

DISEÑO DE UN PROTOTIPO PARA LA PROGRAMACIÓN DETALLADA Y DESPACHO DE LA PRODUCCIÓN BASADO EN EL ESTÁNDAR ISA-95



DESIGN OF A PROTOTYPE SYSTEM FOR DETAILED PRODUCTION SCHEDULING AND DISPATCHING BASED ON ISA-95 STANDARD

AUTOR

ANDREA LÓPEZ AGUDELO
Ingeniera en Automática Industrial
*Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones, Departamento de Instrumentación y Control
anlopez@unicauca.edu.co
COLOMBIA

AUTOR

OSCAR AMAURY ROJAS ALVARADO
Ph.D. (c) en Ciencias Aplicadas
*Universidad del Cauca
Docente
Facultad de Ingeniería Electrónica y Telecomunicaciones, Departamento de Instrumentación y Control
orojas@unicauca.edu.co
COLOMBIA

AUTOR

JUAN DAVID MÉNDEZ ASTUDILLO
Ingeniero en Automática Industrial
*Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones, Departamento de Instrumentación y Control
juanmendez@unicauca.edu.co
COLOMBIA

AUTOR

EDGAR FABIÁN RUANO DAZA
Maestría (c) en Computación
*Universidad del Cauca
Estudiante
Facultad de Ingeniería Electrónica y Telecomunicaciones, Departamento de Sistemas
eruano@unicauca.edu.co
COLOMBIA

*INSTITUCIÓN

Universidad del Cauca
UNICAUCA
Universidad Pública de Educación Superior
Carrera 2 # 4N-140, Popayán – Cauca
rectoria@unicauca.edu.co
COLOMBIA

INFORMACIÓN DE LA INVESTIGACIÓN O DEL PROYECTO: Proyecto de grado para optar al Título de Ingenieros en Automática Industrial - UNICAUCA - 2014.

RECEPCIÓN: Marzo 06 de 2014

ACEPTACIÓN: Junio 25 de 2014

TEMÁTICA: Ingeniería del Software

TIPO DE ARTÍCULO: Artículo de investigación científica y tecnológica

Forma de citar: López Agudelo, A., Méndez Astudillo, J. D., Rojas Alvarado, O. A. & Ruano Daza, E. F. (2014). Diseño de un prototipo para la programación detallada y despacho de la producción basado en el estándar ISA-95. En R, Llamosa Villalba (Ed.). Revista Gerencia Tecnológica Informática, 13(37), 81-95. ISSN 1657-8236.

RESUMEN ANALÍTICO

El presente artículo evidencia el proceso de análisis y desarrollo de un prototipo software que sirve como herramienta de apoyo en el proceso de toma de decisiones para la programación detallada y despacho de la producción de una planta de fabricación que opera en modo campañas batch, es decir que su programación se realiza buscando mantener un conjunto de lotes (batches) que pertenecen una misma familia de producto consecutivamente sobre la misma unidad de producción con el fin de minimizar tiempos de limpieza y configuraciones. El prototipo toma como base para su diseño y especificación los estándares internacionales de automatización ISA-95.03, ISA-95.02 que definen de manera estandarizada entidades, su contenido y flujos de información necesarios para realizar las tareas correspondientes a las actividades de administración de operaciones de manufactura, e ISA-88.01 que da las pautas para la especificación del proceso batch, tomando las definiciones de éste para el modelado del proceso caso de estudio escogido y la posterior verificación y validación de las funcionalidades del proyecto (Proceso de producción de pintura). Partiendo de esto se realiza un análisis funcional con base a las actividades de programación detallada y despacho de la producción especificadas en el estándar. La funcionalidad implementada resalta la integración vertical entre los niveles de la empresa a partir de la recepción de un programa de producción editado bajo el esquema de integración definido en los estándares generado desde la planeación y convertido por el sistema a campañas batch que son despachadas en una lista de batches hacia un administrador batch que se encarga de ejecutarlo en planta.

PALABRAS CLAVES: B2MML, BATCHML, Campaña Batch, Despacho de la Producción, ISA-95, ISA-88, Merge, Programación de la Producción, Programación Detallada de la Producción, Split.

ANALYTICAL SUMMARY

This article shows the analysis and development process of a prototype software that serves like a support tool for the decision making process for detailed production scheduling and dispatching of a manufacturing plant operated in batch campaigns mode, it means that the scheduling is made looking for maintain a set of batches that belong to the same product family consecutively on the same production unit in order to minimize cleaning and configuration times. The prototype takes as its basis for its design and specification the international automation standards ISA-95.03, ISA-95.02 which define entities in a standardized way, its content and information flow to perform the tasks for manufacturing operations management and ISA-88.01 which gives guidelines for batch processes definition, taking this information for modeling the chosen case study process for the verification and validation of the functionalities of project (Paint production process). Based in this information a functional analysis is performed based to the activities of detailed production scheduling and dispatching specified in the standard. The implemented functionality highlights the vertical integration between enterprise levels beginning with the reception of a production schedule made under the standardized defined integration schema generated from enterprise planning and converted by the prototype to batch campaigns that are dispatched in a batch list to a batch administrator system that ensures the executions of the schedule in floor plant.

KEYWORDS: B2MML, BATCHML, Batch Campaign, Detailed Production Scheduling, ISA-95, ISA-88, Merge, Production Dispatching, Production Scheduling, Split.

INTRODUCCIÓN.

Al tener en cuenta el continuo cambio en el entorno actual de la sociedad, donde existe una rápida introducción de productos en el mercado y una constante búsqueda de innovación en materia tecnológica, las industrias procuran mantenerse activas mejorando continuamente sus productos, procesos y servicios enfrentándose a un medio exigente y con normatividades cada vez más rigurosas, factores que han desencadenado en el incremento de la demanda de sistemas de gestión automatizados y el número de investigaciones buscando resolver falencias de la industria manufacturera actual.

Teniendo en cuenta la importancia que tienen las tareas de programación y despacho de la producción dentro de las empresas de manufactura, es necesario que existan esfuerzos en proveer soluciones tecnológicas que permitan contar con sistemas de apoyo en la toma de decisiones en el proceso de programación y control de la producción batch que sean compatibles y se basen en las funcionalidades y definiciones de los estándares internacionales vigentes, de tal forma que se obtenga una buena programación de la producción y un seguimiento continuo de la misma, ya que con esto las compañías lograrían corregir a tiempo posibles errores, aprovechar al máximo los recursos y el tiempo con el que cuentan obteniendo clientes satisfechos que atraerán a otros, contribuyendo a alargar su estancia en el mercado competitivo. [1]

A partir de lo anterior, el objetivo del prototipo gestor de campañas batch no es optimizar la programación de la producción como se muestra en [1, 2, 3] si no por el contrario es apoyar el proceso de toma de decisiones en esta temática y de esta manera lograr una reducción de tiempos de alistamiento y configuración, disminución de recambios, limpiezas y aprovechamiento de la capacidad del equipo. Esto es posible lograrlo por medio del manejo de la programación detallada y despacho de la producción, a partir de la recepción de un programa de producción generado desde la vista gerencial de planeación, logística y negocios el cual contiene ordenes de producción, que se convertirán en campañas batch que serán asignadas a los recursos disponibles listas para ser ejecutadas.

1. ANALISIS FUNCIONAL DEL PROTOTIPO.

En el estado del arte no son pocas las referencias sobre los problemas asociados a las funciones de planificación, programación y control de la producción desde un enfoque clásico, los cuales aunque con diferentes criterios respecto a sus conceptualización de integración de los procesos, han permitido concluir que los aspectos a tener en cuenta en el proceso de

planificación y control de la producción corresponden a planificación estratégica o a largo plazo, planificación agregada o mediano plazo, programación maestra de producción (MPS), programación de componentes, y finalmente ejecución y control, tal como se muestra en la Figura 1. [4, 5, 6]

Aunque los aportes de la teoría clásica estén posicionados y sean concisos, la forma de abordar y aplicar estas tareas en el entorno empresarial ha sido compleja debido a la poca claridad en la delimitación de las tareas pertenecientes a cada una de las actividades, haciendo que se traslapen responsabilidades como se evidencia en [7], en donde, en la práctica no es clara la separación entre un planificador, de un programador o un programador de un despachador.

FIGURA 1. Jerarquía del Proceso de Planificación, Programación y Control de la Producción desde la teoría clásica.



Con el fin de unificar los diferentes conceptos trabajados en las referencias anteriormente mencionadas y de considerar la experiencia obtenida del desarrollo de proyectos de industrias como Honeywell, Foxboro, Yokogawa, Fisher Rosemount, Chevron entre otras [8]. ISA (International Society of Automation) decide proponer una serie de documentos estándar para el manejo de información generalizado. El resultado de esta iniciativa son los estándares ISA-88 e ISA-95 que brindan una terminología y un modelado estándar con un enfoque jerárquico aplicable a las industrias.

En la Figura 2 se presenta el modelo jerárquico que propone ISA-95 o la comúnmente llamada pirámide

de automatización, donde se evidencia la división de la empresa en niveles en forma vertical, los cuales presentan una serie de aspectos claves que sirven para diferenciar cada uno de dichos niveles según el horizonte de tiempo, el proceso de toma de decisiones y el contexto.

FIGURA 2. Jerarquía Funcional de Actividades Multi-Nivel.



Además de proporcionar una vista lógica de la empresa y de especificar y detallar cada una de las actividades a realizar en los niveles que lo conforman, el estándar propone una jerarquía de programación en su esfuerzo de unificación de conceptos, la cual se presenta en la Figura 3.

Para cumplir con la terminología, modelos y definiciones que actualmente se encuentran estandarizadas, el enfoque de conceptualización y caracterización del prototipo va orientado hacia las actividades ubicadas en el nivel 3 de programación detallada y despacho de la producción, tomando como base las tareas definidas para estas actividades en el estándar ISA-95.03.

Adicionalmente a estos conceptos, la parte 3 del estándar ISA-95 [9] presenta en detalle el modelo genérico de actividades de administración de operaciones de producción, el cual fue tomado como base para realizar el análisis funcional del prototipo, teniéndose como resultado la relación de actividades asociadas al proceso de desarrollo software mostrada en la Tabla 1 y Tabla 2.

FIGURA 3. Jerarquía de Programación desde ISA-95.



TABLA 1. Funcionalidades de la Programación detallada según ISA-95.03 asignadas al prototipo.

TAREAS	PROTOTIPO
Crear y mantener un programa detallado de la producción.	X
Comparar la producción actual con la producción planeada	Función de Nivel 4
Ejecución de las simulaciones tipo 'Que pasa sí'.	No Aplica
Obtener información de las operaciones de mantenimiento, calidad, e inventario.	No Aplica
Determinar la capacidad comprometida de cada recurso para el uso de la función de administración de los recursos de producción.	X
Emitir órdenes de trabajo de producción como fueron definidas por el programa.	X
Asignación de recursos locales para la producción, cuando estas no son identificadas como parte del programa detallado de la producción.	Usuario
Liberar recursos locales para comenzar órdenes de trabajo.	Usuario
Manejo de las condiciones no anticipadas en programación detallada de la producción	No Aplica
Mantener el estado de las órdenes de trabajo. Ejemplo: Aprobado, Corriendo, Pausado o cancelado	X
Asegurar que las limitaciones del proceso y las órdenes bajo el nivel de detalle de la programación detallada son logradas en producción. Esto toma lugar después de que el programa es creado, pero antes de que sus elementos son ejecutados.	X
Informar a la programación detallada cuando un evento no anticipado resulta en la inhabilidad de lograr los requerimientos de programación.	X
Recibir información desde la admón. operaciones de calidad que indica condiciones no anticipadas que se pueden relacionar con eventos del programa.	No Aplica
Recibir información desde la administración de recursos de producción sobre disponibilidad futuro de recursos que se pueden relacionar con eventos del programa.	No Aplica
El envío o puesta a disposición de la lista de despacho producción especificando las actividades productivas que se realizan	X

2. ESCOGENCIA DE LA METODOLOGÍA DE DESARROLLO.

Teniendo en cuenta que la tasa de éxito de los proyectos de desarrollo software puede ser incrementada según la correcta escogencia de una metodología que se adapte a las características específicas de cada proyecto, la adopción de estas por parte de los desarrolladores en los últimos años se ha convertido en un paso indispensable para abordar un proyecto.

Para el desarrollo del proyecto se tienen en cuenta 2 metodologías de desarrollo de las metodologías "clásicas o peso pesado" y 2 de las "suaves o ágiles" [10], cada clasificación de la siguiente manera:

TABLA 2. Metodologías de desarrollo opcionales para el proyecto.

TIPO	METODOLOGÍA
CLÁSICA	RUP
	CRAIG LARMAN
ÁGIL	SCRUM
	Extreme Programming (XP)

Basándose en la información presentada en las referencias [11 - 16] y la investigación realizada en [10], donde se analizan diferentes puntos claves en la selección de una metodología para el desarrollo de software se muestran 10 factores que afectan la selección

y además, se añaden otros 2 factores importantes incluidos por las características propias del proyecto los cuales corresponden a la documentación detallada del proceso de análisis y diseño, y la información ilustrativa sobre la aplicación de la metodología. Estos criterios fueron analizados detalladamente con referencia a los requerimientos anteriores y se procedió a realizar una ponderación cuantitativa en correspondencia a las metodologías escogidas, obteniendo como resultado que la metodología que se adapta a las características de desarrollo del prototipo es la propuesta por Craig Larman.

2.1 APLICACIÓN DE LA METODOLOGÍA SELECCIONADA.

De acuerdo al resultado de la aplicación de criterios de selección realizada anteriormente se decide escoger como metodología a seguir la propuesta por Craig Larman en su libro "UML y Patrones, Introducción al análisis y diseño orientado a objetos", [11]. A continuación se presenta el proceso de aplicación de la metodología en el diseño y especificación del sistema desde el inicio de las actividades de planeación hasta terminar todos los ciclos de desarrollo. La metodología está compuesta de 3 etapas de macronivel, la etapa de planeación, de construcción y la de aplicación, en esta última es en la que se realiza el paso a producción (Funcionamiento real de la aplicación), teniendo en cuenta que es un sistema prototipo no es abordada esta etapa en éste artículo.

2.1.1 Planeación.

La primera etapa está conformada por 7 artefactos que permiten la identificación de los requerimientos funcionales relevantes del sistema y la definición de su estructura por parte de algunos modelos iniciales, sin entrar a especificar detalles de desarrollo o tecnología.

2.1.1.1 Definición de requerimientos.

Para la definición de los requerimientos del sistema se hace uso del análisis funcional obtenido de la interacción con el cliente y el realizado bajo el estándar ISA-95.03 donde se concluye que los requerimientos funcionales del sistema son:

- El sistema deberá soportar el intercambio de información estandarizado para la recepción del programa de producción editado en xml bajo el formato *B2MML (Business To Manufacturing Markup Language)* para el acceso a requerimientos de producción.(R1.1)
- El sistema generará las listas de órdenes de trabajo que se asimila al *BatchList* y la solución deberá

proveer las capacidades de despacho (integración) al sistema FT batch. (R1.2)

- La solución deberá contemplar el diseño e implementación de software bajo las mejoras prácticas que realice las funcionalidades de programación de la producción y posibilite integrar órdenes de trabajo y la consolidación de información de ejecución de producción desde y hacia el FT batch.(R1.3)
- El sistema deberá implementar y garantizar la integridad transaccional de datos.(R1.4)
- Almacenar información relevante del proceso.(R1.5)
- Permitir configurar la información del proceso. (R1.6)
- Realizar acciones tipo Split & Merge.(R1.7)
- La solución deberá ser desarrollada con las tecnologías de Microsoft, Visual Studio.Net, ASP.Net y SQL Server.(R1.8)
- Crear y mantener un programa detallado de producción.(R2.1)
- Determinar la capacidad comprometida de cada recurso para el uso de la función de administración de los recursos de producción.(R2.2)
- Recibir el Programa de Producción proveniente del nivel 4.(R2.3)
- Emitir órdenes de trabajo de producción como fueron definidas por el programa.(R3.1)
- Liberar recursos locales para comenzar órdenes de trabajo.(R3.2)
- Mantener el estado de las órdenes de trabajo. Ejemplo: aprobado, fijo, en proceso o cancelado. (R3.3)
- Asegurar que las limitaciones del proceso y las órdenes bajo el nivel de detalle de la programación detallada son logradas en producción. Esto toma lugar después de que el programa es creado, pero antes de que sus elementos son ejecutados. (R3.4)
- Informar a la programación detallada cuando un evento no anticipado resulta en la inhabilidad de lograr los requerimientos de programación.(R3.5)
- Enviar o poner a disposición la lista de despacho de producción especificando las actividades productivas que se realizan.(R3.6)

2.1.1.2 Casos de uso de alto nivel.

En los casos de uso de alto nivel es necesario definir una primera aproximación de lo que será el proyecto permitiendo determinar el contexto en el que se desenvuelve (frontera), los actores (Operario, Administrador Batch) con los que interactúa y el grado de complejidad de este de una manera muy general sin entrar en consideraciones importantes de diseño, la idea es poder describir el proceso de una forma general sin tener en cuenta especificaciones tecnológicas, las cuales se identifican más adelante en el proceso

de construcción. Los casos de uso considerados en el análisis del prototipo son:

- Inicio de Sesión.

El usuario realiza un inicio de sesión en el sistema, con el fin de que solo los supervisores configurados en el sistema sean los que acceden a las funcionalidades de este.

- Configuración Inicial.

El usuario se encarga de configurar la información inicial necesaria para el correcto funcionamiento del sistema, como los productos que la empresa fabrica, los equipos presentes en esta y los segmentos de proceso que se manejan en la planta.

- Recibir Programa de Producción.

El supervisor busca y selecciona el programa de producción enviado desde el sistema ERP el cual es interpretado por el sistema

- Generar Campañas Batch.

El sistema de la mano del Supervisor propone un programa detallado basado en campañas batch, el Supervisor decide si el resultado final es aceptado.

- Generar Manualmente Campañas Batch.

El supervisor se encarga de generar de forma manual y según su criterio las campañas que no fueron aprobadas.

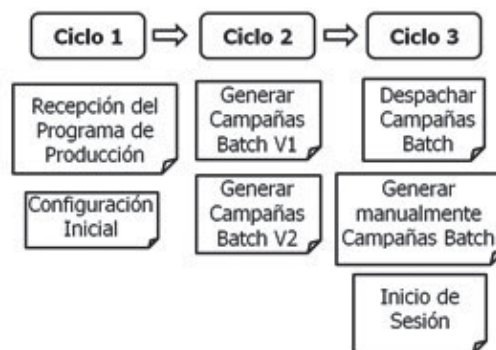
- Despachar Campañas Batch.

El Supervisor envía hacia el administrador batch el programa detallado aprobado y este lo recibe y lo ejecuta como fue definido. Además, el administrador Batch se encarga de enviar información referente al estado de los equipos en el proceso y la ejecución de las campañas.

Es necesario realizar una clasificación de los casos de uso en los diferentes ciclos de desarrollo a abordar, ya que permite identificar el orden adecuado en el cual se deben atacar dichos casos a lo largo de los ciclos de desarrollo que se definan, con el fin de abordar primero los de alto rango que influyen fuertemente en el correcto desarrollo del sistema.

Después de la realización de la clasificación de los casos de uso generados se asignan estos a los ciclos de desarrollo según la calificación obtenida, como se muestra en la siguiente figura.

FIGURA 4. Asignación de los casos de uso a los ciclos de desarrollo definidos para el proyecto.



2.1.1.3 Casos esenciales de uso.

Los casos esenciales de uso describen de forma general lo que ocurre en la realidad mostrando las posibles rutas de eventos que pueden suceder en el funcionamiento básico del sistema. A continuación se describe brevemente el resumen y la estructura del caso de uso esencial para el Despacho de campañas batch.

Es importante tener claro desde el principio del caso el actor o actores que intervienen en este, evidenciándolos e indicando el que tiene el rol de iniciador (Administrador Batch, Supervisor - Iniciador), además evidenciar el propósito y la idea central de éste el cual sirve como una información general del funcionamiento básico, para el caso actual es, la creación de la lista de despacho y el envío de esta hacia el administrador batch.

Otro punto importante es evidenciar la relación existente entre el caso y los requerimientos funcionales planteados inicialmente ya que sirve para determinar la contribución del caso de uso en lograr estos. La relación existente de este caso con los requerimientos funcionales se muestra haciendo énfasis con el identificador de los requerimientos R1.2, R1.3, R1.4, R1.5, R1.8, R3.1, R3.2, R3.3, R3.4, R3.5, R3.6.

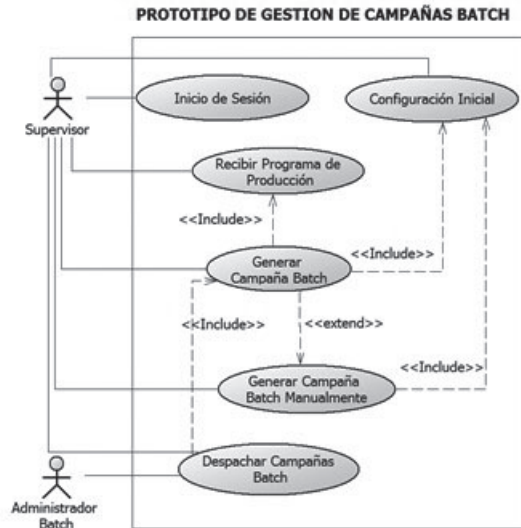
El siguiente paso a mostrar en el caso de uso es el curso normal de los eventos indicando paso a paso el desarrollo del caso y si existen, los diferentes cursos alternos posibles.

2.1.1.4 Diagramas de casos de uso.

El diagrama de casos de uso permite evidenciar las interacciones generales que tienen tanto los actores del sistema con los casos de uso definidos, al igual que la interacción entre los mismos casos de uso en el sistema.

Dicho diagrama es desarrollado tomando como base la información definida en los 2 puntos anteriores (Casos de uso de Alto Nivel y Esenciales)

FIGURA 5. Diagrama de casos de uso del sistema.



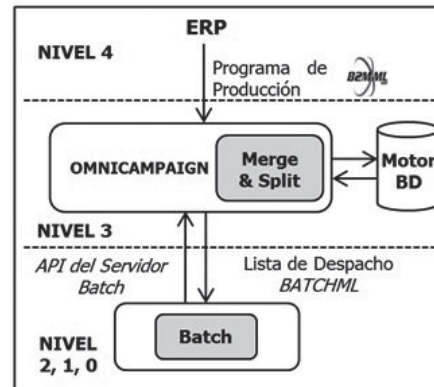
2.1.1.5 Arquitectura del sistema.

En esta etapa se define la estructura macro del sistema como también la estructura interna del mismo, este proceso es importante en el diseño de sistemas de información ya que se define la interacción del sistema con componentes externos siendo más específicos los detalles de tecnología, teniendo en cuenta las interfaces necesarias para un correcto funcionamiento, definiendo y restringiendo fuertemente partes de las etapas de diseño siguientes y la evolución de las mismas.

2.1.1.5.1 Arquitectura de integración del prototipo en la empresa.

Como se evidenció en la especificación de requerimientos y análisis funcional del prototipo en base al estándar ISA-95.03 y además teniendo en cuenta que las actividades que desarrolla esté son la programación detallada y el despacho de la producción, tareas pertenecientes al nivel de ejecución de la manufactura (Nivel 3) se plantea la arquitectura mostrada en la figura 6.

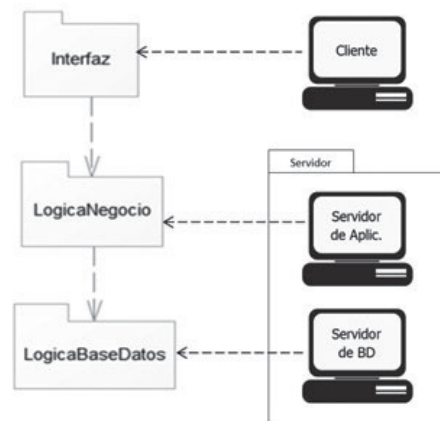
FIGURA 6. Arquitectura general de la empresa.



2.1.1.5.2 Arquitectura definida de la aplicación web.

Para el desarrollo del sistema prototipo de gestión de campañas batch, se emplea la arquitectura en 3 capas tomando como referencia el análisis comparativo con otras arquitecturas y las ventajas y desventajas plasmadas en [17]. La arquitectura de 3 capas busca que cada capa trabaje en un área específica como lo son la interfaz con el usuario, la lógica de la aplicación y la forma de almacenamiento de datos, brindando flexibilidad y robustez que a su vez facilitan la escalabilidad y mantenibilidad del sistema sin afectar ninguno de los módulos.

FIGURA 7. Arquitectura de 3 capas.



Al mantener la separación de la capa de presentación y la capa de negocios se facilita la flexibilidad en la ejecución de cambios de la interfaz, ya que al ser independientes, los cambios realizados en esta no afectan directamente la lógica de negocio y además permite una mayor flexibilidad en el llamado de la capa intermedia desde la capa de presentación. Adicionalmente se puede presentar mayor facilidad para la implementación y de nuevos clientes del sistema en otros dispositivos que se deseen vincular al proyecto [17].

Siendo coherentes con la arquitectura seleccionada y las ventajas deseadas, dentro de los patrones de arquitectura que permitirán dar la base o cuerpo del desarrollo software del prototipo se ha utilizado el patrón MVC (*Modelo Vista Controlador*) [18]. El modelo contiene la lógica a efectuarse según los eventos recibidos por el sistema, teniendo por esta razón una relación directa con el nivel de lógica de negocio; La vista se encarga de controlar la presentación de la información, de mostrar las páginas web necesarias para la interacción con el usuario, relacionándola con la capa de presentación; el controlador es el responsable de descifrar los eventos del sistema ocasionados por el usuario y de informarle éstos al modelo, por lo cual se encuentra ubicado entre la capa de interfaz y la lógica de negocio [19].

2.1.1.6 Modelo conceptual.

El modelo conceptual es una de las etapas más importantes del proceso de análisis, ya que permite obtener una representación de los conceptos del mundo real aplicados en el dominio del problema [11]. La idea es abstraer la mayor cantidad de información del problema mostrando los conceptos asociados a este, los atributos y las asociaciones que estos forman.

Se ponen en práctica dos tipos de técnicas en la identificación de conceptos planteadas en la metodología a continuación. La primera consiste en identificar los conceptos según una lista definida que contiene aspectos claves en el desarrollo de sistemas y la segunda se basa en los casos de uso planteados, además de esto es importante resaltar que se tienen en cuenta las especificaciones del estándar ISA-95.02 para la definición de los atributos. Los conceptos identificados para el caso de uso Despachar Campañas Batch son:

- Administrador Batch
- Equipo
- Producto
- Supervisor
- Lista de Despacho
- Programa Detallado de Producción

Una vez identificados los objetos para cada uno de los casos de uso se obtiene el modelo conceptual definido para el desarrollo del prototipo.

Dentro de los atributos que se definen en los objetos identificados para el desarrollo del modelo conceptual han sido considerados los plasmados en el estándar ISA-95.02 mostrados en [20], además de los requeridos para el debido funcionamiento del prototipo.

2.1.1.7 Esquema de la base de datos.

Con el fin de almacenar la información necesaria resultado del análisis y desarrollo de las funcionalidades del prototipo de gestión de campañas batch, se ve la necesidad de crear una base de datos que se irá completando a lo largo de la implementación de los ciclos de desarrollo, para la cual se redacta un documento donde se plantea un problema inicial con respecto a las necesidades de la base de datos esbozando las posibles entidades a involucrarse en el proceso con sus respectivos atributos y relaciones, obteniendo así como primera medida una base para la creación de los modelos entidad relación y relacional que son llevados hasta su tercera forma normal y de aquí generar la base de datos a ser utilizada.

2.1.2 Construcción.

En la etapa de construcción son desarrollados una serie de artefactos teniendo en cuenta la clasificación realizada de cada caso de uso a los respectivos ciclos de desarrollo (Ver FIGURA 4) que permitirán paulatinamente el perfeccionamiento del sistema, estos ciclos se encuentran conformados por las etapas y artefactos mostrados a continuación.

2.1.2.1 Análisis.

En el análisis se pretende entrar en más detalles con respecto a labores y relaciones entre actores, conceptos y sistema que conlleven al cumplimiento de los requerimientos y funciones principales del sistema.

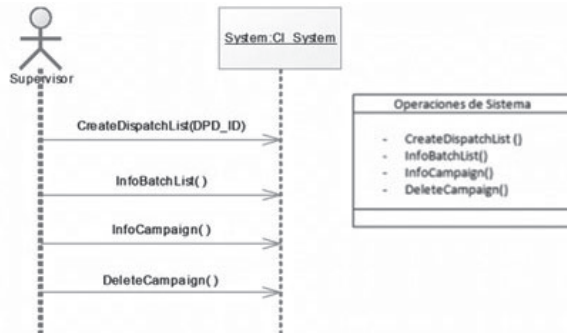
2.1.2.1.1 Diagramas de secuencia.

Los artefactos que se utilizan como base para este modelo son los diagramas de caso de uso definidos anteriormente, específicamente la sección del curso normal de los eventos. El diagrama muestra los eventos externos del sistema los cuales son generados por

los actores, estos eventos generan operaciones del sistema las cuales se registran en una tabla aparte que serán detalladas posteriormente por los contratos de operaciones.

A continuación se muestra el diagrama de secuencia correspondiente al caso de uso despachar campañas batch

FIGURA 8. Diagrama de secuencia del caso de uso despacho de campañas batch



2.1.2.1.2 Contratos de operaciones.

En esta etapa se define un contrato para cada operación evidenciada dentro del diagrama de secuencias de los casos de uso. Los contratos muestran el resultado de aplicar una operación en el sistema.

2.1.2.2 Diseño.

En la fase del diseño se busca la realización de un análisis dinámico en el cual se evidencia como los conceptos y entidades se comunican entre sí por medio de mensajes con tareas particulares a cada uno de tal forma que se llegue al cumplimiento de los objetivos específicos del sistema.

2.1.2.2.1 Diagramas de interacción.

El diagrama de Interacción permite no solo mostrar que se hace sino también el cómo, es decir, que es posible ver como los objetos interactúan con el fin de dar cumplimiento a sus respectivos objetivos, mediante dos tipos de diagramas, los de colaboración o los de

secuencias. Particularmente para el modelado del prototipo se hace uso de los diagramas de secuencia.

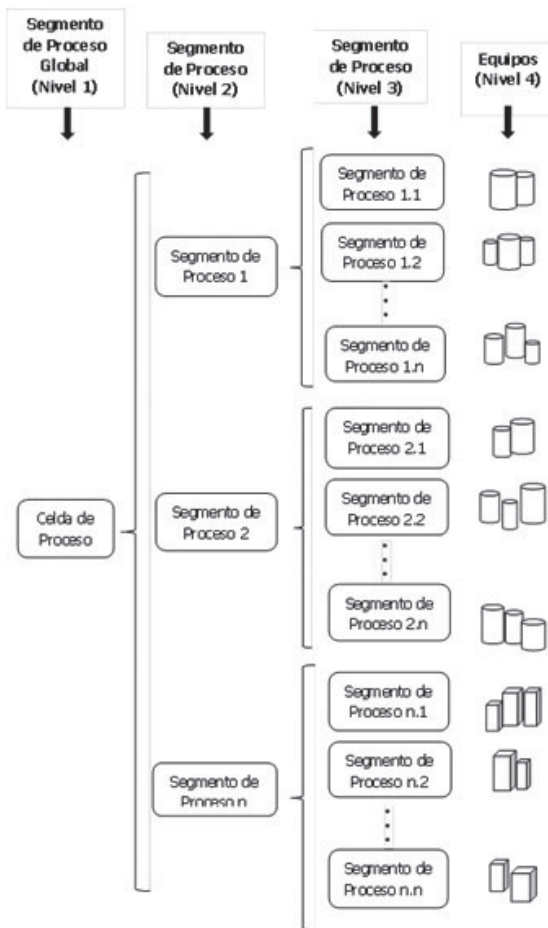
2.1.2.2.2 Diagramas de diseño de clases.

El diagrama de clases muestra la estructura del sistema tal cual como se va a implementar, evidenciando los atributos que las clases contienen, los métodos asociados y las relaciones estructurales entre éstas.

3. TOPOLOGÍA DEL PROCESO, VALIDACIÓN Y VERIFICACIÓN DEL PROTOTIPO.

Para el desarrollo de cualquier tipo de solución que busque resolver problemas de programación de la producción batch, es de vital importancia definir el modelo del proceso (topología, características, restricciones, etc.) que se desea programar, ya que una pequeña consideración puede desencadenar en un gran cambio en el análisis y diseño de éste.

Para el desarrollo del prototipo fue necesario definir un modelo de proceso genérico que trata de abarcar la mayor cantidad de aspectos aplicables en los problemas de programación de procesos Batch. Dentro del análisis realizado y a partir del modelo de segmento de proceso definido en el estándar ISA-95.01 e ISA-95.02 y los conceptos planteados de celda de proceso, unidad de producción entre otros en el estándar ISA-88.01 [21], se ha definido una topología del proceso para el sistema prototipo que consta de tres niveles de segmentos de proceso especificados así: un segmento de proceso global o de primer nivel que hace referencia a toda la celda de proceso, cuyos requerimientos vienen dados en unidades empacadas de producto terminado, segmento que contiene a los segmentos de segundo nivel; los que se asignan teniendo en cuenta las etapas del proceso, las cuales son identificadas por realizar una actividad de procesamiento principal, los requerimientos asociados a estos vienen en unidades de productos intermedios, estos segmentos a su vez contienen los del tercer nivel; que se presentan con el fin de brindar flexibilidad en la programación y da la oportunidad de atender procesos con topologías más complejas que dispongan de segmentos de proceso en paralelo los cuales realicen la misma actividad de procesamiento y que a su vez estarían compuestos de las unidades físicas de procesamiento o equipo en el piso de planta. En la FIGURA 9 se muestra el modelo del proceso planteado.

FIGURA 9. Topología general del proceso.

Para poder llevar a cabo un análisis sobre lo desarrollado con respecto a lo requerido se ha recurrido a una serie de herramientas conocidas como los procesos de Verificación y Validación (V&V) que tienen como objetivo determinar si el software es útil en una situación operativa además de mejorar la calidad por medio de la detección de fallos y omisiones durante todo el ciclo de vida del desarrollo de software [22], permitiendo así corregir en el momento oportuno posibles situaciones que podrían llevar al incumplimiento de los requerimientos o a funcionalidades no adecuadas.

En los métodos estáticos de V&V utilizados en la validación del prototipo, se encuentran como herramienta las inspecciones de software que consta de las etapas planteadas en [23].

Para el análisis dinámico del prototipo se encuentran como herramienta las pruebas de software, que consisten en definir un entorno controlado para que cuando el sistema sea utilizado en presencia de una

serie de entradas predefinidas genere una serie de salidas esperadas. Para realizar una prueba efectiva es necesario tener en cuenta una serie de pasos que sirven de guía en el proceso de aplicación de una prueba, los cuales fueron adaptados al proceso de V&V de [24].

Teniendo claro el proceso a seguir en el momento de definir y ejecutar una prueba es necesario especificar el tipo de prueba a realizar ya que según su clasificación puede variar la forma de proceder. Las pruebas de software se pueden clasificar como pruebas de caja blanca y de caja negra. Las pruebas de caja blanca son aquellas en las cuales se busca diseñar la prueba teniendo en cuenta el comportamiento interno y la estructura del código fuente a diferencia de las de caja negra que únicamente se conocen previamente las salidas esperadas al aplicar unas entradas determinadas y según este resultado se evalúa si el funcionamiento es el indicado.

Por último, se adopta la estrategia de prueba plasmada en [23] que permite evaluar el software empezando por los componentes más simples y seguir avanzando progresivamente hasta lograr probar completamente y en conjunto el software.

Es importante considerar que tanto los procesos de verificación y validación se efectuaron a lo largo de los ciclos de desarrollo definidos en el proyecto, además teniendo en cuenta que al finalizar los ciclos se obtienen prototipos software funcionales; característica que permite realizar tanto el análisis estático como el dinámico brindando la capacidad de identificar los fallos, defectos y omisiones, disminuyendo así los costos de las correcciones, al ser desarrollado estratégicamente en la etapa de sincronización de artefactos planteada en el mapa de la metodología utilizada.

A continuación se presentan un caso de prueba definido para los procesos de V&V planteados y aplicados al prototipo software para el caso de uso despachar campañas batch.

3.1 Análisis Estático.

Durante la finalización de la implementación de las funcionalidades del prototipo planteadas en cada ciclo de desarrollo se adaptaron una serie de preguntas basadas en [23] las cuales fueron utilizadas para todos los casos de uso, teniendo en cuenta preguntas tácticas con el fin de identificar defectos referentes al diseño, la lógica de programación, variables utilizadas, interfaces implementadas entre otras y en caso de encontrarlas se realizan las correcciones necesarias en cada ciclo de desarrollo.

Como resultado del proceso de V&V utilizado en el análisis estático, se obtuvieron inspecciones basados en listas de chequeo que facilitaron el entendimiento del artefacto a la persona encargada de realizar la inspección.

3.2 Análisis Dinámico.

En el proceso de análisis dinámico del prototipo se realizó la aplicación de las pruebas de caja blanca y de caja negra especificadas anteriormente para los diferentes casos de uso en los ciclos de desarrollo definidos.

Para las pruebas de caja blanca se utilizó la técnica de cobertura de sentencias que consiste en escribir tantos casos de prueba como sean necesarios con el objetivo de que se ejecute por lo menos una vez cada sentencia del código.

3.2.1 Pruebas de Caja Blanca.

Las prueba de caja blanca se realizaron sobre la marcha una vez se iban obteniendo los productos parciales ejecutables, recorriendo las sentencias de código una a una verificando los valores de las variables, tipos de datos, funciones, eventos, condicionales, bucles, detectando los fallos con ayuda del depurador de código presente en la herramienta de programación utilizada y corrigiendo estos para obtener el funcionamiento deseado.

3.2.2 Pruebas de Caja Negra.

En la aplicación de las pruebas de tipo caja negra se hace uso de la estrategia de prueba planteada en [24] para cada caso de uso en los ciclos de desarrollo definidos. A continuación se muestra la aplicación de la prueba al caso de uso Despachar las Campañas Batch definido en el ciclo de desarrollo 3.

3.2.2.1 Pruebas unitarias

Las pruebas unitarias son aquellas que se realizan sobre cada módulo de la aplicación de manera independiente, con lo cual se busca poder determinar si dicho módulo visto como una unidad funcional independiente

que hace parte de todo el sistema se encuentra correctamente codificado y cumple funcionalmente con los requerimientos asociados.

3.2.2.1.1 Definición de la prueba

Diseño de la Prueba: Para la prueba software a realizar al módulo de Despacho de Campañas Batch se empleara la técnica de comparación causa – efecto que consiste en establecer una serie de entradas que desencadenaran una serie de salidas previamente conocidas, con lo cual se contrastan las salidas obtenidas con las esperadas detectando así los fallos existentes en el módulo. Para esta prueba se realiza el análisis de las operaciones de integración con el servidor batch verificando la comunicación de y desde el sistema hacia el administrador batch.

Generación de los casos de prueba:

Para esta parte del análisis dinámico fueron definidos 7 casos de prueba; a continuación se muestra únicamente el caso de prueba número 3

- **Caso 3:** Se desea despachar correctamente un conjunto de batches y generar una lista de despacho. Se espera que el sistema añada al servidor batch los batches elegidos y que genere una lista de despacho tomando como referencia el esquema BATCHML (*Batch Markup Language*) definido.

Procedimientos de la Prueba:

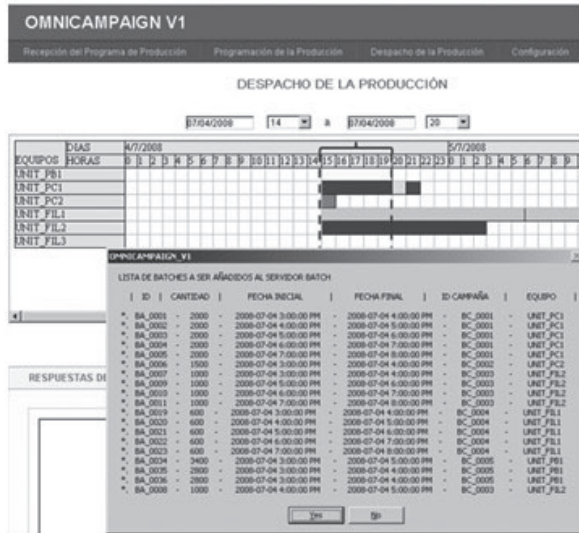
La ejecución de la prueba será realizada por el equipo de desarrollo, siguiendo en orden los casos de prueba definidos y registrando en una lista de chequeo si se cumplen los requerimientos funcionales asociados al módulo probado.

Ejecución de la Prueba:

Caso 3:

En este caso de prueba se va a escoger un rango de fechas que va desde el 04 de julio del 2008 a las 14 (24 Hrs) y termina a las 20 del mismo día, se van a despachar al servidor batch los 19 batches de las campañas en el rango verificando que todos se añadan exitosamente.

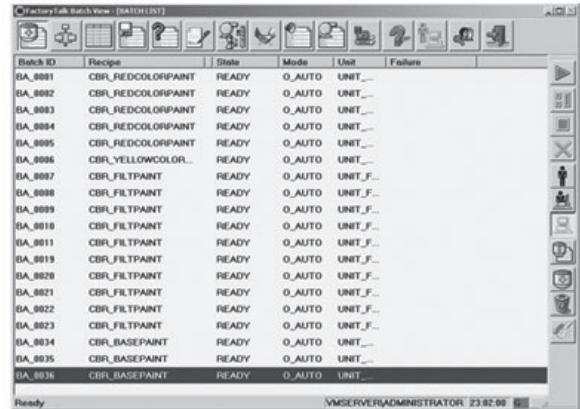
FIGURA 10. Batches a despachar entre el rango de fechas seleccionado.



Como se puede observar haciendo una comparación entre la figura mostrada anteriormente y la siguiente todos los 19 batches despachados son añadidos

satisfactoriamente al servidor batch tal cual fueron definidos en la programación detallada de la producción.

FIGURA 11. Batches despachados al administrador batch comercial.



Una vez despachados los batches es necesario generar la lista de despacho (BatchList) basada en el esquema BATCHML, en la FIGURA 12 se observa parte del código .xml generado por el sistema con la información de los batches despachados, específicamente los batches con identificadores BA_0001 y BA_0002.

FIGURA 12. Parte de la lista de despacho generada bajo el esquema BATCHML.

```
<?xml version="1.0" encoding="utf-8"?>
<b2mml:BatchList
xmlns:b2mml
="http://www.wbf.org/xml/B2MML-V05&quot;
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance&quot;
xsi:schemaLocation="http://www.wbf.org/xml/B2MML-V05
file:///C:/Documents%20and%20Settings/Administrator/Desktop
\Omnicaampaign\OMNICAMPAIGN_V1\OMNICAMPAIGN_V1\
BatchML-V05-BatchInformation.xsd">
  <b2mml:ListHeader>
    <b2mml:BatchListEntry>
      <b2mml:ID>BLE_0001</b2mml:ID>
      <b2mml:BatchListEntryType>Batch</b2mml:BatchListEntryType>
      <b2mml:Status>Ready</b2mml:Status>
      <b2mml:RecipeID>CBR_REDCCOLORPAINT.UPC</b2mml:RecipeID>
      <b2mml:BatchID>BA_0001</b2mml:BatchID>
      <b2mml:CampaignID>BC_0001</b2mml:CampaignID>
      <b2mml:ProductID>IGP-002</b2mml:ProductID>
      <b2mml:RequestedStartTime>2008-07-04T15:00:00Z</b2mml:RequestedStartTime>
      <b2mml:RequestedEndTime>2008-07-04T16:00:00Z</b2mml:RequestedEndTime>
      <b2mml:ActualBatchSize>2000</b2mml:ActualBatchSize>
      <b2mml:UnitOfMeasure>Glns.</b2mml:UnitOfMeasure>
      <b2mml:ActualEquipmentID>UNIT_PC1</b2mml:ActualEquipmentID>
    </b2mml:BatchListEntry>
    <b2mml:BatchListEntry>
    <b2mml:BatchListEntry>
    <b2mml:BatchListEntry>
    <b2mml:BatchListEntry>
  </b2mml:BatchListEntry>
```

Informe de Prueba:

Por último teniendo en cuenta los casos de prueba desarrollados se evalúa el cumplimiento de los requerimientos funcionales asociados al caso de uso despacho de campañas batch por medio de la lista de chequeo mostrada a continuación.

TABLA 3. Requerimientos funcionales cumplidos después de la verificación y validación del caso de uso.

# Ref	Requerimiento	Cumple
R1.4	El sistema deberá implementar y garantizar la integridad transaccional y de datos.	Si
R1.5	Almacenar información relevante del proceso.	Si
R1.6	Permitir configurar la información del proceso	Si
R1.7	Realizar acciones tipo Split & Merge.	Si
R1.8	La solución deberá ser desarrollada con las tecnologías de Microsoft, Visual Studio.Net, ASP.Net y SQL Server.	Si
R3.1	Emitir órdenes de trabajo de producción como fueron definidas por el programa.	Si
R3.3	Mantener el estado de las órdenes de trabajo. Ejemplo: aprobado, fijo, en proceso o cancelado.	Si
R3.4	Asegurar que las limitaciones del proceso y las órdenes bajo el nivel de detalle de la programación detallada son logradas en producción. Esto toma lugar después de que el programa es creado, pero antes de que sus elementos son ejecutados.	Si
R3.5	Informar a la programación detallada cuando un evento no anticipado resulta en la inhabilidad de lograr los requerimientos de programación.	Si
R3.6	Enviar o poner a disposición la lista de despacho de producción especificando las actividades productivas que se realizan.	Si

4. CONCLUSIONES.

El artículo presentado permite evidenciar con detalle las fases de modelado, diseño e implementación del prototipo de gestión de campañas batch basado en los estándares internacionales vigentes (ISA-88.01, ISA-95.02 e ISA-95.03), en el que se valida la integración vertical con los niveles superiores de la empresa al recibir

e interpretar el programa de producción para generar la lista de despacho de batches bajo los esquemas B2MML y BATCHML definidos por el WBF (*World Batch Forum*).

El prototipo desarrollado permite la programación de campañas batch con la aplicación de operaciones tipo Split y Merge contribuyendo al aumento de la productividad por medio de la reducción de tiempos de alistamiento y configuración, disminución de recambios, limpiezas y aprovechamiento máximo de la capacidad del equipo.

En el desarrollo del prototipo se tuvieron en cuenta buenas prácticas por medio de la aplicación de patrones de diseño brindando características como la flexibilidad al cambio, mantenibilidad y escalabilidad. La aplicación de patrones de diseño en el proceso de análisis y desarrollo de software es una ayuda para asegurar resultados acertados de manera más eficaz y fiable pues son soluciones ya probadas a problemas comunes que facilitan el análisis y codificación del proyecto.

Los resultados permiten evidenciar como al realizar una continua verificación y validación de las funcionalidades de un proyecto a lo largo de su desarrollo permite detectar fallas, omisiones o malas interpretaciones de los requerimientos a tiempo permitiendo tomar las acciones correctivas en el momento oportuno facilitando así la obtención de un producto de calidad, que satisfaga las necesidades de los clientes.

5. REFERENCIAS.

- [1] Potts, C., & Kovalyov, M. (2000). Scheduling with batching: A review. *European Journal of Operational Research*, 120, 228-249.
- [2] Méndez, C. A., Cerdá, J., Grossmann, I. E., Harjunkoski, I., & Fahl, M. (2006). State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Computers and Chemical Engineering*, 30, 913-946.
- [3] Kallrath, J. (2002). Planning and scheduling in the process industry. *OR Spectrum*, 24, 219-250.
- [4] Chapman, S. (2006). *Planificación y control de la producción*. Mexico: Prentice.
- [5] Universidad de los Andes Venezuela. (2008). *La Web del Profesor*. Venezuela. Recuperado (2013, julio 15) de http://webdelprofesor.ula.ve/economia/oliverosm/materiasdictadas/produccion2/clases/planificacion_de_la_produccion_teoría.pdf.

- [6] Domínguez, J., Álvarez, M., García, S., Domínguez, M., & Ruiz, A. (1995). *Dirección de Operaciones: Aspectos Tácticos y Operativos en la Producción y los Servicios*. Madrid: McGraw Hill.
- [7] Crawford, S., & Wiers, VCS. (2001). From anecdotes to theory: reviewing the knowledge of the human factors in planning and scheduling. En BL. MacCarthy, & JR. Wilson, (Ed.). *Human performance in planning and scheduling: fieldwork studies, methodologies and research issues*. (p.15-44). London: Taylor and Francis.
- [8] Vázquez, M. (2003). *Automatización un Dilema de Convivencia*. México. Recuperado (2013, agosto 10) de <http://mx.groups.yahoo.com/neo/groups/DesignMecanico3D/conversations/topics/2044>.
- [9] The International Society of Automation. (2005). *Enterprise Control System Integration Part 3: Activity Models of Manufacturing Operations Management*. North Carolina: American National Standard. ISBN: 1-55617-955-3.
- [10] Geambaşu, C., Jianu, I., & Gavrilă, A. (2011). Influence factors for the choice of a software development methodology. *Accounting and Management Information Systems*, 10, 479-494.
- [11] Larman, C. (2002). *UML y Patrones: Introducción al análisis y diseño orientado a objetos*. México: Pearson.
- [12] Quispe, V., Huamantuco, D., & Vargas, J. (2011). *Metodología RUP (Rational Unified Process)*. Tesis de pregrado no publicada. Universidad Nacional del Altiplano, Puno, Perú.
- [13] IBM. (2001). *Rational Unified Process: Best Practices for Software Development Teams*. California, USA. Recuperado (2013, septiembre 3) de https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf
- [14] Wells, D. (2013). *Extreme Programming: A gentle introduction*. USA. Recuperado (2013, septiembre 26) de <http://www.extremeprogramming.org>
- [15] Jeffrie, R. (2013) *XProgramming.com An Agile Software Development Resource*. USA. Recuperado (2013, septiembre 26) de <http://xprogramming.com/what-is-extreme-programming/>
- [16] Scrum Alliance. (2013). *Scrum Alliance*. USA. Recuperado (2013, septiembre 26) de <http://www.scrumalliance.org/why-scrum>
- [17] Vignaga, A., & Perovich, D. (2008). *Arquitecturas y tecnologías para el desarrollo de aplicaciones web*. Montevideo, Uruguay. Recuperado (2013, octubre 1) de <http://www.itescam.edu.mx/principal/syllabus/fpdb/recursos/r96357.PDF>
- [18] Welicki, L. (2010). *MSDN - Patrones y Antipatrones: una Introducción - Parte II*. Madrid, España. Recuperado (2013, noviembre 11) de <http://msdn.microsoft.com/es-es/library/bb972251.aspx>.
- [19] Microsoft patterns & practices. (2010). *Model-View-Controller*. USA. Recuperado (2013, noviembre 9) de <http://msdn.microsoft.com/enus/library/ff649643.aspx>
- [20] The International Society of Automation. (2001). *Enterprise-Control System Integration Part 2: Object Model Attributes*. North Carolina. American National Standard. ISBN: 1-55617-773-9.
- [21] The International Society of Automation. (1995). *Batch Control Part 1: Models and Terminology*. North Carolina. American National Standard. ISBN: 1-55617-562-0.
- [22] IEEE Computer Society. (2012). *IEEE Standard System and Software Verification and Validation*. New York. IEEE Standard Association. ISBN: 978-0-7381-7268-2.
- [23] Universidad de Sevilla. (2009). *Técnicas de Evaluación Estática*. Sevilla, España. Recuperado (2013, diciembre 10) de <http://www.lsi.us.es/docencia/get.php?id=360>
- [24] Universidad de Sevilla. (2009). *Técnicas de Evaluación Dinámica*. Sevilla, España. Recuperado (2013, diciembre 11) de <http://www.lsi.us.es/docencia/get.php?id=360>