

Sistema de gestión de la información de los procesos de recibo y molinería en la UEB Faustino Vega

Practical system for the management of the information on receipt and milling processes in the UEB Faustino Vega

Ing. Grettel Susel Incencio Peñero ^{1*}, Ing. Elena Martínez Pérez ², Ing. Lisbet Milagros Guerra Cantero ³

1 Ingeniera, Instructora, Cuba, Universidad de Granma, Facultad de Ciencias Técnicas, Departamento Informática, gincenciop@udg.co.cu

2 Ingeniera, Instructora, Cuba, Universidad de Granma, Facultad de Ciencias Técnicas, Departamento Informática, emartinezp@udg.co.cu

3 Ingeniera, Instructora, Cuba, Universidad de Granma, Facultad de Ciencias Técnicas, Departamento Informática, lguerrac@udg.co.cu

*Autor para la correspondencia: gincenciop@udg.co.cu, 23-57-8075.

RESUMEN

El uso de los sistemas de gestión que permitan automatizar los procesos de recibo y molinería en los molinos arroceros, garantiza el ahorro de tiempo y dinero, así como la disminución de los posibles errores que se puedan producir debido al proceder humano en la realización de dichos procesos. Este documento describe el proceso de desarrollo de una aplicación web, que permite automatizar los procesos de recibo y molinería en la UEB Faustino Vega.

ABSTRACT

The use of the management systems that allow automating the processes of receipt and milling in the rice mills, guarantees the saving of time and money, as well as the reduction of the possible errors that can occur due to the human proceeding in the accomplishment of said processes. This document describes the process of developing a web application, which allows automating the processes of receipt and milling in UEB Faustino Vega.

PALABRAS CLAVE

sistemas de gestión, procesos, recibo y molinería, aplicación web

KEYWORDS

management systems, processes, receipt and milling, web application

Índice

INTRODUCCIÓN.....	2
METODOLOGÍA, TECNOLOGÍAS Y HERRAMIENTAS	2
DESARROLLO DEL SISTEMA.....	11
CONCLUSIONES.....	16
REFERENCIAS	16

INTRODUCCIÓN

Actualmente en la UEB Molino Arroceros Faustino Vega Santisteban se realizan los procesos de recibo de arroz cáscara y molinería en dependencia del plan asignado por la Empresa Agroindustrial de Granos Fernando Echenique, todo el control y manejo de la información necesaria en estos procesos se realizan de forma manual, utilizando modelos impresos, y apoyándose en documentos Excel, teniendo en cuenta la gran cantidad de modelos y documentos que se generan a lo largo de una cosecha, se derivan insuficiencias a la hora de obtener cierta información en una fecha determinada, al evaluar resultados de un periodo determinado, todo esto trae consigo una pérdida sustancial de tiempo, alteración por el personal no autorizado y pérdida de la información; además pueden ocasionarse equivocaciones, ya que la gran parte de la información que se maneja son datos numéricos, trayendo serias dificultades para la empresa.

METODOLOGÍA, TECNOLOGÍAS Y HERRAMIENTAS

A continuación se describen las tecnologías, herramientas y metodología utilizadas en el desarrollo de la aplicación. Como resultado de los estudios e investigaciones realizadas, se propone la implementación de un sistema de gestión, ya que son tecnologías actualmente muy prometedoras, seguras y eficaces.

Además se propone la arquitectura cliente/servidor ya que está estrechamente relacionada con las aplicaciones web y cuenta con una serie de ventajas que le atribuyen gran importancia. Entre sus principales ventajas se pueden mencionar la posibilidad de existir varios usuarios nutriéndose de información simultáneamente, así como mayor rapidez para el mantenimiento y el desarrollo de aplicaciones; además de la facilidad de integración entre diferentes sistemas, el intercambio de información, entre otras.

Metodología de desarrollo de software

Las metodologías de desarrollo de software surgen ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un producto software. Dichas metodologías pretenden guiar a los desarrolladores al crear un nuevo software, pero

los requisitos de un software a otro son tan variados y cambiantes, que ha dado lugar a que exista una gran variedad de metodologías para la creación del software.

Para el desarrollo del software se decidió utilizar metodologías ágiles, específicamente la Programación Extrema (XP).

Programación Extrema o XP por sus siglas en inglés (Extreme Programming), es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software. Promueve el trabajo en equipo, la retroalimentación continua entre el equipo de desarrollo y el usuario final, para garantizar que las soluciones obtenidas sean lo más simple posible y sobre todo valor para enfrentarse a los cambios que puedan ocurrir. El principal objetivo de esta metodología es lograr la satisfacción del cliente, tratando de entregar al usuario final lo que necesita en el momento preciso, por lo que necesita que las respuestas sean rápidas a pesar de los cambios. Además XP es una metodología que está orientada por pruebas y refactorización, donde las pruebas se diseñan e implementan antes de programar la funcionalidad, el programador crea sus propios test de unidad. Esta metodología se puede considerar como liviana en el sentido de que pasa por alto la exhaustiva elaboración de casos de uso, definición de requisitos y la producción de una extensa documentación. XP propone como ciclo de vida las fases de planificación, diseño, codificación y prueba y como actividades básicas propone codificar, hacer pruebas, escuchar y diseñar. (Bautista, 2009).

Valores de XP:

- Comunicación.
- Simplicidad.
- Feedback.
- Coraje.

Los cuatro valores mencionados anteriormente, comunicación, simplicidad, feedback y coraje nos brindan un estándar para obtener buenos resultados. Sin embargo, los valores son muy vagos a la hora de ayudarnos a decidir que prácticas utilizar. Para ello se necesita destilar estos valores en principios que puedan ser utilizados.

Principios de XP:

- Rápida retroalimentación.
- Asumir la simplicidad.

- Cambios incrementales.

Lenguajes de programación

Un lenguaje de programación es la herramienta informática que permite establecer una correcta comunicación entre la computadora y el usuario de la misma; sin este no sería posible expresar las instrucciones que se quiere que la computadora ejecute, ni tampoco podría mostrar los resultados de dichas instrucciones.

Existen dos tipos de lenguajes, los lenguajes del lado del cliente y los lenguajes del lado del servidor. Un lenguaje del lado cliente es totalmente independiente del servidor, lo cual permite que la página pueda ser albergada en cualquier sitio. Pero la página no se verá bien si la computadora cliente no tiene instalados los plugin adecuados. El código, tanto del hipertexto como de los scripts, es accesible a cualquiera y ello puede afectar a la seguridad. (Velasco, 2014).

JavaScript

Técnicamente, el JavaScript es un lenguaje de programación interpretado muy similar al C, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de involucrar procesos intermedios.

Es un lenguaje seguro y fiable, aunque como todos, tiene algunas limitantes. Es por ello que no es posible hacerlo todo con JavaScript y se considera necesario utilizarlo conjuntamente con otros lenguajes evolucionados. En el contexto de AJAX, el JavaScript ha servido para darle funcionalidad en el lado del cliente a las aplicaciones, ya sea mediante la validación de formularios, la gestión de peticiones al servidor o los cambios dinámicos en el HTML. (Pérez C. R., 2009).

Algunas características del lenguaje son:

Lenguaje Simple: se conoce como lenguaje simple porque viene de la misma estructura de c y c++; ya que c++ fue un referente para la creación de java por eso utiliza determinadas características de c++ y se han eliminado otras.

Orientado a Objeto: toda la programación en java en su mayoría está orientada a objeto, ya que al estar agrupados en estructuras en estructuras encapsuladas es más fácil su manipulación.

Distribuido: permite abrir sockets, establecer y aceptar conexiones con los servidores o clientes remotos; facilita la creación de aplicaciones distribuidas ya que proporciona una colección de clases para aplicaciones en red.

Robusto: es altamente fiable en comparación con c, se han eliminado muchas características con la aritmética de punteros, proporciona numerosas comprobaciones en compilación y en tiempo de ejecución.

Seguro: la seguridad es una característica muy importante en java ya que se han implementado barreras de seguridad en el lenguaje y en el sistema de ejecución de tiempo real.

Portable: por ser indiferente a la arquitectura sobre la cual está trabajando, esto hace que su portabilidad sea muy eficiente, sus programas son iguales en cualquiera de las plataformas, ya que java especifica tamaños básicos, esto se conoce como la máquina virtual de java.

Interpretado y compilado a la vez: java puede ser compilado e interpretado en tiempo real, ya que cuando se construye el código fuente este se transforma en una especie de código de máquina.

Multihebra o Multihilos: java tiene una facilidad de cumplir varias funciones al mismo tiempo, gracias a su función de multa hilos ya que por cada hilo que el programa tenga se ejecutaran en tiempo real muchas funciones al mismo tiempo.

Dinámico: este lenguaje es muy dinámico en la fase de enlazado, sus clases solamente actuarán a la medida en que sean requeridas o necesitadas con esto permitirá que los enlaces se puedan incluir incluso desde fuentes muy variadas o desde la red.

Produce Applets: en java se pueden crear aplicaciones independientes y applets. Independientes porque se pueden comportar como cualquier programa escrito en cualquier lenguaje. Por otra parte los applets considerados pequeños programas, tienen la capacidad de ejecutar funciones muy complejas.

Alto rendimiento: java es considerado de alto rendimiento por ser tan veloz en el momento de correr los programas y por ahorrarse muchas líneas de código. (Rosabal, 2011).

Personal Home Page (PHP)

En 1994 Rasmus Lerdorf, miembro del equipo de desarrollo de Apache, creó PHP. Es el lenguaje del lado del servidor más extendido en la web, pues ha alcanzado una gran aceptación en la comunidad de desarrolladores, debido a la potencia y simplicidad que lo caracterizan, así como al soporte generalizado en la mayoría de los servidores de hosting. (Granados, 2004)

Es un lenguaje de código abierto muy popular, adecuado para desarrollo web y que puede ser incrustado en HTML. PHP tiene un gran parecido en su sintaxis con lenguajes de programación estructurada, como C, permitiendo a los programadores crear aplicaciones complejas con un corto

periodo de aprendizaje. También permite crear aplicaciones sin necesidad de aprender un gran volumen de funciones.

La orientación a objetos no está muy lograda en PHP 4.0. Los objetos tienen un tratamiento muy pobre e ilógico. La definición de las variables miembro (propiedades) y los métodos son siempre públicos, por lo que la encapsulación es nula. Todos los objetos se pasan como valor por defecto cuando deberían pasarse por referencia. Todas estas propuestas realizadas por el equipo de desarrollo de PHP han desembocado en la creación del motor Zend 2.0 y su consecuencia PHP 5.

PHP 5 incorpora una verdadera orientación a objetos. Añadiendo las palabras reservadas public, protected y private a la definición de las propiedades y métodos de los objetos, se permite una verdadera encapsulación. Además del considerable avance con respecto a los objetos, PHP 5 incorpora un control de errores muy mejorado, al estilo de los lenguajes de programación más avanzados. (González, 2014).

Sistema gestor de base de datos

Un Sistema Gestor de Bases de Datos (SGBD por sus siglas en inglés) es un conjunto de programas que permite a los usuarios crear y mantener una Base de datos (BD), por lo tanto, el SGBD es un software de propósito general que facilita el proceso de definir, construir y manipular la BD para diversas aplicaciones, pueden ser de propósito general o específico. (ELMASRI, 2004).

My Structured Query Language (MySQL). Es uno de los SGBD más populares, desarrollado bajo la (Roberth G. Figueroa, 2008) filosofía de código abierto. La desarrolla y mantiene la empresa MySQLAB pero puede utilizarse gratuitamente y su código de fuente está disponible. Es un gestor de bases de datos multiusuario, el más rápido y robusto, de su categoría, tanto para volúmenes de datos grandes como pequeños en entornos web. (GARCÍA, 2007)

Ventajas:

- Mayor rendimiento.
- Mayor velocidad tanto al conectar con el servidor como al servir selects y demás.
- Mejores utilidades de administración (backup, recuperación de errores, etc.).
- Aunque se cuelgue, no suele perder información ni corromper los datos.
- Mejor integración con PHP.

- No hay límites en el tamaño de los registros.
- Mejor control de acceso, en el sentido de que usuarios tienen acceso a que tablas y con qué permisos.

Inconvenientes:

- No soporta transacciones, "roll-backs" ni subselects.
- No considera las claves ajenas.
- Ignora la integridad referencial, dejándola en manos del programador de la aplicación.

Lenguaje de modelado

A mediados de los noventa existían muchos métodos de análisis y diseño Orientado a Objetos (OO), lo que suponía que los mismos conceptos tenían distinta notación según el método de que se tratara. Ante esta situación de confusión, en 1994 Booch, Rumbaugh y Jacobson decidieron unificar sus métodos dando lugar a UML. Esta unificación fue promovida por el OMG (Object Management Group) de tal manera que UML se convirtió en la notación estándar para la descripción de métodos de software. Según su definición, UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software, desde una perspectiva.

UML es un lenguaje: proporciona un vocabulario y unas reglas que se centran en la representación conceptual y física de un sistema, y que indican cómo crear y leer modelos bien formados. Sin embargo, no dice qué modelos crear ni cuándo se deberían crear, ésta es la tarea del proceso de desarrollo de software.

UML es un lenguaje para visualizar: es un lenguaje gráfico que mezcla gráficos y texto. Detrás de cada símbolo en la notación UML hay una semántica bien definida, de manera que un desarrollador puede escribir un modelo en UML, y otro desarrollador, o incluso otra herramienta, puede interpretar ese modelo sin ambigüedad.

UML es un lenguaje para especificar: cubre la especificación de todas las decisiones de análisis, diseño e implementación que deben realizarse al desarrollar y desplegar un sistema con gran cantidad de software.

UML es un lenguaje para construir: no es un lenguaje visual, pero sus modelos pueden conectarse de forma directa con una gran variedad de lenguajes de programación. Es posible establecer correspondencias desde un modelo UML a un lenguaje de programación como Java, C++, tablas en

una base de datos relacional o al almacenamiento persistente en una base de datos orientada a objetos. Permite ingeniería directa e inversa.

UML es un lenguaje para documentar: cubre toda la documentación de la arquitectura de un sistema y todos sus detalles. También proporciona un lenguaje para expresar requisitos y pruebas del software. Finalmente, UML proporciona un lenguaje para modelar las actividades de planificación de proyectos y gestión de versiones. (Díaz, 2011).

Herramienta CASE

En la ingeniería de software se han establecido diversos procesos, metodologías y herramientas para estandarizar y facilitar el desarrollo de sus productos. Entre estas herramientas se encuentran las enmarcadas dentro de la ingeniería de software asistida por computadoras, las cuales soportan de forma automática varios o todos los pasos de dichas metodologías. Son estas últimas las que se identifican bajo el término CASE por sus siglas en inglés, de Computer Aided Software Engineering y son ampliamente utilizadas por los desarrolladores de software. (Cobos, 2010).

Las herramientas CASE se puede definir entonces como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de desarrollo del mismo (investigación preliminar, análisis, diseño, implementación e instalación). Ellas ayudan a reducir el tiempo de desarrollo de los sistemas informáticos, manteniendo los costos relativamente estables y contribuyendo a mejorar la calidad de los mismos. (Alarcón, 2008).

Embarcadero ER/Studio

El Embarcadero ER/Studio es una poderosa herramienta CASE que ofrece a los administradores y desarrolladores de bases de datos la posibilidad de modelar las mismas de forma visual. Permite discernir convenientemente entre modelos lógicos y físicos de bases de datos relacionales y multidimensionales, aunque siempre mantiene una total integración entre ellos. Soporta además el diseño multinivel y ofrece la capacidad de controlar, documentar y desplegar rápidamente los cambios de diseño en las plataformas más comunes.

Esta herramienta es compatible con las principales bases de datos del mercado internacional (Oracle, Sybase, IBM DB2 UDB y Microsoft SQL Server) y soporta casi todas las plataformas. Adicionalmente, su interfaz es muy amigable y cuenta con integración tanto para exportar como para importar metadatos ya sea desde o hacia los principales productos informáticos relacionados. (Aliaga, 2012).

Entorno de Desarrollo Integrado

Un entorno de desarrollo integrado o IDE (acrónimo en inglés de Integrated Development Environment), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios.

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). (Corporation, 2015)

Netbeans

Es un entorno integrado de desarrollo o IDE. En él podemos realizar todas las tareas asociadas a la programación. Simplifica alguna de las tareas que, sobre todo en proyectos grandes, son tediosas.

Asiste (parcialmente) en la escritura de código, aunque no nos libera de aprender el lenguaje de programación. Ayuda en la navegación de las clases predefinidas en la plataforma (miles). Aunque puede ser costoso su aprendizaje, los beneficios superan las dificultades. (Gimeno, 2011)

Framework de desarrollo

La palabra framework se traduce como marco, almacén, estructura de desarrollo o conjunto de bibliotecas orientadas a la reutilización a gran escala de componentes de software para el desarrollo rápido de aplicaciones, se emplea en muchos ámbitos del desarrollo de sistemas de software. En general, el término framework, se refiere a una estructura de software que posee componentes personalizables e intercambiables para el desarrollo de una aplicación. 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, se define como un conjunto de componentes (por ejemplo clases en java, descriptores y archivos de configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas web. (Gutiérrez, 2010).

CodeIgniter

CodeIgniter es un framework para el desarrollo de aplicaciones, un conjunto de herramientas para los usuarios que construyen sitios web usando PHP. Su objetivo es permitirle desarrollar proyectos mucho más rápido que lo que podría hacer si escribiera el código desde cero, facilitando un rico conjunto de bibliotecas para tareas comunes, así como una interfaz sencilla y una estructura lógica

para acceder a esas bibliotecas. CodeIgniter le permite enfocarse creativamente en su proyecto al minimizar la cantidad de código necesaria para una tarea dada.

Como cualquier otro framework, CodeIgniter contiene una serie de librerías que sirven para el desarrollo de aplicaciones web y además propone diferentes maneras de desarrollarlas las cuales debemos seguir para obtener provecho de la aplicación. CodeIgniter implementa el proceso de desarrollo llamado modelo vista controlador (MVC), que es un estándar de programación de aplicaciones, utilizado tanto para hacer sitios web como programas tradicionales. Este framework contiene muchas ayudas para la creación de aplicaciones de PHP avanzadas, que hacen que el proceso de desarrollo sea más rápido. A la vez, define una arquitectura de desarrollo proporcionando una manera de programar más ordenada y contiene diversas herramientas que ayudan a hacer aplicaciones más versátiles y seguras. Con la utilización de los frameworks de PHP se crean aplicaciones web más profesionales y con código más reutilizable, con la diferencia que codeigniter está creado para que sea más fácil de instalar en cualquier servidor y de empezar a usarse que cualquier otro framework. Además muchas de sus utilidades y modos de funcionamiento son opcionales, lo cual brinda una mayor libertad a la hora de desarrollar sitios web. (Alvarez, 2012).

Características generales de CodeIgniter.

- Compatibilidad.
- Documentación tutorializada.
- Flexibilidad.
- Sistema Basado en Modelo-Vista-Controlador.
- Extremadamente Liviano.
- Facilidad de instalación.

Servidor web

Un servidor web es un programa que procesa cualquier aplicación del lado del servidor, realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente, generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente.

El servidor web se encarga de contestar a estas peticiones de forma adecuada y entrega como resultado una página web o información de todo tipo, de acuerdo a los comandos solicitados. Este

intercambio es mediado por el navegador y el servidor que se comunica el uno con el otro mediante HTTP.

Se pueden utilizar varias tecnologías en el servidor para aumentar su potencia más allá de su capacidad de entregar páginas HTML; éstas incluyen scripts CGI (Common Gateway Interface), seguridad SSL (Secure Socket Layer) y páginas activas del servidor (ASP). (Robles, 2012)

Servidor Web Apache

El servidor web Apache se caracteriza por su sencilla configuración, su robustez y estabilidad. Es muy sencillo ampliar las capacidades del servidor web apache. Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar apache para que ejecute un determinado script cuando ocurra un error en concreto. Es un servidor web flexible, rápido, eficiente, continuamente actualizado y adaptado a los nuevos protocolos HTTP, además es multiplataforma. (Robles, 2012)

El servidor Apache es un software que está estructurado en módulos. La configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo. Los módulos del Apache se pueden clasificar en tres categorías:

Módulos Base: módulo con las funciones básicas del Apache.

Módulos Multiproceso: son los responsables de la unión con los puertos de la máquina, aceptando las peticiones y enviando a los hijos a atender a las peticiones

Módulos Adicionales: cualquier otro módulo que le añada una funcionalidad al servidor.

DESARROLLO DEL SISTEMA

La aplicación contará con dos tipos de usuarios principalmente. El administrador del sistema será el encargado de gestionar toda la información relacionada con el sistema. Cuenta además con todos los privilegios. El empleado representa a cualquier trabajador de la UEB que desee consultar la información referente a los servicios que se les han prestado a los clientes.

La base de datos registrará de forma unificada todas las evaluaciones realizadas, manteniendo así la base de datos de la aplicación actualizada con la información necesaria, garantizando la estabilidad y fácil mantenimiento del sistema.

Funcionalidades del sistema

1. Autenticar usuario: el sistema permite mostrar, agregar nuevos usuarios, además de eliminar o actualizar cualquier dato de los usuarios registrados.
3. Gestionar recibos: el sistema permite insertar, modificar, eliminar y visualizar algún recibo existente.
4. Gestionar lugar de procedencia: el sistema permite insertar, modificar, eliminar y visualizar los lugares de procedencia existentes.
5. Gestionar técnico de calidad: el sistema permite insertar, modificar, eliminar y visualizar los nombres de los técnicos de calidad.
6. Gestionar pesador: el sistema permite insertar, modificar, eliminar y visualizar los nombres de los pesadores.
7. Gestionar búsqueda por fecha: El sistema muestra todos los recibos existentes en una fecha determinada.
8. Gestionar búsqueda por lugar de procedencia: el sistema muestra todos los recibos provenientes de un lugar determinado (escogido por el usuario).
9. Gestionar búsqueda por técnico de calidad: el sistema muestra todos los recibos en los cuales trabajó un técnico de calidad específico (escogido por el usuario).
10. Gestionar búsqueda por humedad fuera de parámetros: el sistema muestra todos los recibos con el por ciento de humedad fuera de parámetros.
11. Gestionar búsqueda por impureza alta: el sistema muestra todos los recibos con el por ciento de impureza alto (mayor de 18).
12. Gestionar búsqueda por procedencia: el sistema muestra todos los recibos dependiendo de su procedencia (estatal o campesino).
13. Gestionar turnos: el sistema permite crear, modificar, eliminar y visualizar los turnos existentes (número de turno y nombre del jefe de turno).
14. Gestionar producción: el sistema permite crear, modificar, eliminar y visualizar las producciones existentes.
15. Gestionar parte diario de producción: el sistema genera el parte de producción de cualquier día introduciéndole una fecha, si existió producción en esa fecha.

16. Gestionar parte por rango de fechas: el sistema genera un parte de producción en un rango de fechas (escogido por el usuario).
17. Gestionar parte general por turnos: el sistema genera un parte general de producción de cada turno.
18. Gestionar reporte de cantidad de cáscara recibida: el sistema muestra la cantidad de cáscara recibida en general (en bruto y neta).
19. Gestionar reporte de cantidad de cáscara molinada: el sistema muestra la cantidad de cáscara molinada en general por todos los turnos.
20. Gestionar reporte de cantidad de cáscara real almacenada: el sistema muestra la cantidad de cáscara real almacenada en los silos.
21. Gestionar reporte general de todos los productos envasados: el sistema muestra la cantidad de productos envasados.
22. Generar gráfica sobre el rendimiento de cada turno: el sistema muestra una gráfica sobre el rendimiento de cada turno.
23. Generar gráfica sobre la cantidad de arroz consumo producido por turnos: el sistema genera una gráfica mostrando la cantidad de arroz consumo producido por cada turno.
24. Generar gráfica sobre la cantidad de cáscara molinada por turnos: el sistema genera una gráfica mostrando la cantidad de arroz cáscara molinada por cada turno.

Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que debe tener el producto, las cuales representan las características que hacen al producto atractivo, rápido y confiable. Seguidamente se especifican los requerimientos no funcionales del sistema propuesto.

Apariencia o interfaz externa:

- Optimizado para una resolución de 1024x768.
- Interfaz amigable y sencilla, con colores suaves a la vista y sin cúmulo de objetos que le sean incómodos al usuario a la hora de interactuar con el sistema.

- Un menú principal que le brinde al usuario rapidez y facilidad para obtener la información que desea.

Software para el cliente:

- Sistema operativo multiplataforma con interfaz gráfica.
- Navegador web (Internet Explorer 6.0 o superior, Mozilla Firefox 3.0 o superior, Opera, Safari).

Software para el servidor:

- Framework Symfony 1.4.10 o superior.
- Servidor web Apache 2.2.12 o superior.
- Versión de PHP 5.3.0 o superior.
- Gestor de Base de Datos MySQL 5.1.37 o superior.
- Software controlador de versiones: CollabNetSubversion-client-1.6.15 o superior.

Hardware para el desarrollo:

- PC con las siguientes características: Intel Pentium 4 o superior, CPU 3GHz o superior, 1GB RAM o superior, 80 GB HDD o superior.

Hardware para explotación del cliente:

- PC con las siguientes características: Intel Pentium 4 o superior, CPU 3GHz o superior, 256 MB RAM mínimo, 512 RAM recomendada o superior.

Hardware para explotación del servidor:

- PC con las siguientes características: Intel Pentium 4 o superior, RAM: 1 GB mínimo (Recomendado 2 GB), 80 GB de disco duro.

Usabilidad:

- Para hacer uso del sistema es necesario poseer conocimientos elementales de computación y sobre el ambiente web en sentido general.
- El diseño debe tener buena visibilidad en los principales navegadores.
- El sistema poseerá estructura y diseño homogéneos en todas sus pantallas, que facilite la navegación tales como menús laterales y/o desplegados que permitan el acceso rápido a la información.

Soporte:

- Sistema multiplataforma.
- Los sistemas deberán estar abiertos a las modificaciones que se requieran, ya sea ante la detección de un fallo o por nuevos requisitos. Después de su puesta en explotación se le dará mantenimiento una vez al año.

Seguridad:

- Chequear que el usuario esté autenticado antes de que pueda realizar alguna acción sobre el sistema.
- La información manejada por el sistema estará protegida de acceso no autorizado y divulgación.
- Garantizar que las funcionalidades del sistema se muestren de acuerdo al tipo de usuario que esté activo.
- Realizar salvallas periódicas de la información y base de datos en otros dispositivos, como solución ante la ocurrencia de problemas.
- Llevar un registro de sucesos donde se archiven los eventos del sistema incluyendo los eventos de error, inicio de sesión, cierre de sesión y modificación de la información.

Disponibilidad:

- El sistema deberá tener un 100% de disponibilidad por lo que podrá ser usado las 24 horas del día por todos sus clientes.

Diseño

El patrón Modelo-Vista-Controlador es una guía para el diseño de arquitecturas de aplicaciones que ofrezcan una fuerte interactividad con usuarios. Este patrón organiza la aplicación en tres modelos separados, el primero es un modelo que representa los datos de la aplicación y sus reglas de negocio, el segundo es un conjunto de vistas que representa los formularios de entrada y salida de información, el tercero es un conjunto de controladores que procesa las peticiones de los usuarios y controla el flujo de ejecución del sistema. (Gutiérrez, 2010)

- **El Modelo:** representa sus estructuras de datos. Típicamente sus clases del modelo contendrán funciones que los ayudarán a devolver, insertar y actualizar información de su base de datos.
- **La Vista:** es la información que se presenta al usuario. Una vista será normalmente una página web, también puede ser un fragmento de página como el encabezado o pié de página. En el desarrollo web para crear una vista se utiliza HTML y PHP, sin contar con que se puede agregar CSS y JavaScript para darle más dinámica al sitio y mejor apariencia.
- **El Controlador:** sirve como un intermediario entre el Modelo, la Vista y cualquier otro recurso necesario para procesar la solicitud HTTP y generar una página web.

CONCLUSIONES

Una vez concluido el proceso de desarrollo del sistema se puede concluir que: el desglose de las historias de usuario en tareas de la ingeniería permitió facilitar la implementación de las funcionalidades, asimismo la utilización del marco de trabajo facilitó la correcta implementación del patrón arquitectónico Modelo-Vista-Controlador, asegurando la aplicación contra ataques comunes.

Este trabajo finaliza con la obtención de una aplicación web que dotó a la UEB Faustino Vega de un sistema de gestión, que permite la gestión de la información en los procesos de recibo y molinería en el molino arrocero.

Este resultado ha sido realizado de manera flexible ante posibles cambios en el proceso de desarrollo de software de la organización donde se aplica, proponiendo tecnologías libres acordes con las políticas que existen en el país y en la Universidad de Granma para lograr una independencia tecnológica.

REFERENCIAS

1. Alarcón, A. &. (2008). Herramientas CASE para ingeniería de Requisitos. Cultura Científica 6.
2. Aliaga, D. M. (2012). Sistema de gestión Administrativa del Proceso Docente para la Facultad de Ciencias Técnicas de la Universidad de Granma.
3. Alvarez, M. A. (2012). Manual de CodeIgniter
4. <http://www.desarrolloweb.com/articulos/introducción-a-jquery.html>.
5. Barrios, Y. P. (2013). Sistema de Gestión de Información del Consumo Energético de la UEB Molino Arrocero "Faustino Vega".

6. Bautista, J. (2009). "Extreme Programming (XP).".
7. Briggs, O. C. (2003). Cascading Style Sheets .
8. <http://www.desarrolloweb.com/manuales-css-hojas-de-estilo.html>
9. Castillo, A. (2010). "Manual de HTML5 en español.".
10. CAVSI. (2014). "¿Qué es un servidor web?". De <http://www.cavsi.com/preguntasrespuestas/que-es-un-servidor-Web/>.
11. Cobos, C. Z. (2010). CMIN - a CRISP-DM-based case tool for supporting data mining projects Ingeniería e Investigación. Bogotá, 30(3), 12.
12. Corporation, L. (2015). "Entornos de Desarrollos Integrados." from <http://es.slideshare.net/GhaBiithahh/entornos-de-desarrollo-integrados>.
13. Díaz, G. A. (2011). Sistema para la Gestión de Información en el Área de Investigación y Posgrado de la Universidad de Granma. Universidad de Granma, Bayamo.
14. Elmasri, N. (2004). Conceptos y arquitectura de un sistema de bases de datos.
15. García, F. a. (2011). Naturaleza y características de los servicios y los contenidos digitales abiertos.
16. García, R. M. (2007). Sistemas Gestores de Bases de Datos.
17. Gimeno, J. M. (2011). "Introducción a Netbeans ".
18. González, E. (2014). "Tutorial básico del programador web: PHP desde cero.".
19. Granados, C. (2004). Manual Imprescindible de PHP.
20. Gutiérrez, J. J. (2010). ¿Qué es un framework web?
21. Isaías Carrillo Pérez, R. P. (2008). Metodología de desarrollo del Software. from http://www.cyta.com.ar/ta0502/b_v5n2a1.htm
22. Madruga, P. M. (2014). "Sistemas gestores de bases de datos.".
23. Pérez, C. R. (2009). Desarrollo de Software de Gestión para la Producción Bovina.
24. Pérez, D. (2007). "¿Qué es Javascript?".
25. Piñeiro, H. Á. (2012). Sistema informático para el proceso de gestión de la información del departamento de maquinaria de la Empresa Agroindustrial de Granos.

26. Ravioli. (2009). "Lenguaje de programación para página web."
27. Roberth G. Figueroa, C. J. (2008). METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES.
28. Robles, A. B. (2012). Módulo Portadores energéticos de la Plataforma Libre para la Gestión de Procesos de la Facultad Regional Granma de la Universidad de las Ciencias Informáticas.
29. Rosabal, A. S. (2011). "Sistema para el Control de Actividades Científicas " Cuadernos de Educación y Desarrollo.from <http://www.quizzpot.com/2009/01/ext-js-framework/>
30. Velasco. (2014). Lenguajes de programación del lado del cliente
31. <http://www.prograweb.com.mx/pweb/0202ladoCliente.html>.
32. Velasco, J. E. P. (2011b). Lenguajes de programación del lado Servidor, from <http://www.prograweb.com.mx/pweb/0203ladoServidor.html>.