los cuales se envían a la empresa. Seguidamente la empresa procesa convenientemente ciertos indicadores físicos, específicos de cada centro de costo, que contribuyen al análisis del costo de producción, para someter toda la captura de datos a importantes validaciones que permitan una buena exactitud y confiabilidad.

Estos datos dan la posibilidad de obtener diariamente el movimiento de los productos semielaborados, en proceso y terminados, además del tiempo perdido, y otros indicadores específicos de cada centro de costo, pudiendo posteriormente emitir informes y balances.

2.2.2 Objeto de automatización

Existen varios procesos que deben ser automatizados, pues de realizarse de forma manual atrasaría el trabajo, además pueden surgir errores o pérdidas de informaciones importantes.

Entre los procesos que serán automatizados se encuentra la gestión de los productos, materias primas y normas de consumo que permiten cómodamente acceder a los diferentes datos donde sea necesario, desglosar en todos sus detalles las operaciones ocurridas por centro de costo para, después de rigurosa validación, almacenar la información.

El proceso de autenticación es uno de los más importantes pues asegura que el usuario acceda a la información según el rol que desempeña en la aplicación, garantizando la integridad y confidencialidad de los datos almacenados. Además es necesario tener recogidos los informes emitidos diariamente, para que en caso de pérdida se tengan los datos de las producciones controlados.

2.3 Propuesta del sistema

El presente trabajo propone el desarrollo de una aplicación web, que debe permitir la gestión (crear, modificar y eliminar) de productos, normas de consumo y materias primas, recogiendo datos indispensables. Además de proporcionar la posibilidad al cliente de realizar operaciones entre las cuales se encuentran ingresar los indicadores específicos y el tiempo perdido diariamente, así como mostrar los planes de producción donde se dispone el listado de productos finales por cada centro de costo. El sistema ofrece un grupo de reportes o tablas de salida que permiten conocer, de manera organizada y práctica, la situación de la producción y muchos otros aspectos del control primario en una fábrica, entre los que se encuentra Tiempo perdido que tiene como finalidad clasificar las posibles causas de interrupciones en la producción, de manera que pueda obtenerse esta información, en el período que se desee, como factor importante a la hora de analizar las desviaciones en el costo.

2.3.1 Personas relacionadas con el sistema

Se define como persona relacionada al sistema a cualquier individuo que interactúa con el sistema, donde obtiene un resultado con el cual se beneficia; además desarrolla las acciones que posibilitan se cumpla la función de un proceso.

Personas relacionadas con el sistema	Justificación
Presidencia Cuba Ron	Es la encargada del funcionamiento de todas sus empresas
Contador principal	Es el encargado de registrar cada operación realizada en la empresa y emitir los reportes.
Contador UEB	Se ocupa de controlar los costos de la producción vinculados a su área de trabajo.
Administrador	Encargado de la administración del sistema.

Tabla 2. 1 Personas relacionadas al sistema

2.3.2 Requerimientos funcionales del sistema

Una vez conocidos todos los conceptos que rodean al objeto de estudio, se puede analizar qué debe hacer el sistema para que se cumplan los objetivos planteados al inicio de este trabajo. Para ello se enumeran a través de requerimientos funcionales las prestaciones que el sistema será capaz de brindar. A pesar que la metodología no requiere identificar estos requisitos serán plasmados para mayor comprensión ya que son los puntos importantes a desarrollar.

R1 Autenticar

R2 Gestionar Usuario

R2.1 Insertar Usuario: adiciona un nuevo producto al sistema

R2.2 Eliminar Usuario: elimina un usuario en el sistema.

R3 Gestionar Productos

R3.1 Insertar Productos: adiciona un nuevo producto al sistema.

R3.2 Modificar Productos: modifica un producto en el sistema.

R3.3 Eliminar Productos: elimina un producto en el sistema.

R3.4 Buscar Productos: realiza búsquedas de productos.

R4 Gestionar Norma de Consumo.

- R4.1 Insertar Normas de Consumo: adiciona una nueva norma de consumo.
- R4.2 Modificar Normas de Consumo: modifica una norma de consumo.
- R4.3 Eliminar Normas de Consumo: elimina una norma de consumo.
- R4.4 Buscar Normas de Consumo: realiza búsquedas de normas de consumo.

R5 Gestionar Materias Primas.

- R5.1 Insertar Materias Primas: adiciona una nueva materia prima.
- R5.2 Modificar Materias Primas: modifica una materia prima.
- R5.3 Eliminar Materias Primas: elimina una materia prima.
- R5.4 Buscar Materias Primas: realiza búsquedas de materias primas.

R6 Movimiento Diario de Productos

- R6.1 Obtener indicadores específicos diarios
- R6.2.Obtener causas de tiempo perdido diarios

R7 Mostrar Planes de Producción

- R7.1 Mostrar Planes de producción.
- R7.2 Mostrar Planes de entrega.
- R7.3 Mostrar Causas de Ajustes
- R7.4 Mostrar Causas de Otras Operaciones

R8 Mostrar Reportes

- R8.1 Reporte Tiempo Perdido Mensual
- R8.2 Reporte Indicadores Específicos Mensual

2.3.3 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.

Usabilidad:

El diseño del sistema ha sido concebido para que los usuarios en todo momento puedan tener control del sitio Web, evitando, de esa manera, la sobrecarga de información y para cada proceso una muestra a través de mensaje en las acciones hechas por los usuarios.

Rendimiento:

La disponibilidad de trabajo en red contra el servidor es constante, teniendo como prioridad garantizar que el tiempo de respuesta a las solicitudes de los usuarios al sistema sea mínimo; o sea, en un período de tiempo breve (escasos segundos). De tal manera se evita la acumulación de trabajo por parte de los responsables. El sistema deberá ser lo más estable y confiable posible.

Soporte:

Es necesario que el producto reciba mantenimiento ante los posibles fallos que puedan surgir.

Portabilidad:

El sistema funcionará sobre plataforma Windows, en caso de cambio de plataforma por pedidos del cliente, el sistema se ajustará totalmente con la mayor brevedad, gracias a la universalidad de las herramientas utilizadas que pueden ser soportadas tanto en software libre como software propietario. El producto está implantado sobre una plataforma Web, codificada en "PHP 5.0" y sus sistemas de base de datos en PostgreSQL.

Seguridad:

La información manejada por el sistema contará de protección ante intrusos y accesos no autorizados, será vista únicamente por aquellos usuarios que tengan derecho a verla. El sistema controlará los diferentes niveles de acceso y funcionalidad de usuarios al sitio; o sea, prioriza la identificación del usuario antes de que sea capaz de realizar cualquier acción sobre el sistema.

Extensibilidad:

Se garantizará la inserción de módulos nuevos en el futuro, ello sin afectar lo realizado hasta el momento y el buen funcionamiento. Las nuevas funcionalidades deberán estar debidamente documentadas.

Interfaz:

La interfaz debe ser sencilla, clara, intuitiva y amigable, manteniendo un formato estándar en todas las páginas. No contará con gran cantidad de imágenes, garantizando que el tiempo de descarga sea mínimo y así no demorar las respuestas al usuario. Teniendo en cuenta el fin para el que se desarrolla, la aplicación debe ser seria, formal y tener una navegación sugerente.

Confidencialidad:

Toda la información estará protegida del acceso no autorizado, los administradores del sistema son los únicos que podrá transformar la información con respecto a las solicitudes, los investigadores solo podrán ver las listas de información y en casos específicos predefinidos realizar las solicitudes correspondientes.

2.4 Historias de usuario

Las historias de usuario tienen el mismo propósito que los casos de uso, pero no son lo mismo. Las escriben los propios clientes, tal y como ven ellos las necesidades del sistema, aunque los desarrolladores pueden brindar su ayuda en la identificación de las mismas. Por tanto serán descripciones cortas y escritas en el lenguaje del usuario, sin terminología técnica. Son usadas para estimar tiempos de desarrollo de la parte de la aplicación que describen. También se utilizan en la fase de pruebas, para verificar si el programa cumple con lo que especifica la historia de usuario. Cuando llega la hora de implementar una historia de usuario, el cliente y los desarrolladores se reúnen para concretar y detallar lo que tiene que hacer dicha historia. El tiempo de desarrollo ideal para una historia de usuario es entre 1 y 3 semanas. Estas historias de usuarios describirán los requisitos funcionales planteados anteriormente.

A continuación se describen las historias de usuario

Historia de Usuario	
Número: 1	Nombre: Autenticar
Usuario: Contador UEB, Contador Principal, Presidencia Cuba Ron	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Bajo
Puntos de Estimación: 1	Iteración Asignada: 1
Descripción: Se pide nombre de usuario y contraseña para entrar a la aplicación según el rol que represente.	
Observaciones: Si el usuario y/o contraseña son inválidos, tiene la posibilidad de corregirlos	

Tabla 2. 2 HU Autenticar

Historia de Usuario	
Número: 2	Nombre: Gestionar usuario
Usuario: Administrador	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Bajo
Puntos de Estimación: 1	Iteración Asignada: 1
Descripción: Se realizan las acciones de insertar y eliminar	
Observaciones:	

Tabla 2. 3 HU Gestionar usuario del sistema

Historia de Usuario	
Número: 3	Nombre: Gestionar productos en el sistema
Usuario: Contador principal	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Bajo
Puntos de Estimación: 2	Iteración Asignada: 1
Descripción: Se realizan las acciones de crear, modificar y eliminar los productos en el sistema.	
Observaciones:	

Tabla 2. 4 HU Gestionar productos en el sistema

Historia de Usuario	
Número: 4	Nombre: Gestionar normas de consumo en el sistema
Usuario: Contador principal	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Bajo
Puntos de Estimación: 2	Iteración Asignada: 1
Descripción: Se realizan las acciones de crear, modificar y eliminar las normas de consumo en el sistema.	
Observaciones:	

Tabla 2. 5 HU Gestionar normas de consumo

Historia de Usuario	
Número: 5	Nombre: Gestionar materias primas en el sistema
Usuario: Contador principal	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Bajo
Puntos de Estimación: 2	Iteración Asignada: 1
Descripción: Se realizan las acciones de crear, modificar y eliminar las materias primas en el sistema.	
Observaciones:	

Tabla 2. 6 HU Gestionar materias primas en el sistema

Historia de Usuario	
Número: 6	Nombre: Obtener indicadores específicos diarios
Usuario: Contador Principal, Contador UEB	
Prioridad en negocio: Media	Riesgo en Desarrollo: Medio
Puntos de Estimación: 1	Iteración Asignada: 2
Descripción: Se manifiestan la información que requiere cada indicador de acuerdo a la unidad de medida en que se expresa.	
Observaciones:	

Tabla 2. 7 HU Obtener indicadores específicos

Historia de Usuario	
Número: 7	Nombre: Obtener causas de tiempo perdido diarios
Usuario: Contador Principal, Contador UEB	
Prioridad en negocio: Media	Riesgo en Desarrollo: Medio
Puntos de Estimación: 1	Iteración Asignada: 2
Descripción: Se captura diariamente la información real de las interrupciones operacionales.	
Observaciones:	

Tabla 2. 8 HU Obtener causas de tiempo perdido

Historia de Usuario

Número: 8	Nombre: Mostrar Planes de producción
Usuario: Contador Principal, Contador UEB, Presidencia Cuba Ron	
Prioridad en negocio: Media	Riesgo en Desarrollo: Bajo
Puntos de Estimación: 1	Iteración Asignada: 2
Descripción: Brinda la posibilidad de acceder al listado de productos finales por cada centro de costo.	
Observaciones:	

Tabla 2. 9 HU Mostrar planes de producción

Historia de Usuario	
Número: 9	Nombre: Mostrar Planes de entrega
Usuario: Contador Principal, Contador UEB, Presidencia Cuba Ron	
Prioridad en negocio: Media	Riesgo en Desarrollo: Medio
Puntos de Estimación: 1	Iteración Asignada: 2
Descripción: Muestra el valor de los productos que serán entregados por mes.	
Observaciones:	

Tabla 2. 10 HU Mostrar planes de entrega

Historia de Usuario	
Número: 10	Nombre: Reportes Tiempo Perdido Mensual
Usuario: Contador Principal, Contador UEB, Presidencia Cuba Ron	
Prioridad en negocio: Baja	Riesgo en Desarrollo: Medio
Puntos de Estimación: 1	Iteración Asignada: 3
Descripción: Muestra las horas perdidas de un mes de aquellos productos de un centro de costo especificado	
Observaciones:	

Tabla 2. 11 HU Mostrar Tiempo Perdido Mensual

Historia de Usuario	
Número: 11	Nombre: Reportes Indicadores Específicos Mensual

Usuario: Contador Principal, Contador UEB, Presidencia Cuba Ron	
Prioridad en negocio: Baja	Riesgo en Desarrollo: Medio
Puntos de Estimación: 1	Iteración Asignada: 3
Descripción: Este reporte muestra los indicadores específicos de un mes de un centro de costo determinado	
Observaciones:	

Tabla 2. 12 HU Mostrar Indicadores Específicos Mensual

Historia de Usuario	
Número: 12	Nombre: Mostrar Causas de Ajustes
Usuario: Contador Principal, Contador UEB, Presidencia Cuba Ron	
Prioridad en negocio: Baja	Riesgo en Desarrollo: Bajo
Puntos de Estimación: 1	Iteración Asignada: 3
Descripción: Muestra posibles motivos que ocasionan movimientos por Ajustes, por perdidas no esperadas y sujetas a investigación	
Observaciones:	

Tabla 2. 13 HU Mostrar Causas de ajustes

Historia de Usuario	
Número: 13	Nombre: Mostrar Causas de Otras Operaciones
Usuario: Contador Principal	
Prioridad en negocio: Baja	Riesgo en Desarrollo: Bajo
Puntos de Estimación: 1	Iteración Asignada: 3
Descripción: Muestra las causas que no necesariamente requieren expedientes investigativos ya que están asociados a operaciones permitidas aunque habitualmente ajenas al desarrollo normal del proceso productivo.	
Observaciones:	

Tabla 2. 14 HU Mostrar Causas de Otras Operaciones

2.5 Estimación de esfuerzo

La estimación de esfuerzo se expresa utilizando como medida el punto. Un punto se considera como una semana ideal de trabajo, donde los miembros de los equipos de desarrollo trabajan el tiempo planeado sin ningún tipo de interrupción. Esta estimación incluye todo el esfuerzo asociado a la implementación de la historia de usuario.

Para el logro del sistema se ha realizado la estimación de esfuerzo por historias de usuario las cuales se muestran a continuación

Historias de usuario	Puntos de estimación
Autenticar	1
Gestionar usuario del sistema	1
Gestionar productos en el sistema	2
Gestionar normas de consumo en el sistema	2
Gestionar materias primas en el sistema	2
Obtener indicadores específicos diarios	1
Obtener causas de tiempo perdido diarios	1
Mostrar Planes de producción	1
Mostrar Planes de entrega	1
Mostrar Tiempo Perdido Mensual	1
Mostrar Indicadores Específicos Mensual	1
Mostrar Causas de Ajustes	1
Mostrar Causas de Otras Operaciones	1

Tabla 2. 15 Estimación de esfuerzos por HU

2.6 Plan de iteraciones

Al estar descritas las historias de usuarios y haber estimado el costo de esfuerzo para cada una, se procede a realizar el plan de iteraciones, que no es más que las partes en las que se dividirá la implementación y que se muestran a continuación.

2.6.1 Iteración 1

Esta iteración tiene como objetivo la implementación de las historias de usuario de mayor prioridad; las cuales nos brinda la posibilidad de gestionar usuarios, productos, normas de consumo y materias primas. Con esta iteración el usuario puede además autenticarse. De esta manera se trata de tener preparadas las funcionalidades básicas e indispensables del sistema. Igualmente se tendrá la primera versión de pruebas.

2.6.2 Iteración 2

El objetivo de esta iteración es implementar las funcionalidades con prioridad media, las cuales permiten obtener los indicadores específicos y las causas del tiempo perdido de la producción, mostrar los planes de producción y de entrega. Las versiones de pruebas anteriores junto a estas se les mostrará al cliente.

2.6.3 Iteración 3

En esta iteración se implementarán las historias de usuario de baja prioridad. Estas funcionalidades brindan al sistema cierta comodidad en la gestión de las funcionalidades de prioridad alta, además de corregir errores anteriores. Esta iteración permitirá mostrar reportes como el tiempo que se ha perdido en los centros de costo así como los indicadores específicos en el mes. Además de mostrar las causas de ajustes y de otras operaciones A partir de este momento quedará conformado totalmente el sistema y sometido a pruebas rigurosamente

2.7 Plan de duración de iteraciones

El plan de duración de iteraciones tiene como finalidad mostrar la duración de cada iteración y el orden en que se implementarán las historias de usuario, teniendo una mayor organización

Iteraciones	Orden de las historias de usuario a implementar	Duración total de la iteración
Iteración 1	<ol style="list-style-type: none">1. Autenticar2. Gestionar usuario del sistema3. Gestionar productos4. Gestionar normas de consumo5. Gestionar materias primas	8 semanas

Iteración 2	<ol style="list-style-type: none"> 1. Obtener indicadores específicos diarios 2. Obtener causas de tiempo perdido diarios 3. Mostrar planes de producción 4. Mostrar planes de entrega 	4 semanas
Iteración 3	<ol style="list-style-type: none"> 1. Mostrar Tiempo Perdido Mensual 2. Mostrar Indicadores Específicos Mensual 3. Mostrar Causas de Ajustes 4. Mostrar causas de Otras operaciones 	4 semanas

Tabla 2. 16 Plan de duración de iteraciones

2.8 Plan de entregas

Con el plan de duración se da un aproximado de las entregas de las versiones, contando que la implementación empieza en el mes de Febrero

A entregar	Final 1ra iteración 4ta semana de Marzo	Final 2da iteración 4ta semana de abril	Final 3ra iteración 4ta semana de mayo
Sistema de gestión de datos primarios	0.1	0.2	1.0

Tabla 2. 17 Plan de duración de entrega

2.9 Conclusiones

En este capítulo se ha desarrollado la propuesta del sistema, detallando como ocurrirán los procesos que se llevan a cabo en la gestión de los datos primarios para la contabilización del costo de la producción para la empresa Cuba Ron. Se logró la especificación de los requerimientos tanto funcionales como no funcionales, los cuales pueden estar sujetos a cambio durante el transcurso del ciclo de desarrollo. Se ha abordado sobre la planificación del sistema concentrándose en historias de usuario, plan de entregas, las iteraciones en las cuales se decidirá el sistema para una mejor implementación, sin embargo tras esta fase nos ha quedado claro que es prácticamente imposible establecer una planificación inalterable de lo que es las entregas, es de esperar que la planificación mostrada en principio varíe así como la

desaparición y sustitución de algunas historias de usuario. La importancia del cliente resulta demostrada como máxima ya que el enfoque de los miembros del equipo puede llegar a tener una interpretación diferente.

CAPÍTULO 3: CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

3.1 Introducción

La metodología XP cuenta con la realización del diseño del sistema, centrándose en la arquitectura, además es necesario de la utilización de tarjetas C.R.C que contribuye a la tarea del diseño. Las tareas de la ingeniería constituyen pasos indispensables a seguir para desarrollar la aplicación.

3.2 Metáforas del sistema

En la empresa Cuba Ron es necesario tener una aplicación que permita obtener de cada centro de costo los productos ya sean elaborados, semielaborados o terminados, con los cuales, se puedan realizar operaciones contables y emitir reportes diarios

3.3 Arquitectura del sistema

Para desarrollo web, la solución más común para la organización de su código de hoy en día es el patrón de diseño MVC. En resumen, el patrón de diseño MVC define una manera de organizar el código de acuerdo a su naturaleza. Este patrón separa el código en tres capas:

- La capa **Modelo** define la lógica de negocio (la base de datos pertenece a esta capa). Symfony guarda todas las clases y archivos relacionados con el modelo en el directorio lib/model/.
- La **Vista** es con lo que el usuario interactúa (un motor de plantillas es parte de esta capa). En Symfony, la vista es principalmente la capa de plantillas PHP. Estas son guardadas en el directorio templates/.
- El **Controlador** es la pieza de código que llama al Modelo para obtener algunos datos que le pasa a la Vista para la presentación al cliente. Todas las solicitudes son gestionadas por un controlador frontal (index.php y Aplicacion_dev.php). Estos controladores frontales delegan la verdadera labor a las acciones, estas acciones son, lógicamente, agrupadas en módulos.

3.3.1 Patrones de diseño en Symfony

En el sistema se puede observar que cada una de las páginas tiene el mismo aspecto; para esto se utiliza el patrón decorator, donde la plantilla es decorada después de que el contenido es mostrado por una plantilla global, llamada layout en Symfony



Ilustración 4 Plantilla global en Symfony

El layout de una aplicación se llama layout.php y se puede encontrar en el directorio apps/Aplicación/templates/. Este directorio contiene todas las plantillas globales para una aplicación.

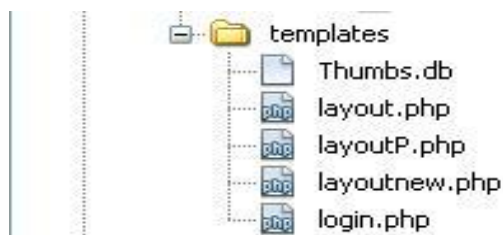


Ilustración 5 Plantillas globales de la aplicación

3.3.2 Estructura del sistema.

Para el entendimiento de este sistema se ha realizado una breve explicación sobre la estructura del framework Symfony.

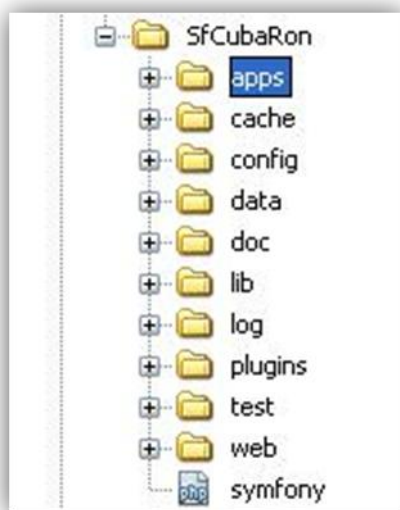


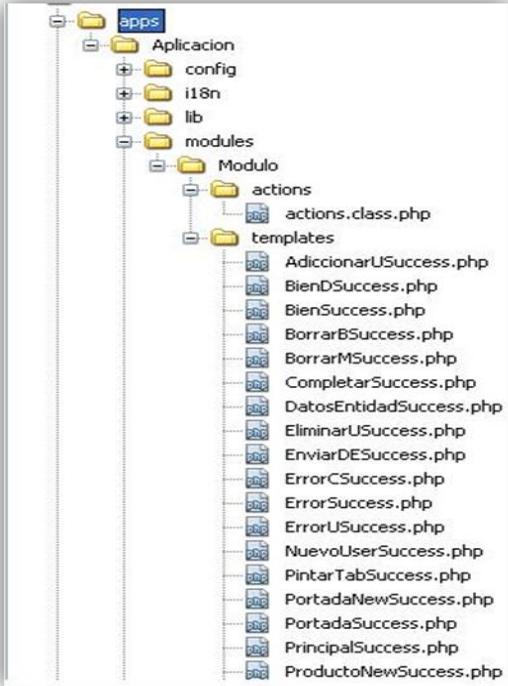
Ilustración 6 Estructura de Symfony

Directorio	Descripción
apps/	Hospeda todas las aplicaciones del proyecto
cache/	Los archivos en caché
config/	Los archivos de configuración del proyecto
lib/	Las bibliotecas y clases del proyecto
log/	Los archivos de registro
plug-ins/	Los plug-ins instalados
test/	Los archivos de pruebas unitarias y funcionales
web/	El directorio raíz web

Tabla 3. 1 Estructura del Symfony

¿Por qué Symfony genera tantos archivos? Uno de los principales beneficios de la utilización de un completo framework es normalizar los desarrollos. Gracias a estructura predeterminada de archivos y directorios de Symfony, cualquier desarrollador con algunos conocimientos de Symfony puede asumir el mantenimiento de cualquier proyecto. En cuestión de minutos, será capaz de bucear en el código, corregir los errores, y añadir nuevas funciones.

En el archivo apps se muestran las siguientes carpetas:



Directorio	Descripción
config/	Los archivos de configuración de la aplicación
lib/	Las bibliotecas y clases de la aplicación
modules/	El código de la aplicación (MVC)
templates/	La plantilla global
actions/	Acciones del módulo
templates/	Plantillas del módulo

Tabla 3. 2 Estructura del sistema a desarrollar

Ilustración 7 Estructura del sistema a desarrollar

El archivo actions/actions.class.php define todas las **acciones** disponibles

3.4 Diagramas de clases del diseño

Describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación. Los diagramas de clases del diseño se detallan en el [Anexo 1](#). Estos diagramas no son necesarios según esta metodología pero sirven de apoyo y entendimiento.

3.5 Diseño de la base de datos

Uno de los pasos cruciales en la construcción de una aplicación que maneje una base de datos, es sin duda, el diseño de la base de datos. Si las tablas no son definidas apropiadamente, se puede tener muchos problemas al momento de ejecutar consultas a la base de datos para tratar de obtener algún tipo de información.

Uno de los pasos importantes para desarrollar una base de datos es proporcionar un acceso eficaz a los datos, mantener la integridad de los datos a lo largo del tiempo y ser coherente, claro y de fácil comprensión. [\(Ver Anexo 2\)](#)

3.6 Tarjetas C.R.C

Estas tarjetas se dividen en tres secciones que contienen la información del nombre de la clase, sus responsabilidades y sus colaboradores.

Una clase es cualquier persona, cosa, evento, concepto, pantalla o reporte. Las responsabilidades de una clase son las cosas que conoce y las que realizan, sus atributos y métodos. Los colaboradores de una clase son las demás clases con las que trabaja en conjunto para llevar a cabo sus responsabilidades.

En la práctica conviene tener pequeñas tarjetas de cartón por ejemplo, que se llenarán y que se mostrarán al cliente, de manera que se pueda llegar a un acuerdo sobre la validez de las abstracciones propuestas.

A continuación se muestran las tarjetas correspondientes:

Productos	
Funcionalidades	Colaboraciones
Adicionar Productos	Centro de Costo
Modificar Productos	Materias Primas
Eliminar Productos	Normas de Consumo
Buscar Productos	
Mostrar Productos	

Tabla 3. 3 Tarjeta C.R.C Productos

Normas de Consumo	
Funcionalidades	Colaboraciones
Adicionar Normas de Consumo	Materias Primas
Modificar Normas de Consumo	Productos
Eliminar Normas de Consumo	
Buscar Normas de Consumo	
Mostrar Normas de Consumo	

Tabla 3. 4 Tarjeta C.R.C Norma de Consumo

Materias Primas	
Funcionalidades	Colaboraciones
Adicionar Materias Primas Modificar Materias Primas Eliminar Materias Primas Buscar Materias Primas Mostrar Materias Primas	Productos Tipo de Materia Prima

Tabla 3. 5 Tarjeta C.R.C Materias Prima

Centro de Costo	
Funcionalidades	Colaboraciones
Mostrar Reportes	Productos Indicadores Específicos Causa de Tiempo Perdido

Tabla 3. 6 Tarjeta C.R.C Centro de Costo

3.7 Tareas de la ingeniería

Para definir las tareas de la ingeniería se cuenta con una plantilla, la cual permite definir cada una de las actividades a las que estarán asociadas a las historias de usuario y que permitirán su implementación.

A continuación se mostrarán las tareas asignadas por iteraciones.

3.7.1 Iteración 1

Esta iteración desarrolla las historias de usuario que poseen prioridad alta con el fin de obtener una primera versión del producto, dando a conocer a los clientes las funcionalidades indispensables para el sistema, las cuales forman las bases de la siguiente iteración

Historias de usuario	Estimación	Real
Autenticar	1	1
Gestionar usuarios	1	1

Gestionar productos	2	2
Gestionar normas de consumo	2	2
Gestionar materias primas	2	2
Total	8	8

Tabla 3. 7 Historias de usuario desarrolladas en la primera iteración

Tareas de la Ingeniería	
Número de tarea: 1	Numero de historia de usuario:
Nombre de la tarea: Creación de la interfaz del sistema	
Tipo de tarea: Configuración	Puntos estimados:
Fecha inicio:	Fecha fin:
Programador responsable: Liusmy Cárdenas Medina – Abel Savigne Suarez	
Descripción: Se diseña la página principal (portada) del sistema.	

Tabla 3. 8 Tarea para crear la interfaz del sistema

Tareas de la Ingeniería	
Número de tarea: 2	Numero de historia de usuario: 1
Nombre de la tarea: Mostrar formulario para la autenticación	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 01/02/2009	Fecha fin: 8/02/2009
Programador responsable: Liusmy Cárdenas Medina - Abel Savigne Suarez	
Descripción: Mostrar formulario con los campos de usuario y contraseña y botón de aceptar, el cual debe permitir el acceso al sistema	

Tabla 3. 9 Tarea 2 Mostrar formulario de validación de cuentas

Tareas de la Ingeniería	
Número de tarea: 3	Numero de historia de usuario: 3
Nombre de la tarea: Diseñar el formulario para la gestión de un producto	

Tipo de tarea: Diseño	Puntos estimados: 1
Fecha inicio: 08/02/2009	Fecha fin: 15/02/2009
Programador responsable: Liusmy Cárdenas Medina - Abel Savigne Suarez	
Descripción: Se crean y habilitan todos los campos necesarios del formulario para la gestión del producto	

Tabla 3. 10 Tarea 3 Diseño del formulario para la gestión de un producto

Tareas de la Ingeniería	
Número de tarea: 4	Numero de historia de usuario: 3
Nombre de la tarea: Validar campos del formulario para la gestión de un producto	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 15/02/2009	Fecha fin: 22/02/2009
Programador responsable: Liusmy Cárdenas Medina - Abel Savigne Suarez	
Descripción: Verificar que los datos que se insertan en los campos son los correctos	

Tabla 3. 11 Tarea 4 Validar campos del formulario para la gestión de un producto

Tareas de la Ingeniería	
Número de tarea: 5	Numero de historia de usuario: 4
Nombre de la tarea: Diseñar el formulario para la gestión de una norma de consumo	
Tipo de tarea: Diseño	Puntos estimados: 1
Fecha inicio: 22/02/2009	Fecha fin: 01/03/2009
Programador responsable: Liusmy Cárdenas Medina - Abel Savigne Suarez	
Descripción: Se creará la plantilla que permitirá al usuario realizar diferentes operaciones con la normas de consumo.	

Tabla 3. 12 Tarea 5 Diseñar el formulario para la gestión de una norma de consumo

Tareas de la Ingeniería	
Número de tarea: 6	Numero de historia de usuario: 4
Nombre de la tarea: Validar campos del formulario para la gestión de una norma de consumo	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 01/03/2009	Fecha fin: 08/03/2009
Programador responsable: Liusmy Cárdenas Medina - Abel Savigne Suarez	
Descripción: Verificar que los datos que se insertan en los campos son los correctos	

Tabla 3. 13 Tarea 6 Validar campos del formulario para la gestión de una norma de consumo

Tareas de la Ingeniería	
Número de tarea: 7	Número de historia de usuario: 5
Nombre de la tarea: Diseñar el formulario para gestionar una materia prima.	
Tipo de tarea: Diseño	Puntos estimados: 1
Fecha inicio: 08/03/2009	Fecha fin: 15/03/2009
Programador responsable: Liusmy Cárdenas Medina - Abel Savigne Suarez	
Descripción: Crea un formulario que permitirá insertar, modificar y eliminar una materia prima.	

Tabla 3. 14 Tarea 7 Diseñar el formulario para gestionar una materia prima

Tareas de la Ingeniería	
Número de tarea: 8	Numero de historia de usuario: 5
Nombre de la tarea: Validar campos del formulario para la gestión de una materia prima	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 15/03/2009	Fecha fin: 22/03/2009
Programador responsable: Liusmy Cárdenas Medina - Abel Savigne Suarez	
Descripción: Verificar que los datos que se insertan en los campos son los correctos	

Tabla 3. 15 Tarea 8 Validar campos del formulario para la gestión de una materia prima

Tareas de la Ingeniería	
Número de tarea: 9	Numero de historia de usuario: 2
Nombre de la tarea: Diseñar el formulario para la gestión de usuario	
Tipo de tarea: Diseño	Puntos estimados: 1
Fecha inicio: 22/03/2009	Fecha fin: 29/03/2009
Programador responsable: Liusmy Cárdenas Medina - Abel Savigne Suarez	
Descripción: Se crean y habilitan todos los campos necesarios del formulario para la gestión de usuarios	

Tabla 3. 16 Tarea 9 Diseñar el formulario para la gestión de usuario

3.7.2 Iteración 2

En esta iteración se desarrollarán las historias de usuarios (HU) que contengan prioridad media, desplegándose en tareas; y también dándole continuidad a aquellas tareas de la primera iteración que aun no han concluido o que necesitan de perfeccionamiento.

Historias de usuario	Estimación	Real
Obtener indicadores específicos diarios	1	1
Obtener causas de tiempo perdido diarios	1	1
Mostrar planes de producción	1	1
Mostrar planes de entrega	1	1
Total	4	4

Tabla 3. 17 Historias de usuario desarrolladas en la segunda iteración

Tareas de la Ingeniería	
Número de tarea: 1	Numero de historia de usuario: 8
Nombre de la tarea: Diseñar la plantilla para obtener indicadores específicos	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 29/03/2009	Fecha fin: 05/04/2009

Programador responsable: Liusmy Cárdenas Medina - Abel Savigne Suarez
Descripción: Permite dado un centro de costo conocer el nombre del indicador que se quiere controlar y su unidad de medida.

Tabla 3. 18 Tarea 1 Diseñar la plantilla para obtener indicadores específicos

Tareas de la Ingeniería	
Número de tarea: 2	Numero de historia de usuario: 9
Nombre de la tarea: Diseñar la plantilla para obtener las causas de tiempo perdido	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 05/04/2009	Fecha fin: 12/04/2009
Programador responsable: Liusmy Cárdenas Medina - Abel Savigne Suarez	
Descripción: Seleccionando un centro de costo permite ver el tipo de tiempo perdido, ya sea total o parcial, en caso de que se parcial se mostrará el nombre del equipo afectado.	

Tabla 3. 19 Tarea 2 Diseñar la plantilla par obtener las causas de tiempo perdido

Tareas de la Ingeniería	
Número de tarea: 3	Numero de historia de usuario: 10
Nombre de la tarea: Diseñar la plantilla para mostrar los planes de producción	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 12/04/2009	Fecha fin: 19/04/2009
Programador responsable: Liusmy Cárdenas Medina - Abel Savigne Suarez	
Descripción: Permite ver los productos en producción listados en una tabla.	

Tabla 3. 20 Tarea 3 Diseñar la plantilla para mostrar los planes de producción

Tareas de la Ingeniería	
Número de tarea: 4	Numero de historia de usuario: 11
Nombre de la tarea: Diseñar la plantilla para ver los planes de entrega	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 19/04/2009	Fecha fin: 26/04/2009
Programador responsable: Liusmy Cárdenas Medina - Abel Savigne Suarez	
Descripción: Permite ver los productos terminados, mostrándolos en una lista	

Tabla 3. 21 Tarea 4 Diseñar la plantilla para ver los planes de entrega

3.7.3 Iteración 3

Para esta ultima iteración se desarrollarán las HU de prioridad baja no por esto dejan de ser importante pues son las que le dan culminación a las iteraciones del sistema, por lo que se tiene en esta iteración que realizar una rigurosa verificación de las iteraciones anteriores para no dejar ningún requisito incompleto.

Historias de usuario	Estimación	Real
Mostrar Tiempo Perdido Mensual	1	1
Mostrar Indicadores Específicos Mensual	1	1
Mostrar Causas de Ajustes	1	1
Mostrar Causas de Otras Operaciones	1	1
Total	4	4

Tabla 3. 22 Historias de usuario desarrolladas en la tercera iteración

Tareas de la Ingeniería	
Número de tarea: 1	Numero de historia de usuario: 12
Nombre de la tarea: Diseñar plantilla para mostrar el tiempo perdido del mes	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 26/04/2009	Fecha fin: 03/05/2009
Programador responsable: Liusmy Cárdenas Medina - Abel Savigne Suarez	

Descripción: Muestra las horas perdidas del mes
--

Tabla 3. 23 Tarea 1 Diseñar plantilla para mostrar el tiempo perdido del mes

Tareas de la Ingeniería	
Número de tarea: 2	Numero de historia de usuario: 13
Nombre de la tarea: Diseñar la plantilla para ver los indicadores específicos del mes	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 10/05/2009	Fecha fin: 17/05/2009
Programador responsable: Liusmy Cárdenas Medina - Abel Savigne Suarez	
Descripción: Crea un listado de las producciones con sus indicadores específicos.	

Tabla 3. 24 Tarea 2 Diseñar la plantilla para ver los indicadores específicos del mes

Tareas de la Ingeniería	
Número de tarea: 3	Numero de historia de usuario: 12
Nombre de la tarea: Diseñar la plantilla para mostrar las causas de ajustes	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 17/05/2009	Fecha fin: 24/05/2009
Programador responsable: Liusmy Cárdenas Medina - Abel Savigne Suarez	
Descripción: Crea un listado con las causas de ajustes	

Tabla 3. 25 Tarea 3 Diseñar plantilla para mostrar las causas de ajustes

Tareas de la Ingeniería	
Número de tarea: 4	Numero de historia de usuario: 13
Nombre de la tarea: Diseñar la plantilla para mostrar las causas de otras operaciones	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 24/05/2009	Fecha fin: 30/05/2009

Programador responsable: Liusmy Cárdenas Medina - Abel Savigne Suarez
Descripción: Crea un listado con las causas de otras operaciones

Tabla 3. 26 Tarea 4 Diseñar plantilla para mostrar las causas de otras operaciones

3.8 Pruebas unitarias

Las pruebas unitarias según la metodología XP se utilizan para verificar el funcionamiento del sistema; como la programación es en pareja, mientras uno va programando pues el otro va buscando errores e ideando posibles mejoras.

El framework Symfony tiene incluido su propio framework de prueba llamado Lime, el cual proporciona soporte para las pruebas unitarias, es más eficiente que otros frameworks de pruebas de PHP y tiene las siguientes ventajas:(16)

- Ejecuta los archivos de prueba en un entorno independiente para evitar interferencias entre las diferentes pruebas. No todos los frameworks 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 y su “regression testing”, 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.

3.9 Pruebas de aceptación

Las pruebas de aceptación no son más que validaciones que se le realizan al sistema para ver si cumple con el funcionamiento esperado y le permite al usuario determinar su aceptación en cuanto a su funcionalidad y rendimiento. Estas pruebas son definidas por el usuario del sistema y preparadas por el equipo de desarrollo.

Las prueba de aceptación son pruebas de caja negra que se realizan partiendo de la historias de usuario, estas pueden tener varias pruebas hasta que quede asegurado la aceptación del sistema. Un sistema está completamente aceptable cuando se satisfacen todos los requisitos funcionales especificados por el cliente teniendo en cuenta también los requisitos no funcionales relacionados con el rendimiento, seguridad de acceso al sistema, a los datos y procesos, así como a los distintos recursos del sistema.

Caso de Prueba de Aceptación	
Código: HU1_P1	Historia de Usuario: 1
Nombre de la persona que realiza la prueba: Liusmy Cárdenas Medina	
Descripción: Autenticarse en el sistema	
Condiciones de Ejecución: El usuario a autenticarse debe estar creado en el sistema con sus pertinentes autorizaciones	
Entrada/ Pasos de ejecución: Se debe llenar los campos de usuario y contraseña	
Resultado Esperado: El usuario es autenticado correctamente	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 3. 27 Prueba de aceptación Autenticarse en el sistema

Caso de Prueba de Aceptación	
Código: HU2_P1	Historia de Usuario:
Nombre de la persona que realiza la prueba: Liusmy Cárdenas Medina	
Descripción: Insertar un usuario en el sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado y tener los permisos necesarios.	
Entrada/ Pasos de ejecución: Se debe llenar todos los campos vacios para agregar un usuario.	
Resultado Esperado: El usuario es agregado en el sistema	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 3. 28 Prueba de aceptación Insertar usuario en el sistema

Caso de Prueba de Aceptación	
Código: HU2_P2	Historia de Usuario: 2

Nombre de la persona que realiza la prueba: Liusmy Cárdenas Medina
Descripción: Eliminar usuario del sistema.
Condiciones de Ejecución: El usuario debe estar autenticado y tener los permisos necesarios. 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 3. 29 Prueba de aceptación Eliminar usuario del sistema

Caso de Prueba de Aceptación	
Código: HU3_P1	Historia de Usuario: 3
Nombre de la persona que realiza la prueba: Liusmy Cárdenas Medina	
Descripción: Insertar un producto en el sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado y tener los permisos necesarios.	
Entrada/ Pasos de ejecución: Se debe llenar todos los campos vacios para la inserción de un producto.	
Resultado Esperado: El producto es insertado en el sistema	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 3. 30 Prueba de aceptación Insertar producto en el sistema

Caso de Prueba de Aceptación	
Código: HU3_P2	Historia de Usuario: 3
Nombre de la persona que realiza la prueba: Liusmy Cárdenas Medina	
Descripción: Modificar un producto en el sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado y tener los permisos necesarios. El producto ha modificar debe estar insertado	
Entrada/ Pasos de ejecución: Se busca el producto a modificar y se procede a la modificación	
Resultado Esperado: El producto es modificado	

Evaluación de la Prueba: Satisfactoria.

Tabla 3. 31 Prueba de aceptación Modificar producto en el sistema

Caso de Prueba de Aceptación	
Código: HU3_P3	Historia de Usuario: 3
Nombre de la persona que realiza la prueba: Liusmy Cárdenas Medina	
Descripción: Eliminar un producto en el sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado y tener los permisos necesarios. Debe de existir el producto a eliminar	
Entrada/ Pasos de ejecución: Se busca el producto a eliminar y se procede a la eliminación	
Resultado Esperado: El producto es eliminado	
Evaluación de la Prueba: Satisfactoria.	

Tabla 3. 32 Prueba de aceptación Eliminar producto del sistema

Caso de Prueba de Aceptación	
Código: HU4_P1	Historia de Usuario: 4
Nombre de la persona que realiza la prueba: Liusmy Cárdenas Medina	
Descripción: Insertar una norma de consumo en el sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado y tener los permisos necesarios.	
Entrada/ Pasos de ejecución: Se debe llenar los campos vacíos para insertar una norma de consumo	
Resultado Esperado: La norma de consumo es insertada.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 3. 33 Prueba de aceptación Insertar norma de consumo en el sistema

Caso de Prueba de Aceptación	
Código: HU4_P2	Historia de Usuario: 4
Nombre de la persona que realiza la prueba: Liusmy Cárdenas Medina	
Descripción: Modificar una norma de consumo en el sistema.	

Condiciones de Ejecución: El usuario debe estar autenticado y tener los permisos necesarios. Debe existir un producto con su norma
Entrada/ Pasos de ejecución: Buscar el producto. Cambiar la norma existente por la nueva
Resultado Esperado: La norma de consumo es modificada
Evaluación de la Prueba: Satisfactoria.

Tabla 3. 34 Prueba de aceptación Modificar norma de consumo en el sistema

Caso de Prueba de Aceptación	
Código: HU4_P3	Historia de Usuario: 4
Nombre de la persona que realiza la prueba: Liusmy Cárdenas Medina	
Descripción: Eliminar una norma de consumo en el sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado y tener los permisos necesarios. Debe existir un producto con su norma	
Entrada/ Pasos de ejecución: Buscar la norma de consumo. Eliminar la norma de consumo	
Resultado Esperado: La norma de consumo es eliminada	
Evaluación de la Prueba: Satisfactoria.	

Tabla 3. 35 Prueba de aceptación Eliminar norma de consumo en el sistema

Caso de Prueba de Aceptación	
Código: HU5_P1	Historia de Usuario: 5
Nombre de la persona que realiza la prueba: Liusmy Cárdenas Medina	
Descripción: Insertar una materia prima	
Condiciones de Ejecución: El usuario debe estar autenticado y tener los permisos necesarios.	
Entrada/ Pasos de ejecución: Se debe llenar los campos vacíos para añadir una materia prima.	
Resultado Esperado: La materia prima es añadida al sistema	
Evaluación de la Prueba: Satisfactoria.	

Tabla 3. 36 Prueba de aceptación Insertar materia prima en el sistema

Caso de Prueba de Aceptación

Código: HU5_P2	Historia de Usuario: 5
Nombre de la persona que realiza la prueba: Liusmy Cárdenas Medina	
Descripción: Modificar una materia prima	
Condiciones de Ejecución: El usuario debe estar autenticado y tener los permisos necesarios. El producto debe de existir con su materia prima	
Entrada/ Pasos de ejecución: Buscar materia prima del producto	
Resultado Esperado: La materia prima es modificada	
Evaluación de la Prueba: Satisfactoria.	

Tabla 3. 37 Prueba de aceptación Modificar materia prima

Caso de Prueba de Aceptación	
Código: HU5_P3	Historia de Usuario: 5
Nombre de la persona que realiza la prueba: Liusmy Cárdenas Medina	
Descripción: Eliminar una materia prima	
Condiciones de Ejecución: El usuario debe estar autenticado y tener los permisos necesarios.	
Entrada/ Pasos de ejecución: Buscar si la materia prima existe y eliminar	
Resultado Esperado: La materia prima es eliminada	
Evaluación de la Prueba: Satisfactoria.	

Tabla 3. 38 Prueba de aceptación Eliminar materia prima

Caso de Prueba de Aceptación	
Código: HU6_P1	Historia de Usuario: 6
Nombre de la persona que realiza la prueba: Liusmy Cárdenas Medina	
Descripción: Ingresar indicadores específicos diarios	
Condiciones de Ejecución: El usuario debe estar autenticado y tener los permisos necesarios. Debe existir al menos un producto.	
Entrada/ Pasos de ejecución: En el menú Entrada de Datos , clic en la opción Indicadores específicos y entrar los datos solicitados	
Resultado Esperado: Los indicadores son agregados	

Evaluación de la Prueba: Satisfactoria.

Tabla 3. 39 Prueba de aceptación Ingresar indicadores específicos

Caso de Prueba de Aceptación	
Código: HU7_P1	Historia de Usuario: 7
Nombre de la persona que realiza la prueba: Liusmy Cárdenas Medina	
Descripción: Ingresar tiempo perdido diarios	
Condiciones de Ejecución: El usuario debe estar autenticado y tener los premisos necesarios. Debe existir al menos un producto.	
Entrada/ Pasos de ejecución: En el menú Entrada de Datos , clic en la opción Tiempo perdido y entrar los datos solicitados	
Resultado Esperado: El tiempo perdido es agregado	
Evaluación de la Prueba: Satisfactoria.	

Tabla 3. 40 Prueba de aceptación Ingresar tiempo perdidos diarios

Caso de Prueba de Aceptación	
Código: HU8_P1	Historia de Usuario: 8
Nombre de la persona que realiza la prueba: Liusmy Cárdenas Medina	
Descripción: Mostrar planes de producción	
Condiciones de Ejecución: El usuario debe estar autenticado.	
Entrada/ Pasos de ejecución: En el menú Datos Base , clic en la opción Planes de producción, seleccionar Producción	
Resultado Esperado: Se muestran los planes de producción	
Evaluación de la Prueba: Satisfactoria.	

Tabla 3. 41 Prueba de aceptación Mostrar planes de producción

Caso de Prueba de Aceptación	
Código: HU9_P1	Historia de Usuario: 9
Nombre de la persona que realiza la prueba: Liusmy Cárdenas Medina	
Descripción: Mostrar planes de entrega	
Condiciones de Ejecución: El usuario debe estar autenticado.	

Entrada/ Pasos de ejecución: En el menú Datos Base, clic en la opción Planes de producción, seleccionar Entrega.
Resultado Esperado:
Evaluación de la Prueba: Satisfactoria.

Tabla 3. 42 Prueba de aceptación Mostrar planes de entrega

Caso de Prueba de Aceptación	
Código: HU10_P1	Historia de Usuario: 10
Nombre de la persona que realiza la prueba: Liusmy Cárdenas Medina	
Descripción: Reporte Tiempo perdido mensual	
Condiciones de Ejecución: El usuario debe estar autenticado.	
Entrada/ Pasos de ejecución: En el menú Reportes , clic en Tiempo perdido	
Resultado Esperado: Muestra un reporte de las horas y las causas del tiempo perdido en el mes	
Evaluación de la Prueba: Satisfactoria.	

Tabla 3. 43 Prueba de aceptación Mostrar Tiempo perdido mensual

Caso de Prueba de Aceptación	
Código: HU11_P1	Historia de Usuario: 11
Nombre de la persona que realiza la prueba: Liusmy Cárdenas Medina	
Descripción: Reporte Indicadores específicos mensual	
Condiciones de Ejecución: El usuario debe estar autenticado.	
Entrada/ Pasos de ejecución: En el menú Reportes , clic en Indicadores específicos	
Resultado Esperado: Muestra los valores acumulados en el mes y la unidad de medida que represente un producto	
Evaluación de la Prueba: Satisfactoria.	

Tabla 3. 44 Prueba de aceptación Mostrar Indicadores específicos

Caso de Prueba de Aceptación	
Código: HU12_P1	Historia de Usuario: 12
Nombre de la persona que realiza la prueba: Liusmy Cárdenas Medina	
Descripción: Mostrar Causas de ajustes.	
Condiciones de Ejecución: El usuario debe estar autenticado	
Entrada/ Pasos de ejecución: En el menú Datos Base, clic en Causa de ajustes	
Resultado Esperado: Se muestran las causas de ajustes	
Evaluación de la Prueba: Satisfactoria.	

Tabla 3. 45 Prueba de aceptación Mostrar Causas de ajustes

Caso de Prueba de Aceptación	
Código: HU13_P1	Historia de Usuario: 13
Nombre de la persona que realiza la prueba: Liusmy Cárdenas Medina	
Descripción: Mostrar Causas de ajustes.	
Condiciones de Ejecución: El usuario debe estar autenticado	
Entrada/ Pasos de ejecución: En el menú Datos Base, clic en Causa de Otras Operaciones	
Resultado Esperado: Se muestran las causas de las operaciones que son permitidas pero no están en el desarrollo de normal del proceso productivo	
Evaluación de la Prueba: Satisfactoria.	

3.10 Conclusiones

En este capítulo quedó terminada la aplicación con la implementación y las pruebas de las tareas propuestas logrando un software seguro y confiable. Demostró que son necesarias las pruebas para la corrección de funcionalidades y satisfacción del cliente.

Conclusiones Generales

Con la realización del trabajo se mostró la necesidad de un sistema capaz de controlar los datos primarios de la producción en la empresa Cuba Ron, permitiendo resolver los objetivos planteados. Luego de estudiar las tendencias de los diferentes sistemas contables nacionales e internacionales se pudo deducir la importancia de tener el control de los datos primarios que se utilizan en el proceso de contabilización de costo de la empresa Cuba Ron.

A partir del intercambio con los trabajadores de la empresa Cuba Ron se logró identificar los requisitos funcionales y no funcionales del sistema, cumpliendo de esa forma con las necesidades del cliente los cuales quedaron satisfechos con la solución propuesta. La realización de pruebas al sistema fue clave fundamental del éxito.

El desarrollo del trabajo utilizando la metodología XP permitió documentar el mismo desde el comienzo, lo que facilitará su estudio por parte de futuros miembros del equipo de desarrollo, permitiendo de esta forma, una comprensión más rápida y fácil de la concepción general del sistema. La versión del sistema obtenida, constituye una base para el trabajo del equipo, así como para futuras versiones del sistema.

Recomendaciones

Una vez concluido el desarrollo de este trabajo se recomienda para la mejora del sistema:

- Entrenar al personal que trabajará en la aplicación.
- Planificar una futura iteración para agregar funcionalidades al sistema en caso que surjan nuevos requisitos.
- Continuar con la realización de nuevos módulos que permitan el completamiento de un sistema contable en general.
- Realizar una ayuda online que sirva de apoyo al personal que trabaja en la aplicación.

Referencias bibliográficas

1. **Rolle, Hans y Zietzke, Julian.** Concepto de contabilidad. *Limitaciones del sistema contable*. [En línea] [Citado el: 2 de Febrero de 2009.] <http://www.monografias.com/trabajos10/sistcont/sistcont.shtml#concepto> .
2. **Feliz Álvarez, Isabel Cristina.** Definiciones de sistema de contabilidad. *Sistema de contabilidad*. [En línea] [Citado el: 15 de Diciembre de 2008.] <http://www.monografias.com/trabajos2/sistdecont/sistdecont.shtml>.
3. **George.** Concepto de contabilidad de costos. *Contabilidad de costos 1, Naturaleza de la contabilidad de costos*. [En línea] [Citado el: 2 de Febrero de 2009.] <http://www.monografias.com/trabajos39/contabilidad-de-costos/contabilidad-de-costos.shtml?monosearch>.
4. **AgroWin** sistema de gestión total para el agro. [En línea] [Citado el: 3 de Febrero de 2009.] <http://www.insoftweb.com/agrowin/default.htm>.
5. **Contapyme** sistema de gestión empresarial y contable para pymes. [En línea] [Citado el: 3 de Febrero de 2009.] <http://www.insoftweb.com/contapyme/default.htm> .
6. **Cerezal Tamargo, Lourdes.** La contabilidad en una nueva tecnología. *La revista del empresario cubano*. [En línea] [Citado el: 20 de Enero de 2009.] http://www.betsime.disaic.cu/secciones/tec_feb_02.htm.
7. **Zona PHP.** [En línea] [Citado el: 2 de Febrero de 2009.] <http://www.uterra.com/zonaPHP/>.
8. **de la Torre, Aníbal.** Lenguaje del lado servidor. *Lenguaje del lado servidor o cliente*. [En línea] 2006. [Citado el: 6 de Mayo de 2009.] http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html.
9. **Sánchez Mendoza, Maria A.** Metodologías de desarrollo de software. *Informatizate*. [En línea] 7 de Junio de 2004. [Citado el: 10 de Marzo de 2009.] http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.
10. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El proceso unificado de desarrollo de software*. Madrid : PEARSON S.A, 2000. ISBN: 84-7829-036-2 .
11. **Qué es SCRUM.** [En línea] [Citado el: 18 de Mayo de 2009.] <http://www.proyectosagiles.org/que-es-scrum>.
12. **Dormido, Sebastián.** Introducción al Modelado y Simulación de Sistemas Técnicos y Físico con Modelica. [En línea] julio de 2006. [Citado el: 19 de Mayo de 2009.] <http://www.addlink.es/productos.asp?pid=673>.

13. **Murillo Alfaro, Félix.** Lenguaje de modelado de realidad virtual(VRLM). *Realidad virtual*. [En línea] [Citado el: 19 de Mayo de 2009.] <http://www.inei.gob.pe/biblioineipub/bancopub/inf/Lib5047/PAG15.HTM>.
14. Enterprise Architect. *Sparxsystems*. [En línea] 2008. [Citado el: 19 de Mayo de 2009.] <http://www.sparxsystems.com.ar/products/ea/features.html>.
15. **Gutierrez, Javier J.** ¿Que es un framework web? [En línea] [Citado el: 3 de Febrero de 2009.] http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf.
16. **Potencier, Fabien y Zaninotto, Francois.** Symfony, la guía definitiva. *librosweb.com*. [En línea] 20 de Febrero de 2008. [Citado el: 3 de Febrero de 2009.] http://www.librosweb.es/symfony_1_0/capitulo1/symfony_en_pocas_palabras.html.
17. **Leopoldo, Carlos.** Zend framework, una introducción. [En línea] 27 de Noviembre de 2007. [Citado el: 19 de Mayo de 2009.] <http://www.carlosleopoldo.com/post/zend-framework-una-introduccion/>.
18. —. CakePHP. [En línea] 24 de Mayo de 2007. [Citado el: 19 de Mayo de 2009.] <http://www.carlosleopoldo.com/post/cakephp/>.

Bibliografía

1. Beck, Kent y Andres, Cynthia. 2004. Extreme Programming Explained: Embrace Change. 2da edición. . s.l.: Addison-Wesley, 2004.
2. Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2000. El Proceso Unificado de Desarrollo de Software. Madrid: s.n., 2000
3. MOLPECERES, A. Procesos de desarrollo: RUP, XP y FDD. 2003, Disponible en: http://www.javahispano.org/contenidos/es/procesos_de_desarrollo/.
4. POSTGRESQL. Advantages. Disponible en: <http://www.postgresql.org/about/advantages>.
5. POSTGRESQL, E. E. D. D. D. Tutorial de PostgreSQL. Disponible en: <http://es.tldp.org/Postgresql-es/web/navegable/tutorial/tutorial.html>.
6. POTENCIER, F. Z. Y. F. Symfony, la guía definitiva. 2008, Disponible en: <http://www.librosweb.es/symfony/>.
7. PRESSMAN, R. S. Ingeniería del Software, un enfoque práctico. Quinta ed. Editorial Félix Varela, 2002.
8. SANCHEZ, M. A. M. Metodologías De Desarrollo De Software. 2004, Disponible en: http://www.informatizate.net/articulos/pdfs/metodologias_de_desarrollo_de_software_07062004.pdf
9. XP. A gentle introduction. [En línea] 2006, Disponible en: <http://www.extremeprogramming.org>.

Glosario de Términos

1. **ASP** (Active Server Pages): Páginas Active Server. Entorno para crear y ejecutar aplicaciones dinámicas e interactivas en la Web. Se puede combinar páginas HTML, secuencias de comandos y componentes ActiveX para crear páginas y aplicaciones Web interactivas.
2. **Bases de datos**: Es un conjunto integrado de datos junto con una serie de aplicaciones para su manejo accesibles simultáneamente por diferentes usuarios y programas.
3. **CASE**: Acrónimo de Computer Aided Software Engineering.
4. **Herramientas CASE**: Herramientas utilizadas para el desarrollo de proyectos de Ingeniería de Software.
5. **HTML (HyperText Markup Language)**: Lenguaje de Marcado de Hipertexto. Lenguaje en el que se escriben las páginas a las que se accede a través de navegadores WWW. Admite componentes hipertextuales y multimedia
6. **Lenguaje**: En informática, cuando hablamos de lenguaje nos referimos generalmente al de programación, conjunto de instrucciones que las aplicaciones necesitan para que el ordenador ejecute determinadas operaciones. Hay lenguaje de alto y bajo nivel, de tercera y cuarta generación, lenguaje natural y lenguaje máquina, etc.
7. **Linux**: 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. La mayoría de los programas que ruedan bajo Linux son freeware, y muchos de ellos del Proyecto GNU. Linux tiene todas las características que se pueden esperar de un moderno y flexible UNIX. Incluye multitarea real, memoria virtual, librerías compartidas, dirección y manejo propio de memoria y TCP/IP. Usa las características hardware de la familia de procesadores 386 para implementar las capacidades anteriores.
8. **Open Source (Código Abierto)**: Open Source es un término que describe partes de la licencia del movimiento por el software libre.

9. **Perl**: *Practical Extraction and Report Language*. Es un lenguaje de programación desarrollado por Larry Wall inspirado en otras herramientas de UNIX como son: sed, grep, awk, c-shell.
10. **PHP**: Lenguaje procesador de Scripts servidor. Libre y de código abierto.
11. **Servidor**: Sistema que proporciona recursos (por ejemplo, servidores de ficheros, servidores de nombres). En Internet este término se utiliza muy a menudo para designar a aquellos sistemas que proporcionan información a los usuarios de la red.
12. **SQL - Structured Query Language**: 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.
13. **UNIX**: Sistema operativo portable, flexible, potente, con entorno programable, multiusuario y multitarea, muy difundido
14. **World Wide Web**: Telaraña mundial, Red mundial, WWW. Sistema global de la información basado en la tecnología del hipertexto. Soporta todo tipo de información (audio, video, imagen, texto, etc.) y se accede de manera fácil por los usuarios a través de los navegadores.

Anexo 1 Diagramas de clases del diseño

Diagrama de clases del diseño_ Gestionar Productos

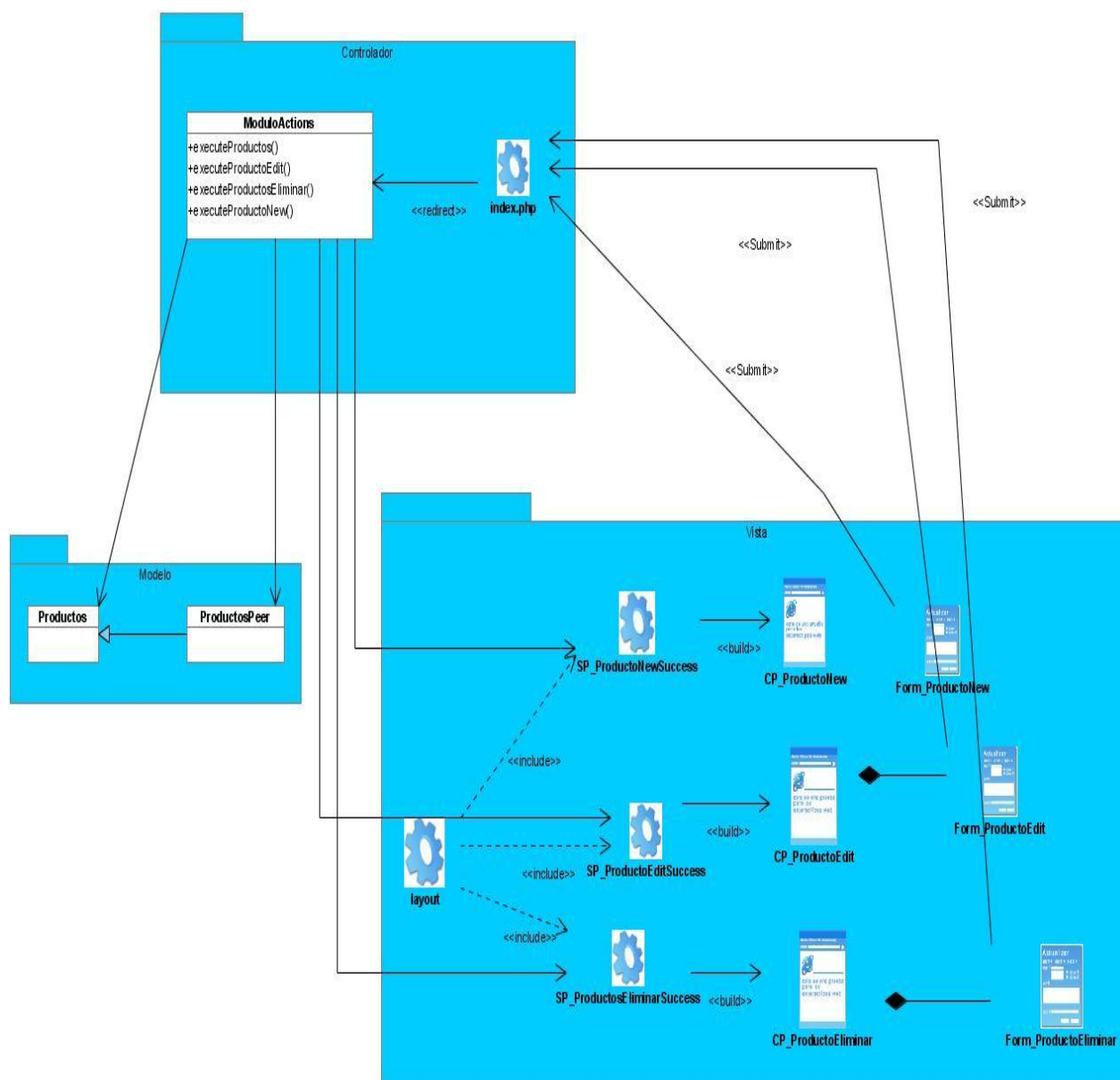


Diagrama de clases del diseño_ Mostrar Productos

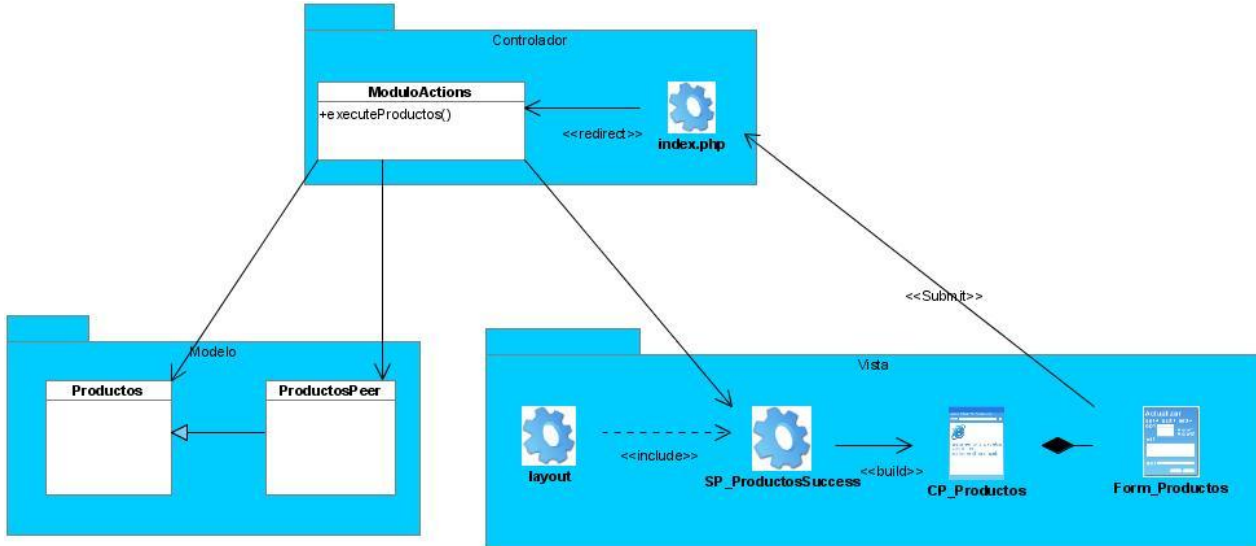


Diagrama de clases del diseño_ Gestionar Materias primas

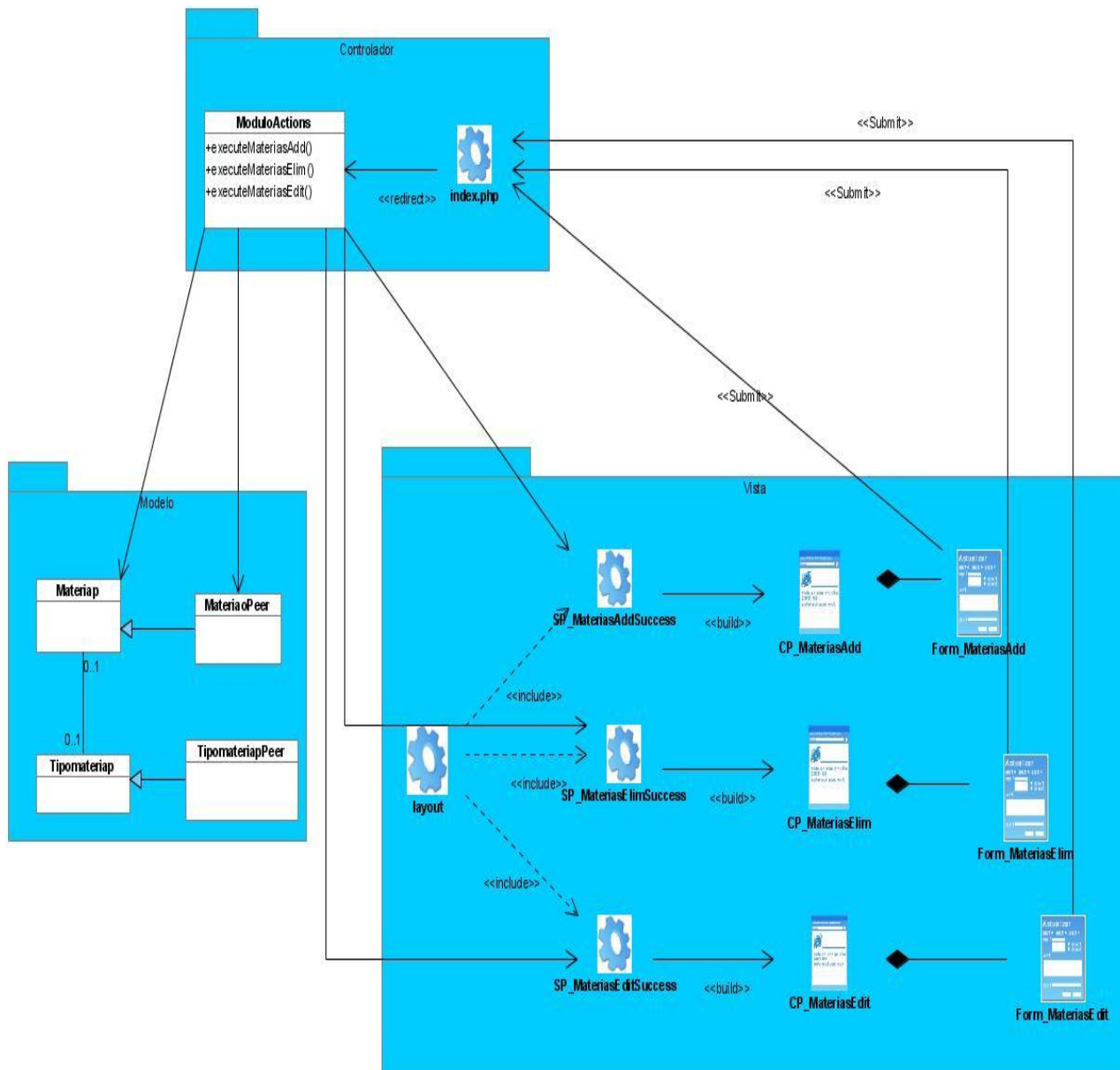


Diagrama de clases del diseño_Gestionar normas de consumo

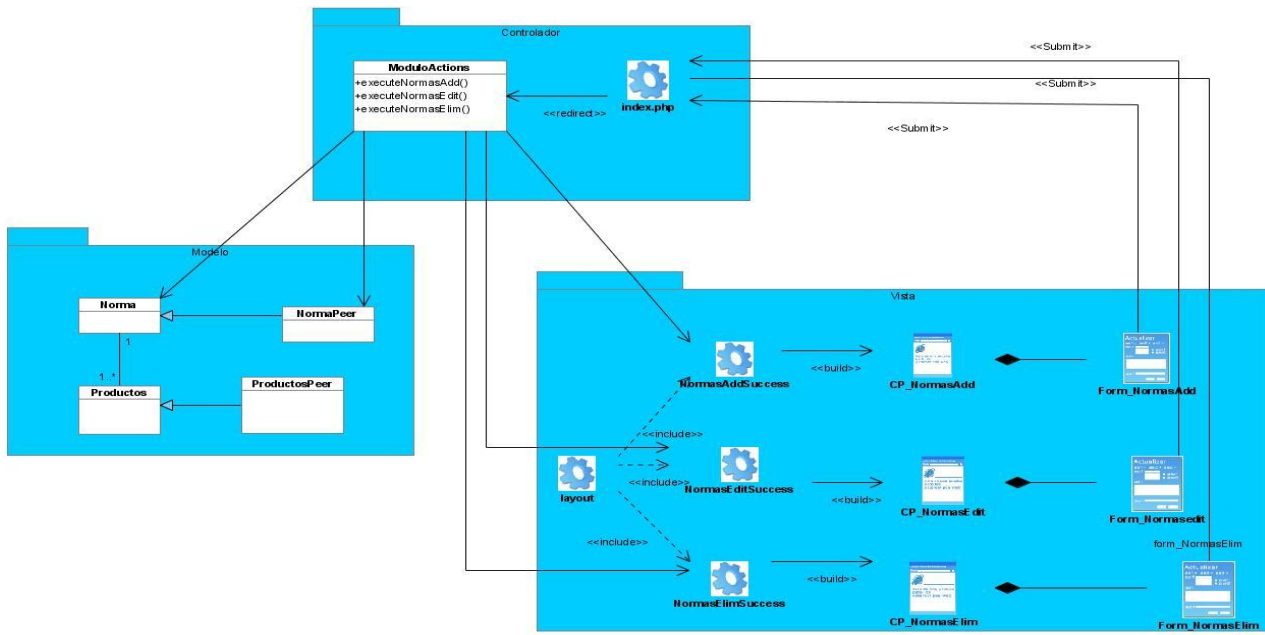
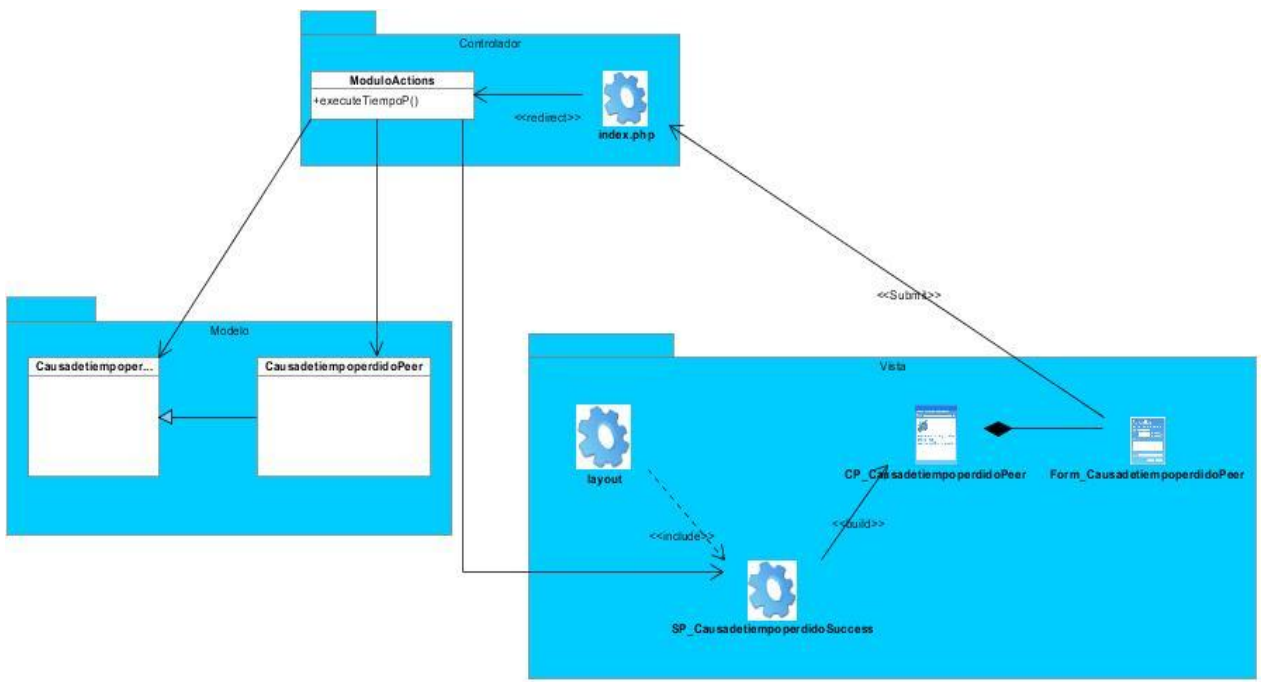


Diagrama de clases del diseño_Mostrar Tiempo Perdido

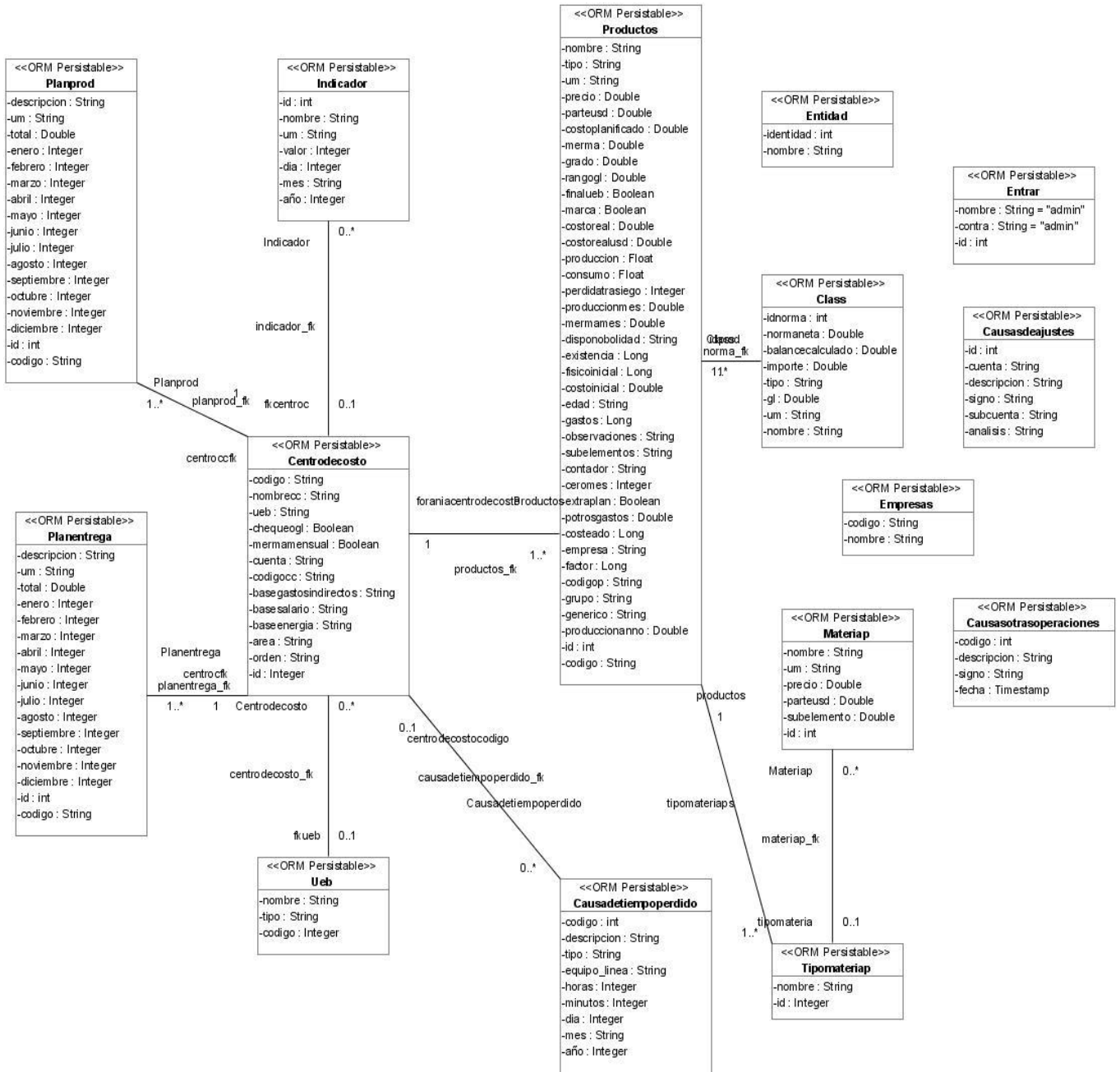


Anexo 2 Modelo de Base de datos

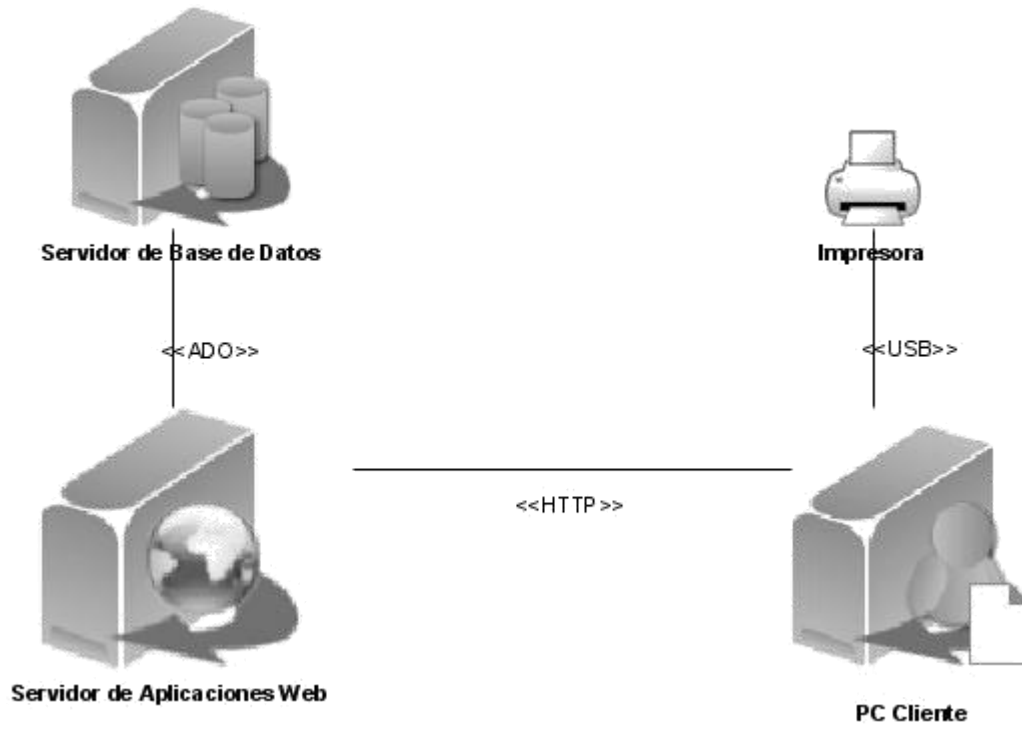
Diagrama entidad-relación



Diagrama de clases Persistentes



Anexo 3 Diagrama de despliegue



Anexo 4 Agile Manifesto

En febrero de 2001, en las montañas Wasatch de Utah, se reunieron 17 desarrolladores convencidos de que era necesario un cambio en las metodologías “clásicas” de desarrollo de software. Entre ellos se encontraban los creadores de XP, SCRUM, DSDM, Adaptive Software Development, Crystal, Feature-Driven Development y Pragmatic Programming. Juntos proclamaron lo que se ha dado a conocer como el “Manifiesto for Agile Software Development”, estableciendo en él cuatro principios [5]:

“Se valora a los individuos y las interacciones sobre los procesos y las herramientas

Se valora a las aplicaciones que funcionan sobre la documentación exhaustiva

Se valora la colaboración del cliente sobre las negociaciones contractuales

Se valora la respuesta al cambio sobre el seguimiento de un plan

Esto significa que, sin desconocer el valor de los segundos ítems, se valoran más los primeros”

Este manifiesto se basa en los siguientes principios:

- Satisfacer al cliente a través de entregas continuas y tempranas es la mayor prioridad.
- Los cambios a los requerimientos son bienvenidos, aún en fases tardías del desarrollo.
- Entregar frecuentemente software que funciona, desde un par de semanas a un par de meses, prefiriendo los periodos más cortos.
- Desarrolladores, gerentes y clientes deben trabajar juntos diariamente, a lo largo del proyecto.
- Construir proyectos alrededor de personas motivadas, dándoles el entorno y soporte que necesitan, y confiando en que realizarán el trabajo.
- El método más eficiente y efectivo de transmitir información entre un equipo de desarrolladores es la conversación frontal (cara a cara).
- Tener software que funciona es la medida primaria del progreso.
- El proceso ágil promueve el desarrollo sostenible. Los sponsors, desarrolladores y usuarios deben ser capaces de mantener un ritmo de trabajo constante en forma permanente a lo largo del proyecto.
- La atención continua a la excelencia técnica y el buen diseño mejoran la agilidad.
- Simplicidad – el arte de maximizar el trabajo que no se debe hacer – es esencial.
- Las mejores arquitecturas, requerimientos y diseños surgen de los equipos auto-organizados.

- A intervalos regulares, el equipo debe reflexionar sobre como ser más efectivos, y ajustar su comportamiento de acuerdo a ello.

Los desarrollos de software ágil, que adoptan los principios del “Agile Manifesto”, no son anti-metodológicos. Por el contrario, siguen su metodología, diferente a la de los métodos clásicos de desarrollo. Se trata de lograr un equilibrio, en el que, por ejemplo, la documentación es concreta y útil, y no burocrática y los planes existen, pero reconociendo sus limitaciones en el actual mundo en permanente cambio.

Valores en XP

XP se basa en cuatro valores, que deben estar presentes en el equipo de desarrollo para que el proyecto tenga éxito

Comunicación

Muchos de los problemas que existen en proyectos de software (así como en muchos otros ámbitos) se deben a problemas de comunicación entre las personas. La comunicación permanente es fundamental en XP. Dado que la documentación es escasa, el diálogo frontal, cara a cara, entre desarrolladores, gerentes y el cliente es el medio básico de comunicación. Una buena comunicación tiene que estar presente durante todo el proyecto.

Simplicidad

XP, como metodología ágil, apuesta a la sencillez, en su máxima expresión. Sencillez en el diseño, en el código, en los procesos, etc. La sencillez es esencial para que todos puedan entender el código, y se trata de mejorar mediante recodificaciones continuas.

Retroalimentación

La retroalimentación debe funcionar en forma permanente. El cliente debe brindar retroalimentación de las funciones desarrolladas, de manera de poder tomar sus comentarios para la próxima iteración, y para comprender, cada vez más, sus necesidades. Los resultados de las pruebas unitarias son también una retroalimentación permanente que tienen los desarrolladores acerca de la calidad de su trabajo.

Coraje

Cuando se encuentran problemas serios en el diseño, o en cualquier otro aspecto, se debe tener el coraje suficiente como para encarar su solución, sin importar que tan difícil sea. Si es necesario cambiar

completamente parte del código, hay que hacerlo, sin importar cuanto tiempo se ha invertido previamente en el mismo.

Anexo 5 Críticas a la metodología XP

Una de las críticas a XP es la dificultad de estimar cuánto va a costar un proyecto. Dado que el alcance del mismo no está completamente definido al comienzo, y que la metodología XP es expresamente abierta a los cambios durante todo el proceso, se torna sumamente difícil poder realizar un presupuesto previo. Para desarrollos “in house” este punto puede no ser crítico, pero sí lo es especialmente para empresas desarrolladoras de software, dónde deben presupuestar (y ganar) proyectos.

Por otro lado, muchas de las prácticas sugeridas por XP son compatibles y utilizadas, de una u otra forma, en muchas otras metodologías. Por ejemplo, tener un diseño simple, utilizar estándares y mantener un ritmo sostenido de trabajo es claramente ventajoso, sea cual sea la metodología utilizada. Sin embargo, otras prácticas pueden ser más discutibles. Pensar únicamente en el diseño de lo que se debe entregar inmediatamente, sin tener en cuenta lo que deberá realizarse inmediatamente después, o dentro de poco tiempo, no siempre es la mejor decisión. Es cierto que muchas veces, por intentar generalizar o prever futuros casos, se invierte tiempo extra para entregables que quizás no lo requerirían. Sin embargo, si el análisis fue correctamente realizado, este tiempo extra inicial redundará en un menor tiempo total. Claro que, si se piensa que siempre existirán cambios imprevistos, no conviene invertir tiempo en generalizar o prever futuras funciones. Es una práctica que puede admitir diversos puntos de vista.

Otra posible crítica a XP refiere a la baja documentación, pensando en el posterior mantenimiento del sistema. Si bien durante el proyecto el equipo tiene en mente todas sus particularidades, hay que prever que pasará luego de entregado, cuando el equipo se disuelva, y sea necesario realizar algún cambio o mejora. XP propone la recodificación permanente, para asegurar la claridad y simplicidad del código. Sin embargo, es posible que aún un código simple y claro no baste cuando otro equipo de trabajo tenga que tomar el sistema y cambiarlo. Es posible que sea necesario mantener cierta documentación, aunque es compatible que ésta debe ser la mínima necesaria.

En un artículo de Matt Stephens, se critica la “auto-realimentación” de XP, ya que sus prácticas requieren ser aplicadas en su totalidad para que el método sea viable: No disponer de requerimientos detallados escritos lleva a mantener un diseño simple. Un diseño simple lleva a que sea necesario recodificar constantemente. La recodificación necesita de permanentes pruebas unitarias. Las pruebas unitarias

fallarán poco, gracias a la programación de a pares, los que requieren del cliente en sitio para poder conocer los detalles a implementar. Finalmente, tener al cliente en sitio hace irrelevante disponer de requerimientos escritos detallados, cerrando el círculo. Según este artículo, una falla en cualquiera de estas prácticas hará que todo el proyecto fracase.