

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/315670283>

Una Perspectiva de Ingeniería del Software para Definir Sistemas Avanzados de Planificación de Empresas

Conference Paper · September 2015

CITATIONS

0

READS

248

2 authors:



Melina Vidoni

National Scientific and Technical Research Council

21 PUBLICATIONS 12 CITATIONS

SEE PROFILE



Aldo Vecchietti

National Scientific and Technical Research Council

85 PUBLICATIONS 677 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Analysis, Characterization, Design and Development of Advanced Planning Systems to Optimize Manufacturing Processes [View project](#)



Information System to Solve Industrial Planning and Scheduling Problems [View project](#)

Una Perspectiva de Ingeniería del Software para Definir Sistemas Avanzados de Planificación de Empresas

Melina Vidoni, Aldo Vecchietti

Instituto de Desarrollo y Diseño, INGAR CONICET-UTN, Santa Fe, Argentina.
{melinavidoni, aldovec}@santafe-conicet.gov.ar

Resumen. Los sistemas de planeamiento han recibido un mayor interés desde que se ha ampliado la utilización de sistemas ERP (Enterprise Resource Planning); sin embargo, esto ha incrementado las diferencias que existen entre la teoría y la práctica del planeamiento y *scheduling*. Conocidos como APS (Advanced Planning Systems), muchos autores han concluido que estos nuevos sistemas no han sido estudiados desde un punto de vista de la Ingeniería del Software. Por lo tanto, el objetivo de este trabajo es describir los APS a través de sus requerimientos funcionales, atributos de calidad y arquitectura del software, definidos siguiendo las pautas establecidas en varios estándares internacionales de Ingeniería del Software, tales como el SEBoK (System Engineering Body of Knowledge), el modelo SQuaRE (Software Product Quality Requirements and Evaluation), entre otros.

Palabras clave: ERP, APS, advanced planning system, requerimientos funcionales, arquitectura del software.

1. Introducción

Desde su aparición, los ERP (Enterprise Resource Planning) se han convertido en un estándar *de facto* para que las organizaciones contaran con un sistema para automatizar la administración de su información [1]. Sin embargo, y debido a la intrínseca diversidad de los procesos de las organizaciones, es necesario que los ERP sean personalizados a la medida de la empresa que los implementa [2]. A pesar de esto, tienen falencias en áreas en algunos procesos de producción; lo cual es señalado por varios autores que hablan de la 'brecha' existente entre lo que un ERP puede hacer en términos de planeamiento y *scheduling* de la producción [3, 4, 5]. En 1993, Burgeois et al. [6] afirman que el incremento en la competencia y la necesidad de mejoría en la producción requieren nuevas herramientas para mejorar la optimización y el control del *scheduling* de la producción. Posteriormente, en 2003, Fleischman et al. [7] manifiestan que los sistemas ERP no proveen suficientes funciones de planeamiento, y Stadtler [8] corroboró que los ERP tienen deficiencias en el área de planeamiento. Helo et al. [9] avanzan aún más y listan explícitamente cuatro debilidades particulares de los ERP: falta de funcionalidad empresarial extendida, carencia de flexibilidad en adaptarse a los cambios en las cadenas de suministro,

insuficiencia de capacidades de decisión avanzadas y falta de una arquitectura modular abierta. En 2014, Ötzurt et al. [10] reafirman que estos problemas recaen directamente en el hecho de que la lógica de razonamiento en la que se basan los sistemas MRP-II (Manufacturing Resource Planning) es errónea.

Como consecuencia, las empresas productoras requieren un sistema más específico para administrar el planeamiento y *scheduling* de la producción, con el objetivo de optimizar los recursos materiales y humanos, fortalecer la economía de la organización y ofrecer un buen servicio a sus clientes [8]. Ese tipo de sistema posee varios nombres, de los cuales Advanced Planning Systems (APS) es el más frecuentemente utilizado [5].

Actualmente, existen algunos ERP que ofrecen módulos extra con funcionalidades de APS, como ser SAP con su módulo APO (Advanced Planning and Optimization) [11]; sin embargo, la magnitud de este sistema, y similares, junto con la complejidad de su implementación, incluyendo los cambios organizacionales que conlleva, hace que las Pequeñas y Medianas Empresas busquen desarrollar e implementar aplicaciones *ad-hoc* vinculadas a sus sistemas empresariales, que provean prestaciones para mejorar la funcionalidad de los mismos, con un presupuesto e impacto acordes a su tamaño organizacional.

Es una práctica común que la implementación de los APS, en las empresas que realizan un desarrollo propio, sea ejecutado de forma improvisada, involucrando varios *stakeholders* (o interesados), consultores y personal interno, sin una metodología establecida para guiar el proceso. Como consecuencia, hay un alto interés y necesidad en comprender de mejor forma los factores de éxito y fracaso en la implementación de este tipo de sistemas [12], lo que se ve agravado por la falta de una definición precisa y aceptada sobre qué es un APS, cuáles son sus prestaciones, qué módulos y soluciones incluyen [7], etc. Muchas implementaciones disponibles en el mercado tampoco se ajustan a una definición común [5]. A su vez, una gran parte de los trabajos académicos existentes son principalmente teóricos sin implementaciones reales, enfocadas en modelos matemáticos para resolver un *scheduling*, más que enfocarse en un sistema como un todo [13, 14]. Stadlter [8] menciona que hay varias áreas que necesitan mejoría y difusión: definir los módulos que incluye un sistema de planeamiento, los problemas que desafían las necesidades actuales y la integración entre las diversas funciones de estos sistemas. Henning [4] también indica que existe una falencia en la investigación desde el área de Sistemas de Información respecto a los Advanced Planning Systems. En 2012, Aslan et al. [15] insistieron que los problemas continuaban: la literatura es escasa, no hay detalles en el concepto de sistemas de planeamiento, y hay una alta necesidad de información del funcionamiento interno y las aplicaciones de tales sistemas. Framinan y Ruiz [3] también mencionan este tema, al decir que, muy a menudo, la investigación en el área no aborda la arquitectura de los sistemas, y no provee a los desarrolladores con una visión comprensiva de los APS. Más aún, Kallestrup et al. [16] reafirman la necesidad de investigación en el desarrollo de los sistemas de *scheduling*. De manera similar, Zoryk-Schalla et al. [12] aclaran que implementar un APS puede demandar cambios organizacionales, y que son escasos los trabajos que hacen referencia a ello.

El objetivo del presente trabajo es, desde un enfoque de la Ingeniería en Sistemas,

caracterizar los *Advanced Planning Systems* definiendo sus requerimientos funcionales y atributos de calidad, los cuales derivan en una propuesta de arquitectura del software. Para esto, se analizaron las definiciones propuestas en la literatura actual sobre APS, estableciendo los vínculos comunes entre varios conceptos, y definiendo las características principales destacadas en esos trabajos. A su vez, la propuesta se basa en las especificaciones de varios estándares internacionales como ser: SEBoK (System Engineering Body of Knowledge) [17], ISO/IEC 2501:2011 [18] (modelo SQuaRE: Software Product Quality Requirements and Evaluation) y el ISO/IEC/IEEE 42010:2011 [19], para arquitectura del software.

2. Definiciones Iniciales

2.1 Planeamiento de Fábricas (Factory Planning)

Es frecuente encontrar en la bibliografía que los sistemas de planeamiento tienen distintos grupos de funcionalidades, a menudo llamados ‘módulos’ o ‘niveles de planeamiento’ [5, 7, 8, 10, 12, 20, 21, 22, 23]. Estos niveles se clasifican de acuerdo con el segmento de la empresa o cadena de suministro que optimizan (por ejemplo: producción, distribución, logística, ventas, etc.) y al horizonte de tiempo al que se aplican (corto, medio o largo). Dichos módulos son: Planeamiento Estratégico de la Red, Planeamiento de Demanda (DP), Planeamiento Maestro (MP), Planeamiento y Scheduling de la Producción, Planeamiento de Distribución, Planeamiento del Transporte, y Cumplimiento de Demanda.

No obstante, la mayoría de estos niveles poseen características comunes, lo que permite agruparlos una categoría denominada *factory planner*, concepto propuesto por Zoryk-Schalla et al. [12]. En este trabajo, se considera al *factory planning* como un concepto amplio, que incluye varios tipos de planeamiento a diferentes horizontes de tiempo, sin restringirse a ninguno de los subtipos mencionados previamente.

2.2 Advanced Planning Systems

Como se indicó previamente, no existe una definición aceptada sobre qué es un APS y los módulos y soluciones que ofrecen [3, 5, 7]. Para resolver esta situación, y elaborar una definición que sirva como base para este trabajo, se analizaron las definiciones y acrónimos existentes, junto con su significado. Los dos conceptos más utilizados son abreviados como APS, pero poseen significados distintos: Advanced Planning and Scheduling [5, 10, 14, 15, 20, 21, 24, 25, 26, 27] y Advanced Planning System [4, 7, 8, 12, 16, 22, 28].

El primero, Advanced Planning and Scheduling, es comúnmente definido como la automatización e informatización del proceso de planeamiento de un problema, a través de la utilización de simulación u optimización [5]; sin embargo, muchos trabajos que definen este concepto, lo ligan a un tipo específico de solución, en particular, modelos matemáticos [14, 26]. En las definiciones encontradas, siempre se menciona algún *método de resolución* (MR) que es utilizado para resolver el problema de planeamiento; los métodos más comunes de la literatura son los modelos

matemáticos [5, 6, 10, 12] y los algoritmos genéticos [21, 25, 29, 30], aunque también pueden utilizarse otros métodos, como simulación o teoría de juegos.

El segundo concepto es Advanced Planning Systems, y es usualmente definido como una herramienta de software que ayuda a planificar y tomar decisiones sobre los *schedulings* operacionales detallados [7]. Las definiciones comunes distinguen a este concepto como a un sistema de información y, aún más, como un caso particular de un DSS (Decision Support Systems) [10]. Muchos trabajos [3, 7, 8, 20, 24, 28] también mencionan la necesidad de enlazar los Advanced Planning System con el ERP (o sistema transaccional) actual de la empresa, con el objetivo de extraer la información necesaria para realizar la optimización, y posteriormente guardar dichos resultados en el ERP. Dado que una organización puede tener un ERP, un Sistema de Información (SI) de desarrollo propio, un software empaquetado, u otros, se utilizará el término *sistemas empresariales* (SE) para englobar estos sistemas y evitar posibles confusiones. Finalmente, es importante destacar el énfasis que existe en considerar al *planeador humano* como una parte importante del mismo: no sólo que debe estar siempre en control de lo que se realiza, sino que el APS debe ayudarlo en la toma de decisiones y no reemplazarlo [3, 4, 7].

La relación más importante que se puede deducir, y que no fue tratado como tal en la literatura, es que un Advanced Planning System ayuda a resolver problemas de *factory planning* (habitualmente agrupados bajo el concepto de Advanced Planning and Scheduling). Como consecuencia, se propone la siguiente definición:

Los Advanced Planning Systems (APS) son sistemas de información que utilizan métodos de resolución avanzados (tales como investigación operativa, algoritmos genéticos, simulación, entre otros) para resolver uno o más problemas de factory planning. Un APS debe ser capaz de establecer una conexión al Sistema Empresarial de la organización para extraer los datos necesarios y almacenar los resultados aprobados. Un APS no reemplaza al planeador humano sino que lo complementa, permitiéndole estar siempre en control para aceptar, modificar o rechazar los resultados ofrecidos por el sistema.

Hay que destacar que la definición propuesta apunta a un tipo de sistema que está intrínsecamente relacionado con el problema de *factory planning* que busca resolver, sin ligarse a un método de resolución en particular.

3. Enfoque de Ingeniería del Software

La Ingeniería del Software (IS) es una disciplina ingenieril enfocada con todos los aspectos de producción del software, desde la temprana etapa de especificación, hasta el mantenimiento durante el uso [31]. La sub-disciplina encargada con generar dicha especificación, es la Ingeniería de Requerimientos (IR), que se centra en elicitar, modelar y analizar los requerimientos y el ambiente del sistema en construcción [32]. Elicitar requerimientos es un proceso relativamente complicado, no trivial, que extrae conceptos desde los stakeholders y los transforma en requerimientos.

Un defecto de los trabajos de la literatura relacionados con los APS como sistemas [7], es que no abordan la temática desde una perspectiva de la Ingeniería del Software

[4, 14, 15]. Si se analiza la bibliografía se puede ver que existen contradicciones y desarreglos en cuanto a los Requerimientos Funcionales y Atributos de Calidad que los sistemas deben tener. Sin embargo, es importante precisar estas características para realizar un modelado e implementación consistente de los APS [12]. Para ello, existen tres pilares fundamentales sobre los que se basa un sistema: Requerimientos Funcionales (RF), Atributos de Calidad (AC), y Arquitectura del Software (AS).

La meta principal es proponer una caracterización genérica del concepto de APS, la cual pueda extenderse como una herencia y especializarse para abarcar características particulares de implementaciones concretas. Como consecuencia de este enfoque, no existen stakeholders individuales, sino ‘grupos objetivo’ a los cuales se intenta llegar; por lo tanto, se emplea la literatura actual como medio para elicitar los requerimientos en los que se basa la propuesta.

3.1 Requerimientos Funcionales

El SEBoK [17] define a los *requerimientos del sistema* como descripciones de las funcionalidades que debe alcanzar como un todo para poder satisfacer las necesidades de los stakeholders, y que deben ser expresadas de forma correcta. Siendo más específicos, el mismo estándar define *requerimientos funcionales* (RF) como una: “[...] *declaración que describe cuantitativamente una funcionalidad o tarea que el sistema debe realizar mientras esté operando; RF describen lo que el sistema debe ser capaz de hacer o llevar a cabo*”.

Al utilizar la literatura existente para elicitar requerimientos básicos, se encuentra que pocos trabajos proveen listas de *features* de APS como sistema, y aquellas que lo hacen, suelen tener uno o más de los siguientes inconvenientes:

- Presentan características del sistema, pero en realidad listan requerimientos para un método de resolución en particular.
- Presentan atributos de calidad como requerimientos funcionales, y los definen incorrectamente.
- No se incluyen algunos requerimientos exclusivos del sistema, tales como login o ingreso, tipo de datos a ser almacenados, etc.

Siguiendo la definición propuesta de APS, puede comprobarse que éstos están intrínsecamente relacionados con el problema de *factory planning* que buscan automatizar. Como consecuencia, los requerimientos funcionales se ven afectados por esta relación. De este modo, un RF puede estar relacionado tanto con el sistema como con el método de resolución, o también puede afectar a sólo uno de estos.

Por lo tanto, la Tabla 1 presenta los requerimientos que afectan al sistema, mientras que la Tabla 2 lista los que afectan al enfoque de resolución. En ambos casos, la primera y segunda columna detallan un código y nombre para cada requerimiento con el objetivo de demostrar que es un mismo RF puede estar en ambas tablas, afectando tanto al sistema como a su MR.

A continuación se presenta una descripción y aclaración más profunda de cada requerimiento y, si corresponde, se citan los trabajos de los que fueron elicitados:

Tabla 1. Requerimientos funcionales relacionados con el sistema.

Sistema		
#	Nombre	Descripción
(a)	<i>Puntos de Optimización</i>	El sistema debe tener un menú u ofrecer una forma de seleccionar qué punto será optimizado. Un punto puede ser: diferentes ubicaciones en la fábrica, logística, transporte y demás.
(b)	<i>Selección del Modelo</i>	El sistema debe sugerir un modelo por defecto (o el más apropiado) para cada punto, pero el planeador humano debe poder seleccionar un modelo distinto.
(c)	<i>Selección de Objetivos</i>	El sistema debe sugerir un objetivo por defecto (o el más apropiado) para usar con cada modelo, pero también debe permitirle al planeador humano seleccionar un objetivo diferente para usar.
(d)	<i>Configuración de Parámetros</i>	El sistema debe proveer una interfaz gráfica de usuario (GUI) para que el planeador pueda colocar valores personalizados en los parámetros y evaluar diferentes posibilidades.
(e)	<i>Generación de Escenarios</i>	El sistema debe permitirle al planificador colocar un rango de variación para cada/algunos parámetros. El sistema luego debe correr cada escenario automáticamente luego de que los parámetros han sido colocados.
(f)	<i>Almacenamiento de Escenarios</i>	El sistema debe automáticamente almacenar los resultados para cada escenario, para que el planificador sea capaz de revisarlos luego. Esto no implica impactar la base de datos del ES sin el consentimiento del planeador humano.
(g)	<i>Comparación de Escenarios</i>	Luego de que todos los escenarios han sido completados, el sistema debe permitir que el usuario los compare. Tal comparación debe mostrar gráficos, estadísticas del tiempo de resolución y más. En caso de infactibilidad, debe mostrar información que permita estudiar las causas de la infactibilidad.
(h)	<i>Entrada de Datos</i>	El sistema debe extraer los datos de entrada para correr cada modelo desde el ES utilizado por la empresa.
(i)	<i>Comprobación de Consistencia</i>	El sistema debe proveer funcionalidades para revisar la consistencia de los datos automáticamente ingresados en el sistema. Debe comprobar la existencia de todos los recursos necesarios para el scheduling de los trabajos, incluyendo la disponibilidad de la materia prima (debe comprar la BoM contra la cantidad existente actual), el estado de las máquinas, etc. Entonces, debe informar claramente al usuario si hay, o no, algún problema con los recursos existentes.
(j)	<i>Salida de Datos</i>	El sistema necesita traducir los resultados del modelo en un formato que sea comprendido por el sistema transaccional. El APS va a impactar en la base de datos del ES sólo el escenario seleccionado por el usuario, ya sea exactamente como el APS lo produjo, o con modificaciones realizadas manualmente.
(k)	<i>Función de Login</i>	El APS debe ofrecer una función de ingreso para restringir el uso. Tales credenciales pueden ser las mismas del ES, o nuevas.
(l)	<i>Intercambio de Información</i>	El intercambio de información entre el APS y otros sistemas debe estar basado en un estándar (tal como S95 o S88).
(m)	<i>Abrir/Guardar Resultados</i>	El sistema debe ser capaz de abrir y mostrar resultados previos con los mismos gráficos usados anteriormente, durante la Comparación de Escenarios.
(n)	<i>Integración de Algoritmos</i>	El sistema debe permitirle al usuario agregar, modificar o editar contenido, en una forma sencilla. Esto puede incluir, no estrictamente: nuevos puntos, nuevos modelos por cada punto, nuevos objetivos por modelos y nuevos rangos/cambios en los parámetros. Esta característica debe estar restringida a ciertos usuarios.
(o)	<i>Detección de Cuellos de Botella</i>	El sistema debe hacerle saber al usuario de cualquier resultado encontrado en la búsqueda por posibles cuellos de botella y recursos deficientes.
(p)	<i>Evaluación Pos-Ejecución</i>	El sistema debe poder monitorear la ejecución de los <i>planeamientos</i> proyectados para poder medir la desviación entre lo teórico y lo realmente ejecutado (esto significa leer los datos del 'tiempo real de producción' del ES, y compararlo contra lo que fue programado). El APS debe permitir colocar esto en modo automático (se lanza solo, guardando un reporte) o manual (disparado por el usuario, mostrando resultados en tiempo real).
(q)	<i>Comprobación de Rescheduling</i>	Si se producen desviaciones de los planes originales, el sistema debe mostrar si los trabajos actuales deben ser vueltos a planificar, o si se debe ejecutar un <i>scheduling</i> completo. La decisión final debe ser tomada por el usuario, o permitir una opción automatizada.
(s)	<i>Base de Datos</i>	El APS debe tener su propia base datos, con el objetivo de almacenar modelos, objetivos, puntos, configuraciones, y demás.

Tabla 2. Requerimientos funcionales relacionados con el enfoque de resolución.

Enfoque de Resolución		
#	Nombre	Descripción
(a)	<i>Puntos de Optimización</i>	Cada punto debe tener al menos un modelo o medio para optimizar.
(b)	<i>Selección del Modelo</i>	El sistema debe sugerir un modelo por defecto (o el más apropiado) para cada punto, pero el planeador humano debe poder seleccionar un modelo distinto.
(c)	<i>Selección de Objetivos</i>	Cada modelo en cada punto puede tener varios objetivos disponibles para optimizar, sin la necesidad de ser multi-objetivo. Los objetivos disponibles deben cambiar de modelo a modelo, y de punto en punto. Un modelo también puede ser multi-objetivo.
(d)	<i>Configuración de Parámetros</i>	El modelo debe tener parámetros modificables que permitan la generación de escenario. Cada parámetro debe tener un valor por defecto.
(h)	<i>Entrada de Datos</i>	Cada modelo debe correr con datos reales de la organización.
(j)	<i>Salida de Datos</i>	Los modelos generan como resultados órdenes de producción y planillas de <i>scheduling</i> .
(o)	<i>Detección de Cuellos de Botella</i>	El modelo debe detectar potenciales cuellos de botella y recursos deficientes antes de que una solución para el problema sea proveída.
(q)	<i>Comprobación de Rescheduling</i>	El modelo debe permitir re-scheduling, manteniendo algunos de los trabajos como ya habían sido previamente programados.
(r)	<i>Modelos y Procedimientos</i>	Debe haber separación del modelo y de los procedimientos de solución, para que diferentes procedimientos puedan ser aplicados a la instancia de un problema.

- **Punto de Optimización** Es el requerimiento central del sistema, ya que un *punto de optimización* es cada problema de *factory planning* que necesita ser resuelto mediante un APS, y debe haber al menos un punto a optimizar. De todos modos, un APS puede optimizar más de un punto, sin existir un límite máximo para cuántos puede tener el sistema. Se elicitó de ideas presentadas en trabajos previos [3, 12, 21], donde proponían que el objetivo de un APS es automatizar la solución del (o los) problemas de planeamiento. A su vez, es importante mencionar que cada punto de optimización puede ser MTS (Make-To-Stock), MTO (Make-To-Order), ETO (Engineer-to-order), o tener una estructura mixta, independientemente de cómo trabajen los otros puntos de optimización. Esto impacta directamente en la arquitectura propuesta.
- **Selección de Modelo y Objetivo** Cada punto disponible para optimización debe tener al menos un *modelo* (o medio de resolución, entendido como un concepto no exclusivo de la investigación operativa) y no tiene restricciones de máxima disponibilidad. A su vez, cada modelo debe tener al menos un *objetivo*.
- **Integración Estándar con ES:** De la literatura se pueden extraer los requerimientos de integrar un APS con el ES que emplee la empresa [3, 7, 8, 28, 24, 20]. Sin embargo, es necesario una comunicación estandarizada para facilitar el traspaso de información, obtener una integración transparente. El tipo de estándar recomendado también ha sido elicitado de varias fuentes [13, 20, 28], las cuales mencionan al ANSI/ISA-95 (conocido como S95) [33] y ANSI/ISA-88.
- **Información y Consistencia** Incluye los requerimientos de ‘Comprobación de consistencia’ y ‘Detección de Cuellos de Botella’, dado que una fuerte interrelación entre estos componentes mejora la generación de planeamientos

realistas; un APS debe ser capaz de detectar restricciones, y proveer al usuario medios para generar/modificar la información relacionada con el planeamiento. Fue elicitado de varios autores [6, 13, 34, 35]

- **Administración de Escenarios** Incluye requerimientos como ‘Configuración de Escenarios’, ‘Generación de Escenarios’ y ‘Comparación de Escenarios’. Es un conjunto de requerimientos clave, ya que ayuda al planeador humano a tomar decisiones estratégicas y mantener consistencia entre ellas; son funcionalidades vitales para permitir comparaciones y generar evaluaciones complejas. Fue elicitado de la literatura en el área [3, 10, 28, 36].
- **Pericia Humana** Aunque no es un RF *per se*, tiene un amplio impacto en la construcción del sistema, y debe considerarse en cada etapa. Los planeadores humanos tienen un rol importante, y deben estar siempre bajo control de los resultados del sistema, pudiendo modelar, configurar reglas de decisión y realizar ajustes manuales. No sólo es parte de la definición de APS, sino que ha sido elicitado de la literatura [3, 4, 5, 12].
- **Base de Datos del APS** Referido al requerimiento ‘Base de Datos’, es una consecuencia de los otros RF presentados, ya que debido a la cantidad de datos generado por un APS, y que no necesariamente son almacenados en el ES, se necesita una base de datos propia; los datos que se guardan incluyen los generados por ‘Selección de Modelos’, ‘Selección de Objetivos’, escenarios aun no impactados en el ES, parámetros, etc. En la literatura se han encontrado fuentes que avalan la necesidad del APS de tener una base de datos propia [3, 7].
- **Actualización e Integración de Algoritmos** Abarca los RF ‘Integración de Algoritmos’ y ‘Función de Log-In’ entre otros. Son consecuencia de la necesidad de administrar los puntos de optimización desde la interface gráfica, a su vez, restringiendo el acceso a tales funcionalidades, para prevenir el uso malicioso del APS. Es un conjunto de requerimientos muy importante que impacta profundamente en los atributos de calidad.

3.2 Atributos de Calidad

En la IS, los Atributos de Calidad (AC) son también conocidos como requerimientos no funcionales. El glosario del SEBoK [17] los define como propiedades o características inherentes de un sistema, que pueden ser distinguidos cuantitativa o cualitativamente a través de medios humanos o automatizados. Un estándar muy empleado era el ISO/IEC 9126-1, pero ha sido reemplazado por el ISO/IEC 25010:2011 [18], conocido como modelo SQuaRE.

Para que los AC puedan ser evaluados y estudiados en un proyecto de software, deben ser medidos a través de *métricas* específicas (definidas como “una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado”) empleadas en *indicadores* (es una métrica o combinación de métricas que proveen conocimiento sobre un proceso o proyecto de software, o sobre el producto en sí mismo). La división ISO/IEC 2502n “Quality Measurement”, es parte del modelo SQuaRE, y será empleada para elaborar las métricas e indicadores pertinentes y recomendadas para un APS. Sin embargo, por cuestiones de espacio, este tema no

será abordado en el presente trabajo, sino que el enfoque estará en presentar una lista inicial de AC sobre la cual basar el estudio.

Los AC presentados en esta lista fueron generados como consecuencia de los RF, buscando complementar su funcionamiento, y siguiendo el modelo SQuaRE [18], mediante las descripciones propuestas por el SEBoK [17]. Es importante mencionar que esta lista se adecúa al propósito de elaborar una especificación genérica, y cada atributo puede ser especificado para casos particulares. Cabe mencionar que también se utilizó la literatura actual para la elicitación, aunque los trabajos sobre APS que mencionan AC son escasos, y generalmente no emplean éste término, usando definiciones de AC tipo *ad-hoc*, no basadas en estándares.

- **Compatibilidad** Es el grado en que dos o más sistemas o componentes pueden intercambiar información y/o llevar a cabo funciones requeridas mientras comparten el mismo hardware o ambiente de software [18]. Deriva de varios RF propuestos en la Tabla 1, tales como ‘Entrada/Salida de Datos’ e ‘Intercambio de Información’, y por ende, fue elicitado de la literatura [8, 28, 24, 20]. Hay dos sub-características seleccionadas: *interoperabilidad* (grado en que dos o más sistemas pueden intercambiar información y usarla adecuadamente) y *coexistencia* (grado de eficiencia al realizar las funciones requeridas, mientras se comparte un ambiente y recursos con otros sistemas) [37].
- **Portabilidad** Es el grado en que un sistema puede transferirse, de forma eficiente y efectiva, de un ambiente operacional (hardware, software, etc.) a otro [18]. Elicitado de la literatura [6, 38], la adaptabilidad de un sistema es una cualidad crucial para planificar su desarrollo e implementarlo correctamente. La única sub-característica seleccionada es *adaptabilidad* (a distintos ambientes o usos [37]).
- **Confiabilidad** Es el grado en que un sistema lleva a cabo funciones definidas durante condiciones específicas, en un período de tiempo [18]. La sub-característica seleccionada es *tolerancia a los fallos*: grado en que puede operar como se planificó durante fallas de hardware o software [37], fue elicitado de dos trabajos [4, 9], donde mencionan robustez y “aptitud para el cambio” como cualidades de los APS.
- **Mantenibilidad** Es el grado de efectividad y eficiencia con que el sistema puede ser modificado [18]. Incluye varias sub-características: *modularidad* (grado en que un sistema está formado por componentes discretos [37], fue elicitado de referencias a la capacidad de modificación y expansión que un APS debe tener [6, 7]), *modificabilidad* (compuesto por Cambio y Estabilidad, es el grado en que el sistema puede ser modificado sin agregarse defectos [37]) y *test-habilidad* (grado de eficiencia y eficacia con que se pueden establecer test para el sistema [37]; es importante debido a la tendencia actual a automatizar el uso de test-unitarios y verificar y validar los sistemas).
- **Usabilidad** Es un atributo importante que indica el grado en que un sistema puede ser comprendido, aprendido, utilizado y atractivo al usuario [18]. Debido a la alta influencia que el planeador humano tiene en un APS [4], este atributo afecta varios RF y tiene las siguientes sub-características: facilidad de aprendizaje, facilidad de operación y control, accesibilidad al usuario y estética satisfactoria [37].

- **Adecuación Funcional** Es el grado en que el producto satisface las necesidades declaradas en los RF [18]. Es un atributo enfocado al desarrollo de un sistema, y las dos sub-características principales son: *corrección* y *adecuación*, ambos implicados en proveer resultados correctos y facilitar el cumplimiento de objetivos, respectivamente [37].
- **Seguridad** Es importante debido a la necesidad de protección de la información y datos para que personas no autorizadas no puedan alcanzar zonas restringidas [18], ni generar un uso malicioso del sistema. Las tres características principales para un APS, son: confidencialidad, integridad y autenticidad [37]. Varios requerimientos de la Tabla 1, tales como ‘Función de Log-in’ e ‘Integración de Algoritmos’ requieren de este atributo de calidad.
- **Performance** Relacionado con la cantidad de recursos que el sistema emplea bajo condiciones específicas [18], requiere evaluar el comportamiento en el tiempo y la utilización de recursos del sistema [37]. Está relacionado con RF de administración de escenarios, ‘Comprobación de Consistencia’ y ‘Detección de Cuellos de Botella’, entre otros.

3.3 Arquitectura del Software

La arquitectura del software (AS) es una parte intrínseca del diseño del software, y debe ser apropiadamente detallada. Abarca decisiones significativas sobre la organización del software, componentes estructurales, interfaces, comportamientos, composición de elementos, entre otros aspectos [39]. El estándar actual para la AS es el ISO/IEC/IEEE 42010:2011 [19], referido a su creación, análisis y mantenimiento. Sin embargo, hay una falta de información sobre este tema en los trabajos de investigación sobre APS, lo que ha sido destacado por varios autores [3, 4, 8].

La arquitectura es un aspecto que ampliamente influencia el diseño e implementación de los sistemas; esto se debe a que diseñar una AS no es una tarea sencilla, y está relacionada con los requerimientos funcionales y atributos de calidad que el sistema quiere alcanzar [39]. La Figura 1 propone, entonces, una arquitectura de capas lógicas (o *layers*) [40], que sólo proveen una forma de organizar el código.

Más allá de la posición de los bloques, la propuesta sólo trabaja con capas, las cuales pueden –posteriormente y al momento de un desarrollo específico- ser implementadas en niveles físicos (o *tiers*), para adecuarse a cada caso particular [39]. Es preciso aclarar que esta propuesta es sólo una etapa inicial, que sólo incluye componentes básicos y que busca ser una línea base sobre la cual proponer un modelo de vistas 4+1 [41], incluyendo escenarios y representando detalladamente cómo los AC y los RF afectan la arquitectura propuesta.

La AS propuesta contiene tres grandes capas:

- **Capa de Datos** Consiste en la base de datos del APS (abreviado como BD-APS) y en los bloques de control que administran y traducen los datos del formato almacenado en la BJD, al que necesita el sistema. Es importante mencionar que la base de datos del Sistema Empresarial (abreviado BD-SE) está fuera de la arquitectura del APS, aunque este tiene un bloque de control (‘Control BD-SE’) para adminis-

trar los datos que persiste en el ES; a su vez la forma de comunicación con ésta (ya sea directa o por medio de una Interfaz de Acceso -la cual puede o no existir-) depende del SE y de decisiones de implementación que se tomen en la organización. De forma contraria, la administración de la BD-APS recae exclusivamente en el APS, a través del bloque ‘Control BD-APS’. El bloque ‘Traductor’ actúa como un *hub* entre los otros bloques, asegurando varias características y atributos de calidad referidos a la abstracción; es también el encargado de actuar como interface entre la Capa de Datos, y las otras capas de la arquitectura.

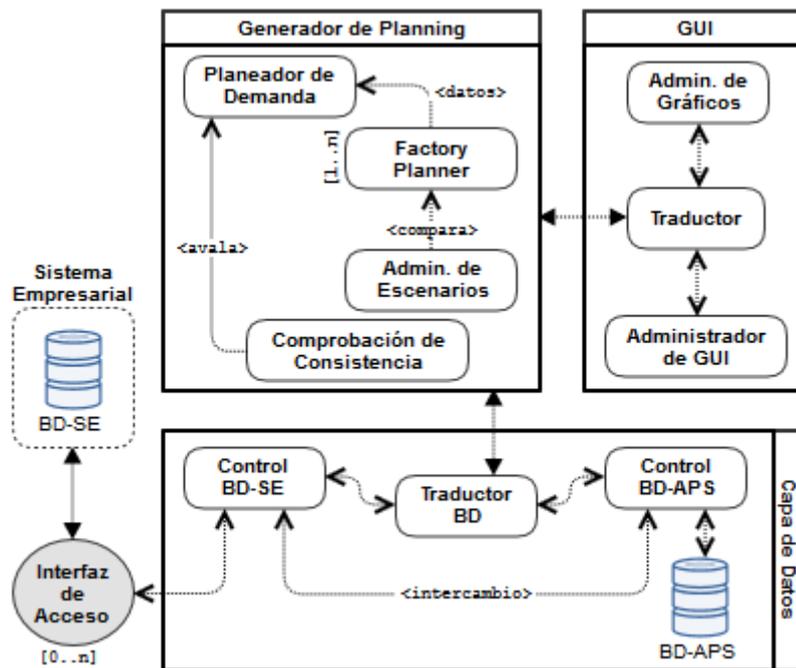


Fig. 1. Propuesta inicial de arquitectura en capas lógicas para un APS.

- **Generador de Planning** Contiene la lógica principal del APS, y su bloque central es 'Factory Planner' (FP), que representa el concepto de *factory planning* previamente mencionado; debe haber al menos un FP por sistema, pero sin límite superior, ya que cada instancia de este bloque representa un Punto de Optimización. Como consecuencia, debe existir un bloque 'Planeador de Demanda' por cada 'Factory Planner' que trabaje con pronósticos (modelo MTS), ya que si un FP es de tipo MTO o ETO, va a requerir datos de las órdenes de compra, y no pronósticos de ventas. El bloque 'Comprobación de Consistencia' está relacionado con el requerimiento de nombre similar, mientras que el bloque 'Admin. de Escenarios' abarca la lógica de los requerimientos que administran escenarios.
- **GUI** Identificada por las siglas de Graphical User Interface, contiene la lógica para la interfaz gráfica, y debe adecuarse a los AC de Usabilidad y Performance, entre otros. El bloque principal es el 'Traductor', de funcionamiento similar a un *hub*, que se comunica con las otras capas, traduciendo los datos al formato neces-

rio para mostrarlos por pantalla; la relación de este bloque con la segunda capa está altamente influenciada por el lenguaje de programación que se utilice. Características más específicas de la GUI deben ser estudiadas luego de que se haya tomado alguna decisión específica de implementación.

4. Conclusiones

El presente trabajo investiga, desde una perspectiva de Ingeniería del Software, las características principales de un Advanced Planning System, desde su definición hasta la generación de Requerimientos Funcionales (RF), Atributos de Calidad (AC) y Arquitectura del Software (AS). La propuesta se diferencia de otros artículos que sólo se enfocan en los modelos y algoritmos para resolver problemas de planeamiento muy específicos, para concentrarse en darle un marco conceptual al desarrollo de sistemas con las características de un APS.

El artículo propone una definición de APS como un sistema de información, basado en el concepto de *planeamiento de fábricas*. A través de un análisis de los trabajos en el área, se propone una caracterización de APS basada en estándares internacionales de amplia aceptación en el área de Ingeniería del Software, tales como el SEBoK (System Engineering Body of Knowledge, mantenido y administrado por la IEEE, entre otros), el modelo SQuaRE (Systems and software Quality Requirements and Evaluation, ISO/IEC 25010:2011) y el ISO/IEC/IEEE 42010:2011 (Systems and Software Engineering, Architecture description). Tal caracterización es genérica y extensible, no restringida a casos particulares, con el objetivo de proveer una definición base que pueda ser usada como punto de partida de un conjunto de definiciones que le agregan especificidad. Una característica principal de esta propuesta es que la caracterización refuerza la relación existente entre los RF, los AC y la AS: los requerimientos funcionales son fortalecidos por los atributos de calidad, y ambos tienen consecuencias en la arquitectura propuesta. Esta interrelación de conceptos agrega robustez al trabajo, y mejora la capacidad del mismo de adaptarse a casos de estudio específicos. Otra característica novedosa es destacar el impacto de los RF tanto en el sistema, como en el método de resolución.

Varios trabajos futuros derivan de esta propuesta. En primer lugar, completar las definiciones de los AC, ofreciendo métricas e indicadores –basados en los estándares de la división ISO/IEC 2502n del modelo SQuaRE- para poder evaluarlos en los sistemas reales; además, se puede analizar el Modelo de Calidad de Datos, generado a partir del estándar internacional ISO/IEC 25012:2008 que provee una base para la calidad de la información. En segundo lugar, la descripción detallada de la AC, a través de un método de documentación tal como el Modelo de Vistas 4+1, es un tópico importante y necesario para proveer una arquitectura que pueda posteriormente ser especificada y detallada para la implementación de casos particulares. Finalmente, la caracterización debería ser evaluada con mayor detalle a través de la elicitación de requerimientos de stakeholders particulares de varios casos de estudio reales.

5. Referencias

1. Wang Shaojun, Wang Gang, Lü Min y Gao Guoan, Enterprise resource planning implementation decision & optimization models, *Journal of Systems Engineering and Electronics*, vol. 19, nº 3, pp. 513-521, 2008.
2. M. A. Rothenberger y M. Srite, An Investigation of Customization in ERP System Implementations, *Transactions On Engineering Management*, vol. 56, nº 4, pp. 663-676, 2009.
3. J. Framinan and R. Ruiz, Architecture of manufacturing scheduling systems: Literature review and an integrated proposal, *European Journal of Operational Research*, vol. 205, no. 2, pp. 237-246, 2010.
4. G. Henning, Production Scheduling in the Process Industries: Current Trends, Emerging Challenges and Opportunities, *Computer Aided Chemical Engineering*, vol. 27, pp. 23-28, 2009.
5. H.-H. Hvolby and K. Steger-Jensen, Technical and industrial issues of Advanced Planning and Scheduling (APS) systems, *Computers in Industry*, vol. 61, no. 9, pp. 845-851, 2010.
6. S. Burgeois, A. Artiba y C. Tahon, Integration of Short Term Scheduling with an MRP-II System: Industrial Implementation, de *International Conference on Industrial Electronics, Control, and Instrumentation*, Maui, HI, 1993.
7. B. Fleischmann y H. Meyr, Planning Hierarchy, Modeling and Advanced Planning Systems, *Handbooks in Operations Research and Management Science*, vol. 11, pp. 455-523, 2003.
8. H. Stadler, Supply chain management and advanced planning—basics, overview and challenges, *European Journal of Operational Research*, vol. 163, no. 3, pp. 575-588, 2005.
9. P. Helo, M. Suorsa, Y. Hao and P. Anussornitisarn, Toward a cloud-based manufacturing execution system for distributed manufacturing, *Computers in Industry*, vol. 65, no. 4, pp. 646-656, 2014.
10. C. Öztürk y A. M. Ornek, Operational extended model formulations for Advanced Planning and Scheduling systems, *Applied Mathematical Modelling*, vol. 38, nº 1, pp. 181-195, 2014.
11. S. Mannah y W. Segatto, SCM Advanced Planning and Optimization (APO), 07 2013. [En línea]. Available: <http://wiki.scn.sap.com/wiki/pages/viewpage.action?pageId=11868>. [Último acceso: 06 2015].
12. A. J. Zoryk-Schalla, J. C. Fransoo and T. G. de Kok, Modeling the planning process in advanced planning systems, *Information & Management*, vol. 42, no. 1, pp. 75-87, 2004.
13. I. Harjunkoski, R. Nyström and A. Horch, Integration of scheduling and control—Theory or practice?, *Computers and Chemical Engineering*, vol. 33, no. 12, p. 1909–1918, 2009.
14. K. Chen y P. Ji, A mixed integer programming model for advanced planning and scheduling (APS), *European Journal of Operational Research*, vol. 181, nº 1, pp. 515-522, 2007.
15. B. Aslan, M. Stevenson and L. Hendry, Enterprise Resource Planning systems: An assessment of applicability to Make-To-Order companies, *Computers in Industry*, vol. 63, no. 7, pp. 692-705, 2012.
16. K. B. Kallestrup, L. H. Lyng, R. Akkerman y T. A. Oddsdottir, Decision support in hierarchical planning systems: The case of procurement planning in oil refining industries, *Decision Support Systems*, 2014.
17. BKCASE Editorial Board, The Guide to the Systems Engineering Body of Knowledge (SEBoK), 1.3 ed., Hoboken, NJ: The Trustees of the Stevens Institute of Technology, 2014.
18. ISO/IEC 25010:2011, Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models, ISO, 2011.
19. ISO/IEC/IEEE 42010:2011, Systems and software engineering -- Architecture description, C. -. S. & S. E. S. Committee, Ed., IEEE Computer Society, 2011.
20. P. Hadaya y R. Pellerin, Determinants of Advance Planning and Scheduling Systems Adoption, de *The Third International Conference on Software Engineering Advances*, Sliema , 2008.
21. L.-C. Kung y C.-C. Chern, Heuristic factory planning algorithm for advanced planning and scheduling, *Computers & Operations Research*, vol. 36, nº 9, pp. 2513-2530, 2009.
22. T. Staebelin and K. Aoki, Planning and scheduling in the automotive industry: A comparison of industrial practice at German and Japanese makers, *International Journal of Production Economics*, 2014.

23. H. Meyr, J. Rohde y H. Stadtler, Basics for Modelling, de *Supply Chain Management and Advanced Planning*, Berlin, Germany, Springer Berlin Heidelberg, 2002, pp. 45-70.
24. M. Badell, J. Romero, R. Huertas y L. Puigjaner, Planning, scheduling and budgeting value-added chains, *Computers & Chemical Engineering*, vol. 28, n° 1-2, pp. 45-61, 2004.
25. Y. H. Lee, C. S. Jeong y C. Moon, Advanced planning and scheduling with outsourcing in manufacturing supply chain, *Advanced planning and scheduling with outsourcing in manufacturing supply chain*, vol. 43, n° 1-2, pp. 351-374, 2002.
26. R. Masuchun, W. Masuchun and T. Thepmanee, Integration m-Machine Scheduling into MRP, in *Fourth International Conference on Innovative Computing, Information and Control (ICICIC)*, Kaohsiung, 2009.
27. P. Kelle and A. Akbulut, The role of ERP tools in supply chain information sharing, cooperation and cost optimization, *International Journal of Production Economics*, Vols. 93 - 94, pp. 41-52, 2005.
28. I. Van Nieuwenhuysse, L. De Boeck, M. Lambrecht y N. J. Vandaele, Advanced resource planning as a decision support module for ERP, *Computers in Industry*, vol. 62, n° 1, pp. 1-8, 2011.
29. M. Gen, L. Lin y H. Zhang, Evolutionary techniques for optimization problems in integrated manufacturing system: State-of-the-art survey, *Computers & Industrial Engineering*, vol. 56, n° 3, pp. 779-808, 2009.
30. C. Moon y Y. Seo, Evolutionary algorithm for advanced process planning and scheduling in a multi-plant, *Computers & Industrial Engineering*, vol. 48, n° 2, pp. 311-325, 2005.
31. I. Sommerville, *Software Engineering*, 8th ed., University of St. Andrews, United Kingdom: Pearson Education, 2006.
32. C. Burnay, I. Jureta y S. Faulkner, What stakeholders will or will not say: A theoretical and empirical study of topic importance in Requirements Engineering elicitation interviews, *Information Systems*, vol. 46, p. 61-81, 2014.
33. ISA, ANSI/ISA-95.00.01-2000. Enterprise-Control System Integration. Part 1: Models and terminology, ISBN: 1-55617-727-5, 2000.
34. J. Wei y Y.-S. Ma, Design of a feature-based order acceptance and scheduling module in an ERP system,» *Computers in Industry*, vol. 65, n° 1, pp. 64-78, 2014.
35. K. J. Chen y P. Ji, «A genetic algorithm for dynamic advanced planning and scheduling (DAPS) with a frozen interval, *Expert Systems with Applications*, vol. 33, n° 4, pp. 1004-1010, 2007.
36. D. E. Shobrys y D. C. White, Planning, scheduling and control systems: why can they not work together, *Computers & Chemical Engineering*, vol. 24, n° 2-7, pp. 163-173, 2000.
37. Committee ISO/IEC JTC 1/SC 7, «ISO/IEC 25010:2011 - Online Browsing Platform,» 2011. [En línea]. Available: <https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en>. [Último acceso: 2014].
38. A. Gunasekaran y E. W. T. Ngai, Information systems in supply chain integration and management, *European Journal of Operational Research*, vol. 159, n° 2, pp. 269-295, 2004.
39. Microsoft Patterns & Practices Team, *Microsoft® Application Architecture Guide (Patterns & Practices)*, 2nd ed., O'Reilly, 2009.
40. R. Lothka, Should all apps be n-tier?, 2005. [Online]. Available: <http://www.lhotka.net/weblog/ShouldAllAppsBeNtier.aspx>. [Accessed 12 2014].
41. P. Kruchten, The 4+1 View Model of Architecture, *IEEE Software*, vol. 12, n° 6, pp. 42-50, 1995.