

Isael Sañay Sañay ^a, Edison Becerra Molina ^b; Oscar Calle Masache ^c

La ingeniería del software para el desarrollo industrial en la zona 6 del Austro

Software engineering for industrial development in zone 6 of the Austro

*Revista Científica Mundo de la Investigación y el Conocimiento. Vol. 3 núm.1,
enero, ISSN: 2588-073X, 2019, pp. 1625-1643*

DOI: [10.26820/recimundo/3.\(1\).enero.2019.1625-1643](https://doi.org/10.26820/recimundo/3.(1).enero.2019.1625-1643)

URL: <http://recimundo.com/index.php/es/article/view/500>

Código UNESCO: 3204.02 Enfermedades Profesionales

Tipo de Investigación: Artículo de Investigación

Editorial Saberes del Conocimiento

Recibido: 15/11/2018

Aceptado: 18/12/2018

Publicado: 25/01/2019

Correspondencia: ssanay@ucacue.edu.ec

- ^a. Magister en Sistemas de Información Gerencial y Profesor Investigador de la Unidad Académica de Tecnologías de la Información y Comunicación. Universidad Católica de Cuenca. E-mail: ssanay@ucacue.edu.ec
- ^b. Magister en Gerencia de Proyectos y Profesor Investigador de la Unidad Académica de Administración. Universidad Católica de Cuenca. E-mail: jbecerram@ucacue.edu.ec
- ^c. Magister en Administración de Empresas con Mención en Recursos humanos y Marketing y Profesor investigador de la Unidad Académica Administración. Universidad Católica de Cuenca. E-mail: ocalle@ucacue.edu.ec

RESUMEN

Para contribuir con el objetivo planteado por el estado ecuatoriano, hemos propuesto implementar una metodología denominada modelos eficientes de procesos del negocio para empresas de producción continua, la misma que estará basada en el uso de las mejores prácticas, que oriente la automatización de la organización de una manera integrada, conectando los procesos en el piso de planta con los procesos del negocio. El presente caso de estudio se aplicó en la Compañía de Economía Mixta, "Unión Cementera Nacional, UCEM". La presente investigación involucra distintos elementos funcionales que interactúan entre sí, como: la planificación de la producción, órdenes de trabajo, mantenimiento, gestión de bodegas, control de la calidad, recursos humanos, logística y manejo de halls; a esta fase se le conoce como el Sistema de Ejecución de Producción (MES) presente en todo proceso productivo; siendo el objetivo de esta investigación desarrollar un software que se adapte a cualquier sistema productivo de las pequeñas y medianas empresas (PYMES), que permitan la autonomía de cada área de producción e implementar un modelo eficiente de contabilidad de costos; pretendemos enlazar la gestión de la administración con el piso de planta es decir se aprecia una referencia-cruzada a nivel horizontal y vertical de los procesos productivos de las organizaciones.

Palabras Claves: PYMES; MES; Contabilidad; Ingeniería del software.

ABSTRACT

In order to contribute to the objective set by the Ecuadorian state, we have proposed to implement a methodology called efficient business process models for continuous production companies, which will be based on the use of best practices, which guides the automation of the organization of an integrated way, connecting the processes on the floor of the plant with the processes of the business. The present case study was applied in the Mixed Economy Company, "Unión Cementera Nacional, UCEM". This research involves different functional elements that interact with each other, such as: production planning, work orders, maintenance, warehouse management, quality control, human resources, logistics and handling of halls; This phase is known as the Production Execution System (MES) present in all production processes; The objective of this research is to develop software that adapts to any productive system of small and medium-sized enterprises (SMEs), which allow the autonomy of each production area and implement an efficient model of cost accounting; We intend to link the management of the administration with the floor of the plant, that is to say, there is a cross-reference at a horizontal and vertical level of the productive processes of the organizations.

Key Words: PYMES; MONTH; Accounting; Software engineering.

Introducción.

Los conceptos y constructos relacionados con la “Fundamentación teórica” busca relacionar la teoría con la aplicación del modelo de desarrollo de software (cascada), del presente artículo “La ingeniería del software para el desarrollo industrial en la zona 6 del Austro” orientado a las PYMES según EcuadorLegalOnline (2015) en su portal WEB define a las PYMES como el “conjunto de pequeñas y medianas empresas que de acuerdo a su volumen de ventas, capital social, cantidad de trabajadores, y su nivel de producción o activos presentan características propias de este tipo de entidades económicas” siendo estas organizaciones el motor de la reactivación de la matriz productiva y la fuente de generación de empleo en el país; propuesta por el gobierno nacional declarados en los objetivos del Plan Nacional del Buen Vivir (2017-2021).

Actualmente, las PYMES carecen de sistemas que apoyen la gestión de producción (software industrial) en concreto de un sistema MES (Sistema de Ejecución de Manufactura) como lo menciona Cordero et al. (2016), razón por lo que la investigación se centra en el análisis, la descripción, la formulación de modelos (flujo de actividades o modelo del negocio), e implementación del software usando ciclo de vida en CASCADA; para lograr esta meta resulta imperioso revisar filosofías, estándares (ISA85, ISA95), y ontologías; que han sido validadas y probadas como mencionan los investigadores Chacón et al. (2016), lo mencionado ha permitido establecer el prototipo funcional de la arquitectura de la aplicación del MES orientado a las PYMES.

Cabe destacar que la presente investigación permitió desarrollar un software contable que se adapte a cualquier sistema productivo de las pequeñas y medianas empresas (PYMES);

enlazando la gestión de la administra con el piso de planta, es decir una referencia-cruzada a nivel horizontal y vertical de los procesos productivos de las organizaciones.

Con la ingeniería del software aplicado a la industria, el análisis indico el uso de metodologías, estándares y de distintas herramientas que se usan en el proceso de desarrollo de productos de programación aplicados al ciclo de vida del producto; se espera reducir costos de desarrollo y por ende el costo del software esto ha de permitir ser accesible a las PYMES los mismos que cumplan con especificaciones de sistemas informáticos comerciales.

Fundamentación teórica

La aplicación del modelo en cascada de desarrollo de software seleccionado, conlleva ejecutar diferentes actividades como es la comunicación, la planeación, modelado, construcción y despliegue como lo indican Pressman (2010), la ejecución de las actividad del proceso genera productos en el ciclo de vida del software artefactos, Pressman (2010), Sommerville (2016) menciona los documentos, datos, diagramas, código ejecutable (...), como por ejemplo en la etapa de comunicación se obtiene: el documento de especificación de requerimientos, diagramas de casos de uso, glosario de términos y la descripción de la arquitectura; la etapa de modelado (análisis y el diseño) sus entregables son: diagrama de clases, diagrama de datos, diagrama de despliegue, casos de pruebas; en la etapa de implementación se han creado los algoritmos y se prueban los módulos utilizando distintas estrategias como: pruebas unitarias, pruebas modulares, pruebas de integración y otras.

En la ejecución de las actividades del ciclo de vida del software se utilizaron herramientas de apoyo, es así que en ingeniería de requerimientos se usó BPM, IDEF, UML y redes de Petri

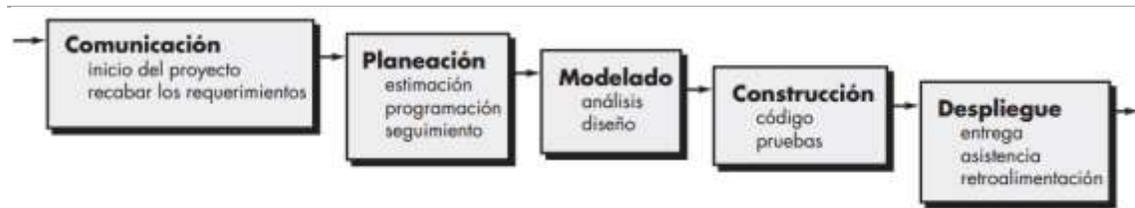
Chacón (2009) para establecer el modelo estático y dinámico parte de la comprensión del domino de la aplicación; siguiendo este enfoque en la recopilación de la información se empleó métodos interactivos y discretos como lo indica Kendall & Kendall, (2011).

En la implementación del software fue necesario emplear herramientas de gestión, en este caso de estudio, se utiliza Maven con el fichero POM.XML (Project Object Model) que permite describir el proyecto de software a construir, sus dependencias y componentes externos, y toda la gestión del ciclo de vida del software. Maven permite mediante las dependencias agregar las herramientas de desarrollo de java como es: SPRING, PRIMFACE, XHTMA, JREPORT, HYBERNATE, MOTORES DE BASES DE DATOS, Y OTROS.

Metodología.

El tipo de investigación está ligado con el diseño de campo y el nivel descriptivo, en cuanto se analizaron los procesos, basados en el enfoque utilizado para la descripción de la empresa, es decir la concepción holónica, lo que permite asegurar la autonomía de cada área de producción. Para cada área se incorporaron los elementos que permiten describir sus sistemas, los insumos y los productos utilizados y llevar la contabilidad de los mismos.

Cada área tiene sus métodos para realizar sus procesos, cada proceso se ejecuta asociado a una orden de producción o trabajo que se asocia a los procesos del negocio. En el caso de vida del desarrollo de software se seleccionó el modelo en cascada con la metodología orientada a objetos.



Resultados.

Diseño de la arquitectura

La investigación se planteó en diferentes etapas: 1) La especificación de la arquitectura de software para cubrir las funciones del Sistema de Gestión de Operaciones, según Cordero et al. (2016) y la arquitectura de Hardware necesaria para el sistema y su integración con el sistema de control de piso de planta (SCADA) y 2) 1. La elaboración de un prototipo de manufactura holónica cuyo entregable considerado el MES será tema de discusión.

La línea base de la arquitectura de la aplicación MES está soportada por la arquitectura modelo de servicios modificación del estilo arquitectónico modelo vista controlador Sommerville (2010), el cual es un patrón que permite separar la interfaz del usuario de la lógica del negocio, y brinda aspectos de reusabilidad y flexibilidad, la arquitectura da soporte al desarrollo de aplicaciones en distintos ambientes como es web, escritorio, cliente-servidor (...), en torno a este estilo existe distintos frameworks que soporta el patrón arquitectónico, entre ellos esta Spring, asp.net MVC, Struts, Primefaces entre otros, la figura 1, muestra el diagrama UML de la arquitectura Modelo Vista Controlador y servicios de aplicación (MVC+S).

La capa controladora, maneja toda la lógica del negocio, y se controlan los diferentes eventos que ejecuta el usuario, convirtiéndose en eje principal de la arquitectura, ésta se

comunica con el modelo y se encarga de mostrar los resultados a través de la vista (la interfaz de los usuarios).

En la capa de modelo se define las entidades que pertenecen a la base de datos que ha de permitir que persista la información que se envía desde la vista, de hecho, que esto se maneja con repositorios de información conocidos como beans.

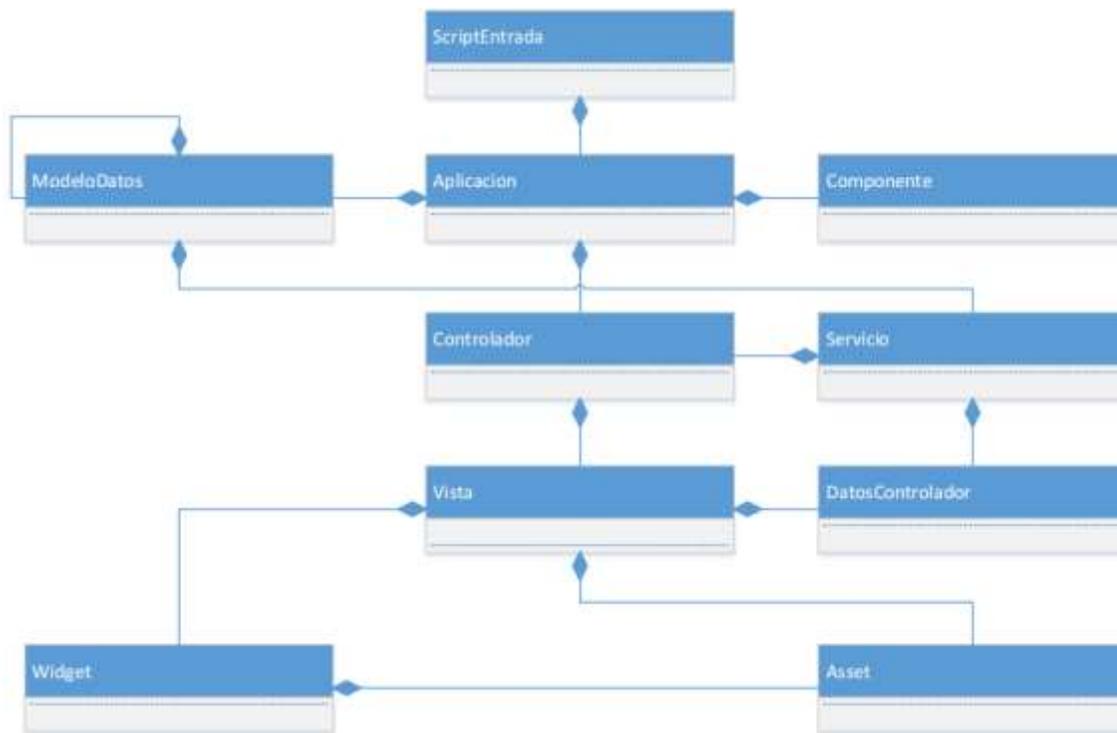


Figura 1. Estilo arquitectónico MVC+S

Elaborado: Sañay, Becerra, Calle (2018)

La capa de vista es la que controla la interacción hombre máquina (IHM) y se encarga de mostrar la información al usuario, a través de esta capa se ejecutan las diferentes peticiones o procesos que permiten facilitar el trabajo diario del usuario aclarando que, al ser el desarrollo en software libre, los desarrolladores están en la libertad de elegir las herramientas según su necesidad en la implementación de interfaz.

Las capas son manejadas por la arquitectura de servicio y MVC, la diferencia se da en la arquitectura de modelo de servicios, ya que esta implementa una capa adicional conocida con el nombre de servicio, en esta capa se maneja el flujo de trabajo (work flow) de las actividades las mismas que son modelos dinámicos que se pueden representar mediante redes de Petri o diagramas UML, de igual forma se puede realizar el control de la persistencia e implementar servicios adicionales que no requiere de una validación adicional y que pueda ser usado por cualquiera de las capas anteriores, en la figura 1, se especifica la comunicación que existe entre éstas capas, la comunicación de las capas indica que lo más importante es la del modelo, ya que todas las capas dependen de ésta, el inicio de toda actividad empieza a través de la misma que realiza una solicitud al controlador, el cual aplica las reglas del negocio para validar la petición, cuando la solicitud es válida éste la transfiere al servicio o la realiza el mismo según sea la necesidad y devuelve un estado de la transacción, la cual es transmitida hacia el usuario a través de la misma capa por la cual se hizo la solicitud en este caso la vista.

Una vez entendido la importancia del Patrón MVC y el Modelo de servicios en el desarrollo de sistemas que poseen varios tipos de clientes, llega el momento de indicar en dónde encaja el método de organización en capas conceptuales. La organización consiste en la estructura de paquetes que globalmente dividirá al sistema en componentes o módulos los

mismos que pueden tener varios clientes (dispositivos inteligentes, WEB, escritorio, servidor), según la cantidad de componentes que se agreguen a la aplicación.

Paquetes para los Módulos del Sistema

El documento especificación de requerimiento permitió identificar la necesidad de información de la capa MES; en él se integran los sistemas de Planificación y Programación Holónica, el Sistema de Laboratorio (de Gestión de la Calidad), Manejo de Personal, Bodega de Materia Prima y Productos Intermedios y Terminados, Ingreso y Egreso de Productos, Manejo de Órdenes, y Logística de Transporte en Halls, la figura 2, muestra la lista de aplicaciones a ser implementadas.

Paquetes para los Módulos del Sistema

El documento de especificación de la arquitectura Cordero et al. (2016) de software para cubrir las funciones del Sistema de Gestión de Operaciones permitió identificar la necesidad de información de la capa MES; en él se integran los sistemas de Planificación y Programación Holónica, el Sistema de Laboratorio (de Gestión de la Calidad), Manejo de Personal, Bodega de Materia Prima y Productos Intermedios y Terminados, Ingreso y Egreso de Productos, Manejo de Órdenes, y Logística de Transporte en Halls, la figura 2, muestra la lista de aplicaciones que deberán ser desarrolladas.

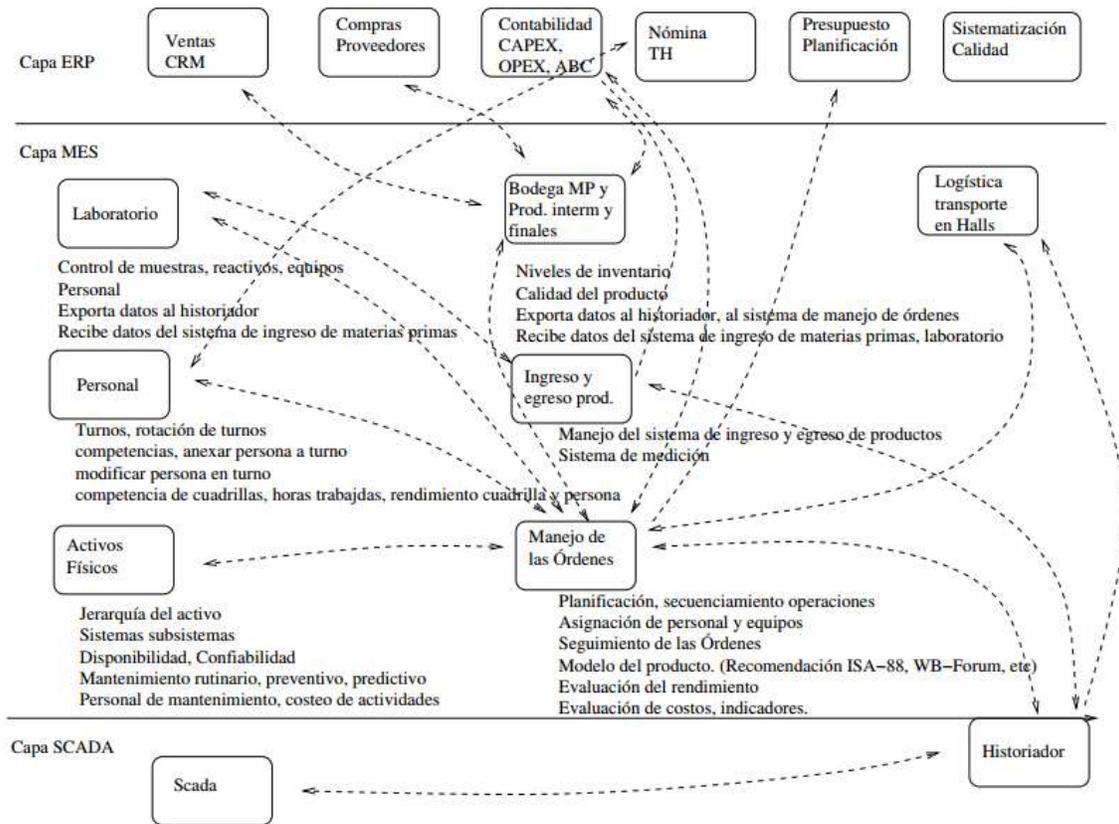


Figura 2. Lista de aplicaciones

Fuente: Cordero et al. (2016)

En base a la lista de aplicaciones del MES, el sistema se ha dividido en módulos cohesivos que encapsulan una funcionalidad muy específica e independiente de otros módulos. Entonces el siguiente paso es agregar los paquetes para los módulos en cada uno de ellos, los paquetes de tipo de cliente y en el de los de la lógica de negocio (contenedor). Por lo que el sistema informático se ha dividido en los módulos de: Gestión de la producción, Personal, Calidad, Gestión de bodega, Activos físicos, Modelo de planta (la tabla 1), muestra cómo serán organizados estos paquetes.

JSF DESKTOP	Gestion Produccion
	Personal
	Calidad
	Gestion Bodega
	Activos Físicos
	Modelo Planta
	Mantenimiento
	Reportes
	Estadísticas
	Costos
	Administración del sistema

Tabla 1. Los Subsistemas del sistema MES.

Elaborado: Sañay, Becerra, Calle (2018)

En base a la lista de aplicaciones del MES, el sistema se ha dividido en módulos cohesivos que encapsulan una funcionalidad muy específica e independiente de otros módulos. Entonces el siguiente paso es agregar los paquetes para los módulos en cada uno de ellos, los paquetes de tipo de cliente y en el de los de la lógica de negocio (contenedor). Por lo que el MES se ha dividido en los módulos de: gestión de la producción, personal, calidad, gestión de bodega, activos físicos, modelo de planta, la tabla 1 muestra cómo serán organizados estos paquetes.

Lo que resta es definir dónde se deben almacenar cada uno de los objetos que componen el sistema en base al Patrón MVC+S. Las clases que componen las vistas son almacenadas en la capa de presentación; el controlador en la de aplicación; y el modelo en la de servicios se muestra en la figura 3.

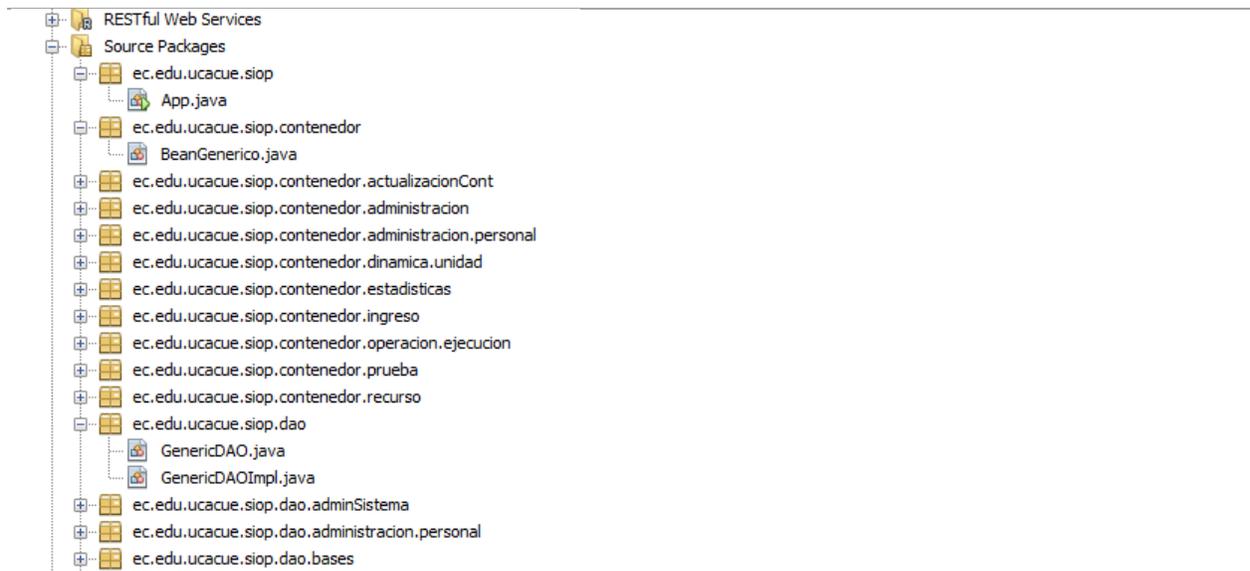


Figura 3. Paquetes de Sistema Informático

Elaborado: Sañay, Becerra, Calle (2018)

Diseño de interfaz gráfica (vista)

La interfaces que se implementan en el sistema deben ser intuitivas y adaptables a la capa de la vista de la arquitectura de desarrollo, por lo que se establece un flujo de trabajo estándar para que el manejo de las interfaces graficas sea sencillo, en base a esto se plantea un diseño en el cual la capa de la vista se divide en cinco sub capas las cuales están identificadas con los siguientes nombres index, crear, modificar, ver, eliminar; la vista se divide para facilitar el desarrollo y organización de la aplicación, pero al momento de ser visualizadas trabajan de forma integrada como si se tratase de una sola vista, este estándar es definido para formularios sencillos que solo requieren realizar procesos de creación, actualización y modificación, para procesos complejos el número en el que se divide la vista puede variar según la necesidad, la vistas heredan los componentes de la vista padre, esta vista está identificada con el nombre de

INDEX, el cual es el formulario principal de las capas y el que permite la comunicación entre todas las capas de la vista de manera intuitiva.

El index es la capa principal de cada formulario web, ya que este permite la agrupación de las demás capas, en esta capa se compone de un listado de información de la entidad a la que represente, y de botones que permiten la comunicación con las demás capas, en esta capa se lleva el control de los permisos que tienen cada usuario, especificando que procesos puede realizar el usuario implementando ese formulario, entre los permisos estándares que se controlan esta la creación, modificación, activación o desactivación de registros, la eliminación y el ver en detalle cada registro de la entidad a la que representa el formulario, en la figura 4 se indica las opciones que brinda esta capa en ella se aplican el CRUD (create: crear, read: leer, update: actualizar, delete: borrar) de los procesos del negocio en la aplicación.



Figura 4. Formulario Principal (Índex)

Elaborado: Sañay, Becerra, Calle (2018)

Módulos del sistema

Con el documento de especificación de la arquitectura de software para cubrir las funciones del Sistema de Gestión de Operaciones en el que se consideran los diferentes subsistemas MES, con el desarrollo de la arquitectura MVC+S se procedió a incorporar los subsistemas como el módulo de administración del sistema encargado de la integración de varias entidades como: submenú, menú, permiso, usuario, rol, persona, control, con este módulo se realiza el control de acceso al sistema, especificando la opciones o procesos que cada usuario puede realizar, el modulo se compone de varios formularios que permiten controlar y acceder al sistema de manera eficiente, también el registro de las acciones que son ejecutadas en el sistema, con el fin de establecer un proceso de auditoría que debe llevar todo sistema informático, en él se especifica el usuario que realiza la acción, la fecha y hora de la transacción, este proceso deberá gestionar automáticamente el sistema y no requiere ser activado por un usuario para ser ejecutado, en la figura 5 se muestra un diagrama UML, la interacción del usuario con el módulo de administración, y la forma en que los componentes interactúan entre sí para realizar un proceso.

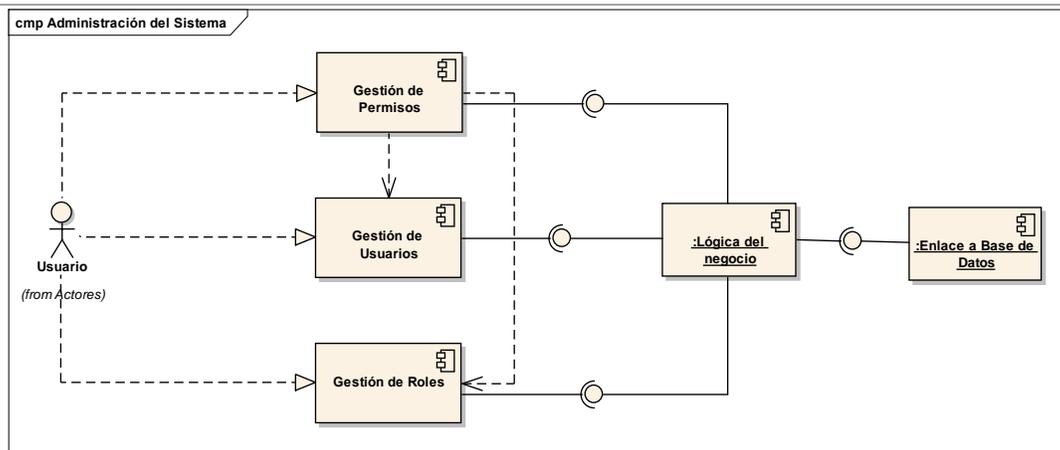


Figura 5. Componentes del Módulo de Administración

Elaborado: Sañay, Becerra, Calle (2018)

Para el acceso al sistema se especificó los métodos de acceso, a través del registro de usuarios los cuales van a trabajar bajo los roles que sean asignado, y tendrán acceso a ciertas opciones del sistema, como se observa la figura 5, para el usuario se ha de definir los roles (Quién) y establecer el conjunto de permisos (Acción) para que ejecute la acción en el documento de negocio (Dónde).

Con este sistema de conjuntos de permisos se facilita la asignación de permisos de los usuarios sobre los diferentes documentos y se pueden crear permisos de todos los tipos de forma muy flexible.

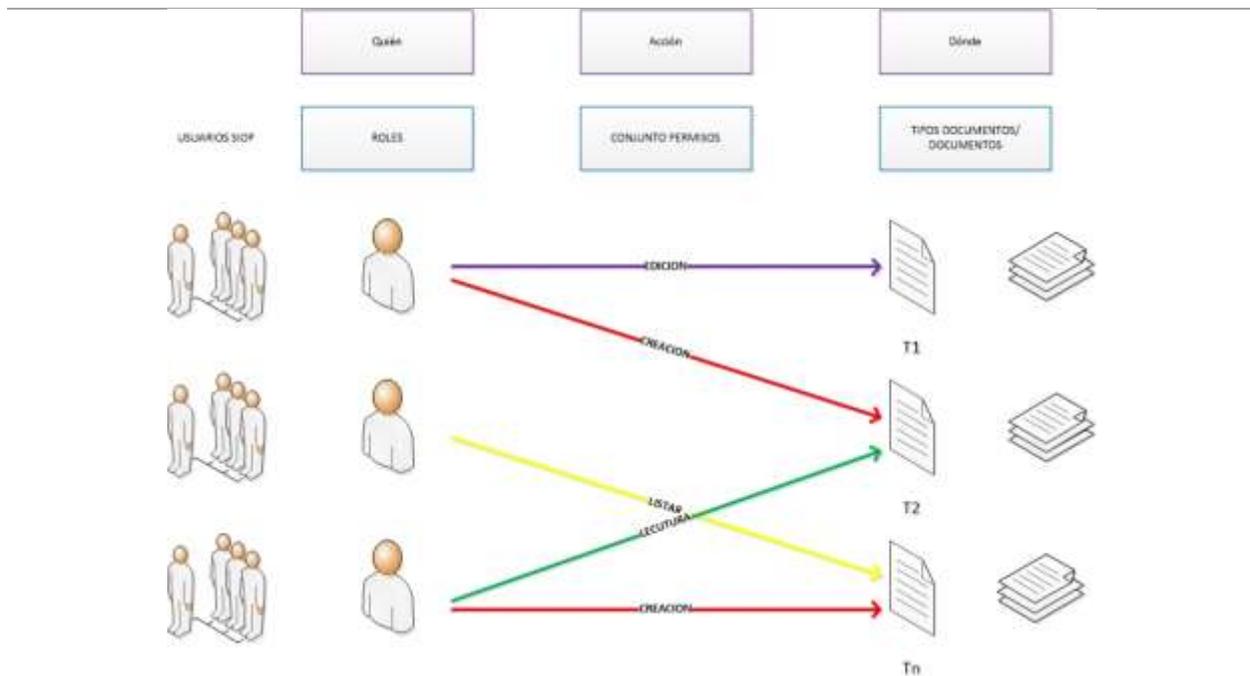


Figura 6. Configuración de usuarios, roles y permisos a documentos de negocio

Elaborado: Sañay, Becerra, Calle (2018)

Lo anterior permitió integrar el módulo de administración de la producción, en el que se implementaron las variables de calidad, variables de alimentación y variables de proceso.

Despliegue de la aplicación

Para que la aplicación funcione es necesario configurar el entorno de despliegue, esto es preparar el servidor de aplicación, para este caso se configuró el sistema operativo libre Linux CentOS tipo server, en el que se levantaron los: Servicio web con Apache Tomcat versión 7, servicio de base de datos MySQL, y el servicio SSH éste último para establecer una conexión remota con el servidor CentOS en caso de que se requiera dar mantenimiento al sistema.

Para publicar la aplicación fue necesario generar un fichero .war (guarda la información del proyecto) el mismo que se copia dentro del directorio .../webapp del Apache Tomcat.

Para su publicación es necesario levantar el servicio de Apache. Levantado el servicio la aplicación se despliega dentro del dominio y el puerto definido para la aplicación.

Conclusiones.

Los resultados en la etapa de desarrollo de la arquitectura y un prototipo informático del Sistema de Ejecución de la manufactura (MES) bajo el enfoque holónico para plantas de producción continua integra todas las funcionalidades esperadas para una planta de producción esto demuestra que es factible integrar en las PYMES.

El riesgo está asociado a la formación del personal en el modelado de los procesos productivos. Desde el punto de vista de desarrollo de software, la curva de aprendizaje fue lenta, pero una vez dominada, el desarrollo se aceleró el proceso.

El uso de herramientas libres en el ciclo de vida de desarrollo de software reduce el costo del producto, es por ello que la tecnológica empresarial en el manejo de los procesos contables y administrativos en las PYMES ayuda al crecimiento y desarrollo empresarial, ofreciendo ventajas competitivas frente a las demás empresas.

El estudio determino que es evidente la necesidad de investigar y promover modelos y procesos para desarrollar competencias de innovación y creatividad en los entornos empresariales.

Bibliografía.

- Chacon, E., & Cardillo, J. (2016). Models for planning and supervisory control for the feeding raw material in cement production.
- Chacon, E., Besembel, I., Rivero, D., & Cardillo, J. (2009). Embedded holonics systems in production process: holonic unit of production. *Revista Técnica*.
- Cordero, D., Chacon, E., Sanay, I., & Criollo, D. (2016). Modelo de Procesos para la Automatización del área de producción en el sector de la Industria cementera pública del Ecuador. *INGENIUS*, 51-63.
- EcuadorLegalOnline. (3 de Agosto de 2015). *EcuadorLegalonline*. Obtenido de <http://www.ecuadorlegalonline.com/sri/pymes/>
- Kendall, K., & Kendall, J. (2011). *Análisis y diseño de sistemas*. Naucalpan de Juárez, Estado de México: Prentice Hall.
- Pressman, R. (2010). *INGENIERÍA DEL SOFTWARE. UN ENFOQUE PRÁCTICO*. México: McGraw-Hill.
- Secretaría Nacional de Planificación y Desarrollo, Senplades. (2017). *Plan Nacional de Desarrollo 2017-2021.Toda una Vida*. Quito: Senplades.
- Sommerville, I. (2016). *Software engineering*. Pearson Education Limited.