

Desarrollo de un Sistema Vía Web para Control de Producción en la Granja Avícola “Marco Antonio Vivanco Álvarez”, aplicando la metodología FDD con herramientas libres.

Mayra Alejandra Samaniego Pallaroso¹
Ing. César Villacis², Ing. Paul Díaz³

1 Escuela Politécnica del Ejército. Ecuador, mayra.samaniego@gmail.com

2 Escuela Politécnica del Ejército. Ecuador, cjvillacis@espe.edu.ec

3 Escuela Politécnica del Ejército. Ecuador, pmdiaz@espe.edu.ec

RESUMEN

El presente proyecto está orientado a desarrollar un sistema web que permita llevar el control de la producción en la cría de pollos de engorde, automatizando el flujo de procesos con la aplicación de la metodología de desarrollo ágil FDD (Feature Driven Development – Desarrollo Basado en Características) y utilizando herramientas de software libre.

El marco de trabajo de FDD, se centra más en obtener resultados de alta importancia para el cliente, que en la documentación excesiva de otras metodologías, en las que se corre el riesgo de no reflejar la verdadera necesidad con la que se construye el sistema, convirtiéndose en una metodología novedosa y fácil de implementar. Como manda FDD, el proceso de desarrollo, desde el levantamiento de requisitos, análisis, diseño e implementación del software, se lo realizó en iteraciones cortas, llevando el control de versiones de todos los artefactos generados, y estimulando la participación activa del usuario al entregarle resultados funcionales en tiempos cortos.

La aplicación práctica de FDD se la llevo a cabo desarrollando el sistema de control de producción para la granja avícola “Marco Antonio Vivanco Álvarez”, siguiendo todos los lineamientos y buenas prácticas que indica la metodología para cumplir con los objetivos requeridos por la granja y entregando el software a la medida de las necesidades del proceso de control de producción que emplean actualmente. El producto de software resultante es un sistema web distribuido, en lenguaje de programación JAVA, basado en el patrón de diseño Modelo Vista Controlador y empleando MySQL como motor de base de datos, además se utilizó el framework JUnit para realizar las pruebas de unidad de manera automatizada.

Palabras Clave: FDD, Característica, Distribuida.

ABSTRACT

This project aims to develop a web system that allows keeping track of production in raising broilers, by automating the process flow with the application of agile development methodology FDD (Feature Driven Development) and using free software tools.

The FDD framework focuses more on results of high importance for the customer, than excessive documentation from other methodologies in which there is a risk of not reflecting the true need for the system to be built.

The practical application of FDD was performed developing the production control system for the poultry farm "Marco Antonio Alvarez Vivanco", following all the guidelines and best practices that describes the methodology to meet the objectives required by the farm and delivering the software tailored to the needs of the production control process which is now used. The resulting software product is a distributed web system, Java programming language, based on the standard Model View Controller design and using MySQL as database engine, also used the JUnit framework for unit testing so tests would be automated.

KeyWords: FDD, Feature, Model.

1. INTRODUCCIÓN

La información y el conocimiento tecnológico se han convertido en fuente de progreso económico y de productividad; sin importar el área de trabajo en que se mire, cualquier persona que realice algún tipo de actividad económica está obligada cada vez más a utilizar la tecnología en sus negocios, desde un email hasta un sistema de control especializado (*ERP*) marcan la diferencia a la hora de producir, y hoy en día resultaría casi imposible imaginar alguna actividad económica en donde no intervengan las tecnologías de información.

Las pequeñas y medianas empresas avícolas en el Ecuador no son la excepción, este sector está invirtiendo en herramientas que faciliten el control de producción, y les permitan darse a conocer en el mercado. Siguiendo esta línea de avance tecnológico y para estar al mismo nivel de competitividad que sus similares, la granja avícola "Marco Antonio Vivanco Álvarez" ubicada en el cantón Chone de la provincia de Manabí, que genera un promedio de ventas de 11500 pollos mensuales, tiene la necesidad de implementar un software que permita manejar, discriminar la información relevante y llevar un control real de la producción, tomando en cuenta factores personalizados, que satisfagan las perspectivas del nivel administrativo y apoyen al nivel medio; por este motivo se hace indispensable la construcción de una herramienta informática, que permita estructurar un sistema de control de producción eficaz y eficiente.

El sistema web diseñado proporciona esa solución informática, utilizando métodos, técnicas de ingeniería de software, y herramientas libres que logren el producto en funcionalidad, oportunidad y costo.

2. METODOLOGÍA

FDD (*Feature Driven Development – Desarrollo Basado en Características*) es una metodología de desarrollo ágil que se basa en la definición y trabajo sobre características para entregar resultados funcionales al cliente. [1]

El procedimiento de FDD empieza con levantar una lista de características detalladas por el cliente, ordenándolas según su criticidad y agrupando las que son similares para formar las iteraciones que deben ser cumplidas en un tiempo no mayor a dos semanas, después de ello, se diseña un modelo global del sistema para dar una idea introductoria a todo el equipo de la dirección y las características con las que debe cumplir el producto final. [2]

Las prácticas que componen FDD son:

1. Desarrollo de un modelo global.
2. Construcción de una lista de características.
3. Planeación por característica.
4. Diseño por característica.

5. Construcción por característica
6. Pruebas por características. [3]

En la Figura 1 se muestra el Marco de Trabajo de la metodología FDD.

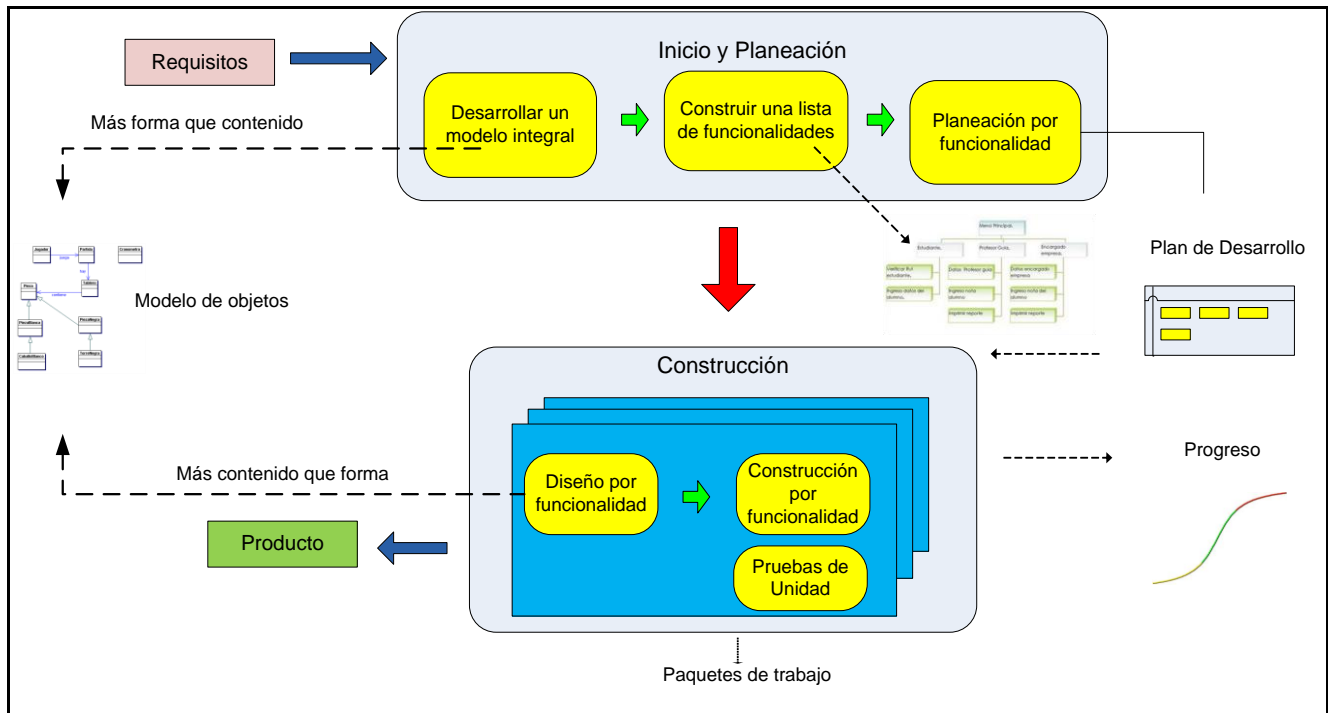


Figura 1 Marco de Trabajo de la Metodología FDD
 Fuente: http://www.koiosoft.com/en_us/wiki/agile-development

Durante el proceso de desarrollo se pudo evidenciar las ventajas que brinda FDD:

- El diseño de un modelo global al inicio del desarrollo elimina posibles confusiones entre los miembros de los grupos y ayuda a comprender de mejor manera y con más rapidez las características que deben ser implementadas.
- El hecho de no utilizar casos de uso, elimina el desperdicio de tiempo en documentar requerimientos que posiblemente cambien en la siguiente inspección del usuario.
- Las características son lo más pequeñas posible, de modo que puedan ser entregadas en máximo dos semanas.
- Las inspecciones continuas y la participación activa del cliente permite la detección de defectos y su corrección a tiempo.
- El cliente percibe resultados en tiempos cortos. [4]

3. HERRAMIENTAS

Para el desarrollo del presente proyecto se ha considerado la utilización de las siguientes herramientas de software libre:

3.1. Java Enterprise Edition

Java Platform Enterprise Edition o JEE, es una plataforma de desarrollo diseñada para construir aplicaciones empresariales multicapas basadas en componentes estandarizados, modulares y reutilizables, permitiendo así, manejar muchos aspectos de la programación de forma automática, logrando aplicaciones robustas que trabajen a través de internet. [5]

3.2. Framework JUnit

Framework de código abierto diseñado para realizar pruebas de unidad repetitivas en lenguaje de programación Java, de manera rápida y fácil. Permite a los desarrolladores construir gradualmente bancos de pruebas para medir el progreso y detectar efectos secundarios no deseados, obteniendo resultados inmediatos.

JUnit promueve la idea de: Testear – Codificar – Testear, de tal manera que aumenta la productividad del programador, la estabilidad del código del programa, reduce el estrés y el tiempo de depuración. [6]

3.3. Apache Tomcat

Servidor web y de aplicaciones, de código abierto, es mantenido y desarrollado por miembros de la Apache Software Foundation y voluntarios independientes. Gestiona solicitudes y es servidor de aplicaciones o contenedor de Servlets/JSP (*Java Server Pages*). [7]

3.4. Eclipse

Proyecto de desarrollo de software de código abierto, cuyo objetivo es la construcción de herramientas integradas para el desarrollo de aplicaciones. Brinda un entorno de desarrollo integrado multiplataforma que facilita la implementación de diferentes lenguajes de programación, y, proporciona apertura para el desarrollo y extensión de los sistemas. [8]

El SDK (*Software Development Kit*) de Eclipse incluye las herramientas de desarrollo de Java, ofreciendo un IDE con un compilador de Java interno y un modelo completo de los archivos fuente de Java.

3.5. Liferay

Plataforma web corporativa de código abierto, diseñada para desarrollar soluciones empresariales con resultados inmediatos y valor a largo plazo. [9] Incluye una amplia gama de funcionalidades, tales como:

- Diseño de interfaces de usuario de manera fácil y ágil.
- Framework de integración de aplicaciones.
- Construcción rápida de sitios.
- Creación de páginas con solo un clic.
- Publicación de contenidos basada en roles.
- Personalización de usuarios.

- Calendario compartido.
- Anuncios y alarmas, entre otros.

Liferay ayuda a las empresas a desarrollar soluciones sólidas que ofrecen los resultados deseados en múltiples ámbitos:

- Portales de autoservicio
- Espacios de trabajo para el intercambio de conocimiento
- Sitios Web 2.0 dinámicos
- Redes sociales que generan ingresos
- Integración con aplicaciones corporativas. [10]

3.6. Hibernate

Herramienta de Mapeo objeto-relacional para la plataforma Java, distribuida bajo los términos de la licencia GNU LGPL, que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos XML o anotaciones en los beans de las entidades. [11]

3.7. Adobe Flex

Herramienta de código abierto, muy productiva para la creación y el mantenimiento de aplicaciones web RIA (*Rich Interface Application*) que se implantan coherentemente en los principales exploradores, equipos de escritorio y sistemas operativos. [12]

3.8. My SQL

Sistema de gestión de bases de datos relacional, su diseño multihilo le permite soportar alta carga de forma muy eficiente.

MySQL brinda rendimiento, alta fiabilidad y facilidad de uso, permite ahorrar tiempo y dinero en grandes volúmenes de sitios Web, sistemas críticos de negocio y software empaquetado. Se ejecuta en más de 20 plataformas, incluyendo Linux, Windows, Mac OS, Solaris, AIX de IBM, que le da el tipo de flexibilidad con alto grado de control. [13]

3.9. Star UML

Herramienta case de código abierto, utilizada para el modelado de datos con UML, arquitectura, metadatos, etc., para el diseño y análisis de software de manera fácil y ágil. [14]

DISEÑO E IMPLEMENTACIÓN

El producto resultante es un sistema web distribuido en cuatro capas: presentación, servicios, negocio y datos; usando el lenguaje de programación JAVA, basado en el patrón de diseño Modelo Vista Controlador, incorporando Liferay como manejador de contenidos y empleando MySQL como motor de base de datos, además se utilizó el framework JUnit para realizar las pruebas de unidad de manera automatizada.

En la Figura 2 se muestra la Arquitectura Lógica del Sistema.

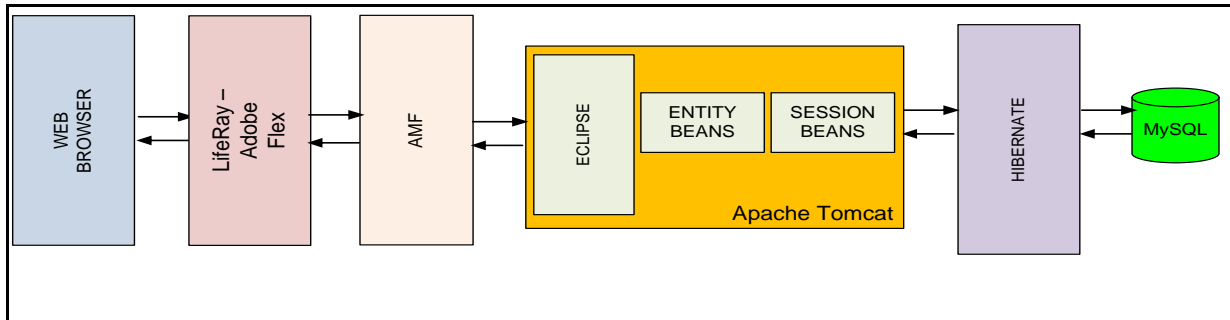


Figura 2 Arquitectura Lógica del Sistema
Fuente: Autor

En el diseño de la capa de presentación se utilizó Liferay que permitió consumir de forma muy sencilla los beans de java en forma de portlets (*componentes modulares*), además permitió realizar de manera automática el formulario de inicio de sesión.

En la Figura 3 se muestra la Pantalla de Inicio del Sistema, implementada con Liferay.

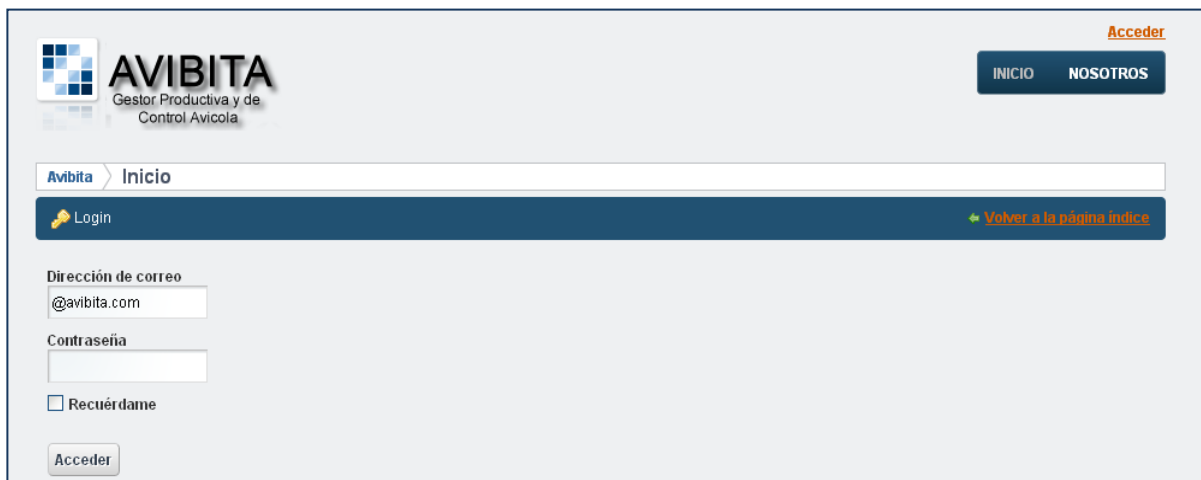


Figura 3 Pantalla de Inicio de Sesión del Sistema
Fuente: Autor

En la Figura 4 se muestra la Pantalla principal del Sistema, implementada con Liferay.

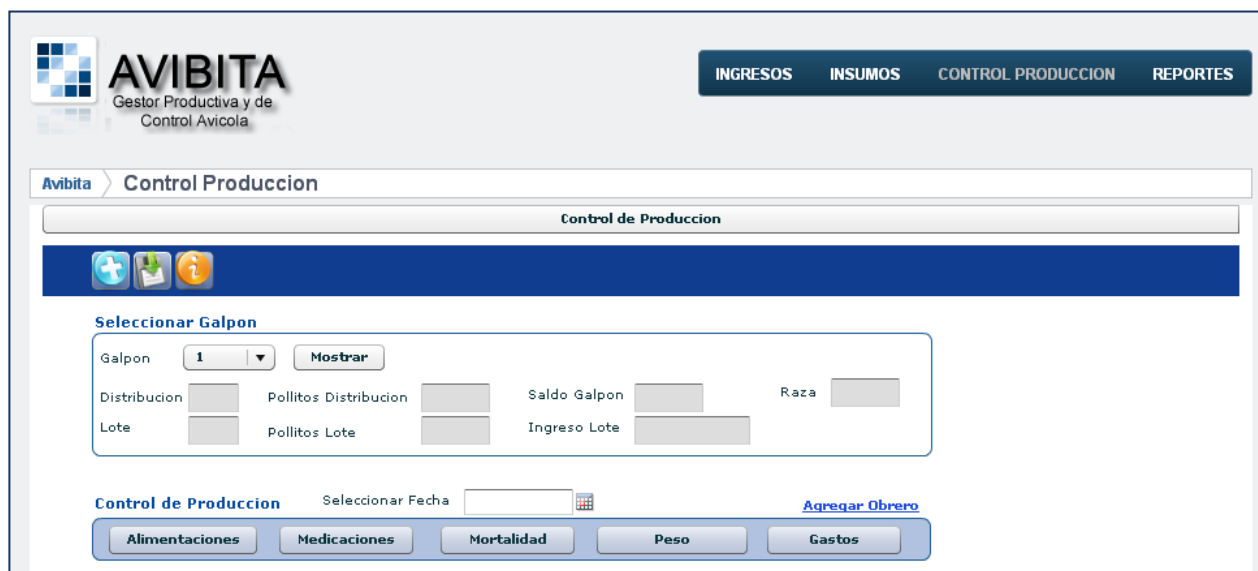


Figura 4 Pantalla Principal del Sistema
Fuente: Autor

El sistema fue desarrollado para cumplir los siguientes requerimientos:

- Administración de galpones, lotes, empleados y medicinas.
- Control de Producción de los diferentes parámetros zootécnicos generados durante el proceso de producción: índice de mortalidad, ganancia de peso promedio, control de vacunas, control alimenticio, medicaciones y gastos.
- Gestión de provisiones de alimento y medicinas.
- Reportes personalizados de acuerdo a los requerimientos del área administrativa.

RESULTADOS

El principal parámetro para analizar los resultados, fue el nivel de usabilidad del sistema en los últimos dos meses.

Se logró proveer de una herramienta fácil de manejar, a través de una interfaz intuitiva para el usuario, que brinde comodidad de navegación y entendimiento a personas con poca experiencia en el uso de ordenadores.

La realización del sistema en su totalidad con herramientas libres, permitió abaratar costos de implementación al evitar la compra de licencias y la inversión en infraestructura excedente; gracias a ello se logró proyectar un retorno de la inversión en un tiempo no mayor a un año.

El uso de reportes en línea permitió acceder de forma rápida a información actualizada del proceso de producción para la toma de decisiones.

TRABAJOS RELACIONADOS

Aunque ya se ofertan en el mercado sistemas para el control avícola, estos son muy generales y no se adaptan a las necesidades puntuales de las pequeñas y medianas empresas del sector que se dedican a áreas específicas de producción como: ponedoras, pollos de engorde, incubadoras, etc., quedando módulos subutilizados, de tal manera que la implementación de uno de estos sistemas no es considerado por los gerentes como una inversión, sino como un gasto.

En la Tabla II se muestran cinco de las características más relevantes que se encuentran en el sistema web desarrollado y que no están en la herramienta de control que se utilizaba anteriormente:

Tabla I Comparación entre las Herramientas usadas para el Control de Producción

Característica	Sistema Web Desarrollado	Herramienta utilizada anteriormente (Excel)
Control en la asignación de obreros para el trabajo de producción en los galpones.	x	
Distribución de alimento y medicinas	x	
Seguimiento de los parámetros de control de producción en tiempo real.	x	
Control de gastos	x	
Control y verificación de la disponibilidad de galpones e insumos.	x	

CONCLUSIONES Y TRABAJO FUTURO

Después de haber realizado el estudio de la metodología FDD, se pudo constatar que la misma facilita la etapa de diseño, al no exigir casos de uso sino una lista de características evaluadas por el cliente; esto gracias a que FDD le presta más importancia a la comunicación y participación activa con el usuario que a la documentación excesiva.

El resultado de una buena planificación en FDD se verá reflejado en el tiempo de implementación de cada iteración, mismo que no debe ser mayor a dos semanas, de no ser así, las características estarían refiriéndose a aspectos demasiado generales y deben ser corregidas.

El acoplarse a FDD no es una tarea difícil, sin embargo se necesita de un alto grado de disciplina para aplicar las buenas prácticas de desarrollo ágil, que en principio podrían demandar mayor esfuerzo y tiempo, como por ejemplo el llevar el control de versiones de todos los artefactos, pero es lo único que nos asegurará que al final podamos tener un producto eficiente y de calidad.

Al trabajar desarrollando el sistema en pequeñas iteraciones, se pudo entregar al cliente resultados funcionales en poco tiempo, dándole satisfacción y motivándolo a participar activamente para alcanzar el producto deseado.

Realizar pruebas antes de la etapa de desarrollo permitió tener mayor seguridad en la implementación del código, además se pudo optimizar el tiempo de depuración de errores gracias a la utilización del framework JUnit.

Las pruebas unitarias no deben ser realizadas con el fin de lograr la perfección del flujo de procesos, sino para detectar posibles errores que puedan afectar al resto de componentes.

Se desarrolló el sistema web empleando herramientas de software libre de alta calidad, reduciendo costos y haciendo que el tiempo de retorno de la inversión no sea mayor a un año. Dentro del software utilizado están: el motor de base de datos MySQL, la herramienta de desarrollo Eclipse, el kit de desarrollo de java JDK, el servidor web y de aplicaciones Tomcat y la herramienta de modelado Star UML.

Durante el tiempo de desarrollo del sistema, se pudo constatar el alto desempeño de las herramientas utilizadas, puesto que ellas proveyeron confiabilidad y eficiencia en cada reproceso y ajuste de las características.

Para el manejo de contenidos se utilizó la plataforma web Liferay, misma que permitió acoplar fácilmente la tecnología EJB de java, al consumir los beans en forma de portlets, además automatizó el formulario de ingreso al sistema.

La arquitectura lógica del sistema se la dividió en cuatro capas: Presentación, Servicios, Negocio y Datos; lo que permitió aislar la funcionalidad, del despliegue de los resultados finales al usuario, utilizando tecnologías diferentes en ambas: Java y Flex, respectivamente.

La división en capas brinda al programador una guía para hacer mejoras a la aplicación, sin que se torne en una tarea tediosa y desgastante, siguiendo el estándar establecido para tal fin y dividiendo las tareas en partes específicas para cada capa del proyecto.

Al realizar la programación del sistema en capas, el código se volvió mucho más claro y fácil de mantener, a medida que se avanzaba en el cumplimiento de las iteraciones.

Como trabajo futuro se pretende aplicar FDD para el desarrollo de software dentro del área contable de la granja.

RECOMENDACIONES

Para que la metodología FDD tenga pleno éxito en su implementación, se recomienda que las características evaluadas por el cliente se refieran a aspectos específicos del negocio, de tal manera que involucren la menor cantidad de clases posibles.

Para obtener software de calidad, se recomienda utilizar conjuntamente con FDD: patrones de diseño, semántica correcta e integración continua.

Se recomienda utilizar solo el mínimo código necesario que permita realizar las pruebas de verificación de cada característica, de lo contrario se perdería la objetividad funcional de las iteraciones y se estaría incurriendo en el error de tomar en cuenta aspectos que tal vez no estén dentro de las necesidades del cliente.

Para la implementación de FDD se recomienda contar con un líder de equipo con experiencia, debido a que esta metodología inicia con un modelo global que es la base para construir la lista de iteraciones, y se necesita de alguien que sirva de guía al resto del equipo de desarrollo.

Cuando se trabaje en desarrollos de gran dimensión, se recomienda dividirlos en módulos funcionales específicos, y trabajar aisladamente en cada uno con la lista de características y pruebas por iteración, para poder entregar el software a medida de la necesidad del cliente.

Se recomienda que el desarrollo de aplicaciones se lo realice en varias capas, al menos 3: Presentación, Negocio y de Datos, de tal manera que el producto final pueda gozar de seguridad, fiabilidad, escalabilidad y portabilidad.

El administrador del sistema deberá contar con un perfil de formación de ingeniero o mayor, y necesitará manejar los siguientes temas: sistemas Web, bases de datos y herramientas de manejo de contenido.

Referencias Bibliográficas

- [1][3][4] Stephen R. Palmer – John M. Felsin, 2002, *Practical Guide to Feature-Driven Development*. Recuperado el 26 de agosto de 2011. Disponible en:
<http://www.ccs.neu.edu/home/lieber/courses/csg260/f06/materials/papers/fdd/pg2fddch03.pdf>
- [2] FDD, Recuperado 14 de septiembre de 2012, de http://www.ingenieriadesoftware.mex.tl/61162_FDD.html
- [5] JEE, Recuperado 27 de noviembre, 2011, de <http://www.oracle.com>
- [6] JUnit, Recuperado 2 de diciembre, 2011, de <http://www.junit.org>
- [7] Tomcat, Recuperado 6 de diciembre de 2011, de <http://www.tomcat.apache.org>
- [8] Eclipse, Recuperado 19 de marzo, 2012, de <http://www.eclipse.org/>
- [9][10] Liferay, Recuperado 19 de marzo, 2012, de <http://www.liferay.com>
- [11] Hibernate, Recuperado 21 de septiembre de 2011, de <http://www.hibernate.org>
- [12] Adobe Flex, Recuperado 11 de enero de 2012, de <http://www.adobe.com>
- [13] MySQL, Recuperado 20 de marzo, 2012, de <http://www.mysql.com>
- [14] StarUML, Recuperado 23 de Febrero de 2012, de <http://staruml.sourceforge.net>