

Evaluación de una Arquitectura de Referencia con Alumnos Avanzados de la Carrera Ingeniería en Sistemas de Información

Melina Vidoni, Jorge Marcelo Montagna, Aldo Vecchietti
Instituto de Desarrollo y Diseño, INGAR CONICET-UTN
Avellaneda 3657

{melinavidoni, mmontana, aldovec}@santafe-conicet.gov.ar

Resumen

Las Arquitecturas de Referencia (AR) proveen lineamientos generales para enmarcar tipos de sistemas. Desde hace unos años, la academia ha generado diversas AR, siempre orientadas a distintos dominios. Un caso particular es el de una AR enfocada en Sistemas de Planeamiento Avanzado (APS). Sin embargo, se necesita evaluar que la definición de la AR sea consistente, al validarla contra los Requerimientos Funcionales y Atributos de Calidad que le dieron origen. No obstante, en la actualidad no existen métodos específicos que se ajusten a las necesidades particulares de las Arquitecturas de Referencia. Por esto mismo, este artículo propone una modificación del método ATAM (Architecture Trade-off Analysis Method), con el objeto de adaptarlo a características propias de las AR, tales como la abstracción y la falta de stakeholders definidos, y en particular enfocada a APS. Para evaluar la validez del método propuesto, éste fue utilizado en una materia electiva de 5° año de la carrera Ingeniería en Sistemas de Información. Los alumnos elaboraron un trabajo práctico de varias entregas y puestas en común grupales. Esto permitió generar un interesante conocimiento en cuanto a las ventajas y desventajas de la metodología propuesta y, fundamentalmente, una experiencia muy importante en cuanto al planteo, tratamiento y evaluación de este tema en un curso de grado.

1. Introducción

En los últimos años, las Arquitecturas de Referencia (AR) han emergido como generalizaciones de arquitecturas concretas que buscan clarificar los límites y alcances de un tipo de dominio de sistemas [1]. Las AR están basadas en funcionalidades y flujos de datos genéricos que definen un tipo o dominio de sistemas [2]. Como consecuencia, son generadas para facilitar el

diseño y desarrollo en múltiples proyectos, por lo que trabajan en un contexto más amplio, más abstracto y menos definido, con stakeholders delimitados sólo como grupos objetivo [3].

Varios autores han utilizado Arquitecturas de Referencia para enmarcar distintos tipos de sistemas. Theilmann y otros [4] detallan la concepción de una AR para administrar un *framework* de SLA (*Service Level Agreement*) multi-nivel, construida sobre un análisis experimental previo. Norta y otros [5] propusieron otra AR, pero enfocada a proyectos Business-To-Business, la cual validaron con una metodología basada en escenarios. Pääkkönen y Pakkala [6] se fundamentaron en arquitecturas existentes para generar una AR para sistemas de *big-data*, para la cual proveyeron una clasificación de posibles tecnologías de implementación. Behere y colaboradores [7] introdujeron una AR para implementar conducción cooperativa en automóviles modernos, para la cual definieron minuciosamente los servicios y requerimientos arquitectónicos necesarios. Finalmente, Vidoni y Vecchietti [8] elaboraron una AR para Sistemas de Planeamiento Avanzado (APS), llamada APS-RA, la cual generaron a partir de Requerimientos Funcionales y Atributos De Calidad que elicitaban de trabajos académicos e implementaciones reales.

Sin embargo, las arquitecturas deben evaluarse para validar que realmente permitan implementar los Requerimientos Funcionales (RF) y Atributos de Calidad (AC) que fueron elicitados en la etapa de análisis [9]. Una evaluación es una forma sencilla de evitar los altos costos de una implementación que no se corresponde con lo solicitado por los *stakeholders* [10]. Una arquitectura que atraviesa exitosamente un proceso de evaluación provee los fundamentos necesarios para el desarrollo de sistemas de información de alta calidad [11].

Actualmente, existen varios métodos para evaluar arquitecturas del software [9, 12, 13]. Sin embargo, se enfocan mayormente en arquitecturas concretas, y no

AR, las cuales tienen varias diferencias conceptuales, además de rasgos inherentes al dominio que definen.

En la literatura existen algunas propuestas para modificar los métodos de evaluación. En 2008, Angelov y colaboradores [11] elaboraron una adaptación a ATAM (*Architecture Trade-off Analysis Method*) modificando las actividades existentes y agregando extensiones, siempre con el objetivo de adecuarlo a las AR. Por otro lado, Martínez-Fernández y otros [14] desarrollaron un *framework* empírico propio enfocado a firmas de desarrollo de tecnología de la información. Finalmente, Galster y Avgeriou [15] emplearon otro enfoque, y propusieron una evaluación previa al desarrollo para generar arquitecturas de referencia con sustentos teóricos, y aptas para su continua reutilización.

Sin embargo, no todas las AR pueden ser evaluadas con estos métodos. Este es el caso de APS-RA, la Arquitectura de Referencia para Sistemas de Planeamiento Avanzado propuesta por Vidoni y Vecchietti [8], la cual considera situaciones específicas de dichos sistemas.

Como consecuencia de esto, este trabajo avanza sobre la modificación a ATAM propuesta por Angelov y colaboradores [11], y realiza cambios concretos para el tipo de AR específica a ser evaluada.

La evaluación de APS-RA empleando ATAM modificado se realiza en una materia electiva de 5° año de la carrera Ingeniería en Sistemas de Información (ISI). Para esto, se elabora un trabajo práctico que consiste de varias entregas por parte de los alumnos, y sucesivas puestas en común. Esta actividad genera resultados muy interesantes, los cuales son tratados en este trabajo.

En primer lugar permite considerar la metodología más apropiada para el tratamiento de Arquitecturas de Referencia y su evaluación, en el marco de una cátedra de la carrera ISI. Se trata de una temática apropiada para alumnos del último año, pues ayuda a consolidar conocimientos de otras asignaturas, al integrarlos y aplicarlos en un caso de estudio que deriva del análisis de la industria. Además, conlleva una importante interacción con los alumnos para lograr conocer el grado de asimilación respecto a metodologías para el desarrollo de Sistemas de Información. En el transcurso del dictado del curso se evalúa la metodología utilizada al considerar aspectos tales como: características del trabajo práctico a desarrollar, tipo de problema a tratar, seguimiento de actividades, metodología de evaluación, etc.

Como un segundo resultado importante, se genera una evaluación rica y productiva de la AR en estudio, logrando, además, establecer conclusiones significativas sobre la metodología propuesta para evaluar Arquitecturas de Referencia.

Finalmente, el artículo está organizado de la siguiente forma. La Sección 2 presenta las diferencias de las AR respecto de las arquitecturas concretas, y analiza sus

ventajas; sus subsecciones examinan las particularidades de APS-RA. A continuación, la Sección 3 presenta el razonamiento para la modificación realizada a ATAM. Luego, la Sección 4 describe en detalle la planificación de la metodología para una cátedra de grado. A continuación, la Sección 5 se enfoca en la evaluación efectuada por los alumnos, cómo se organiza el trabajo y los resultados obtenidos. Por último, la Sección 6 despliega las conclusiones y trabajos futuros.

2. Arquitecturas de Referencia (AR)

Una arquitectura del software (referidas en este trabajo como *arquitecturas concretas*) define un conjunto de funcionalidades y menciona características específicas, tanto del negocio como del sistema, para una implementación en particular [1].

Por otro lado, las Arquitecturas de Referencia están basadas en funcionalidades genéricas abocadas a definir y enmarcar un tipo de sistemas [10]. Como consecuencia, hay cuatro grandes diferencias entre ambas [11]:

- Las AR son genérica, y se diseñan para abarcar la mayor cantidad posible de funcionalidades y cualidades deseadas por el más amplio rango de *stakeholders*, en todos sus contextos [7].
- No existe un grupo de *stakeholders* claramente definido [3].
- Debido a la generalidad de las AR, estas poseen un alto nivel de abstracción, y pueden proveer detalles sólo para elementos específicos [16].
- Una AR abarca más cualidades arquitectónicas que una arquitectura concreta, como consecuencia de su generalidad y mayor audiencia [16].

Estas diferencias con las arquitecturas concretas son, específicamente, las generadoras de las grandes ventajas de las AR: favorecen la reutilización, facilitan ahorros de costo y tiempo al reducir el ciclo de vida del desarrollo del software (SDLC, por las siglas en inglés), disminuyen los riesgos e incrementan la calidad de los sistemas generados al contribuir activamente a la evolución de un dominio de sistemas [14]. Como consecuencia, se mejora y asegura la interoperabilidad de los sistemas a través de la estandarización de sus definiciones [15].

El beneficio económico mejor percibido de las AR son los ahorros de costo durante el tiempo de desarrollo y mantenimiento de los sistemas, debido a que sus arquitecturas fueron generadas reutilizando elementos que adoptaron las mejores prácticas de un entorno [17]. Además, las AR benefician la comunicación entre los *stakeholders*, incluyendo desarrolladores y usuarios, beneficiando la distribución del trabajo y disminuyendo los malentendidos sobre el sistema [18].

Sin embargo, para que estas ventajas sean significativas, la AR debe ser consistente con su definición y permitir la implementación de todos sus

Requerimientos Funcionales (RF) y Atributos de Calidad (AC). La mejor forma de asegurar dicha consistencia es a través de la aplicación de una metodología de evaluación.

2.1. APS-RA: Arquitectura de Referencia para Sistemas de Planeamiento Avanzado

Los Sistemas de Planeamiento Avanzado (o APS, por las siglas en inglés), son sistemas de información concebidos para resolver problemas de planeamiento de operaciones a través de la aplicación de métodos de resolución avanzados; esto se realiza al interoperar continuamente con el Sistema Empresarial (SE) de la organización, para alcanzar un flujo de trabajo coordinado [19]. El concepto de *método de resolución* es un término amplio que incluye a la investigación operativa, algoritmos genéticos, simulación, teoría de juegos, entre otros. A su vez, cabe destacar que los APS no buscan reemplazar al planeador humano (su usuario más importante), sino complementarlo y permitirle estar siempre en control de los resultados [20].

Los sistemas de este tipo poseen dos partes [20, 21]: el lado *software-intensive* orientado a la automatización del proceso [22], y el conjunto de modelos que ofrecen posibles formas de resolver los problemas de planeamiento [23, 19]. Por esto, tanto los RF como los AC de este dominio, deben ser definidos en dos partes: aquellos que afectan al sistema, y aquellos que afectan a los modelos [19]. Es importante mencionar que los AC elicidados son acordes al actual estándar ISO/IEC 25010 [24].

Dichos RF y AC fundamentan la base sobre la cual se desarrolla APS-RA, la Arquitectura de Referencia que busca enmarcar y definir a este dominio de sistemas [8]. APS-RA está documentada siguiendo el Modelo de Vistas "4+1" [25], y utiliza diagramas UML 2.x [26] para presentar cada vista. A su vez, sigue el modelo de documentación propuesto por Bachmman [27], e incluye puntos de variación para darle a la AR la capacidad de cambiar rápidamente en formas pre-planificadas [28].

3. Selección de ATAM

En la actualidad existen varios métodos de evaluación de arquitecturas concretas, y a su vez varias propuestas de adaptaciones para AR. Sin embargo, este artículo las ajusta a las necesidades concretas de una AR para APS.

En la literatura académica hay varios análisis comparativos de los métodos de evaluación disponibles [12, 13], como así también casos de estudio de aplicaciones de dichos métodos [5], que permiten obtener una vista comparativa de ellos.

En este caso particular, la Arquitectura de Referencia a evaluar, APS-RA [8], es un gran determinante del método de evaluación a seleccionar, ya que se necesitan

considerar no sólo las peculiaridades de las AR, sino también las del dominio en estudio.

Luego de un análisis de las opciones disponibles, el método seleccionado es ATAM [29]. Esto se debe:

- Permite analizar múltiples Atributos de Calidad y la documentación trabaja con el Modelo de Vistas "4+1" [25], el cual fue empleado para documentar la APS-RA. Además, permite identificar puntos de intercambio y riesgos [9].
- Incorpora la realización de preguntas y técnicas de medición que pueden integrarse fácilmente en el proceso de diseño [11].
- Referencia explícitamente la evaluación de la "integridad conceptual" como una calidad arquitectónica [9, 11].
- Concentra la atención de los evaluadores en la porción de la arquitectura que es central para alcanzar sus metas [30].
- Busca describir tipos de componentes y su tipología, el patrón de diseño de los datos, y el control de interacción entre componentes [5].

A su vez, un aspecto clave, es que existen adaptaciones de este método para adecuarlo a las necesidades de las arquitecturas de referencias [3, 11].

Dichos autores proponen dos tipos de modificaciones a ATAM: adaptaciones, que realizan cambios a actividades existentes, y extensiones, las cuales son pasos adicionales agregados para evaluar aspectos específicos.

En particular, las adaptaciones más relevantes son:

- Definir los grupos de *stakeholders* para el dominio, y trabajar con representantes de dichos roles, que tengan el conocimiento y motivación para contribuir con la evaluación.
- Evaluar sólo las cualidades que pueden ser consideradas con el nivel de abstracción de la AR, y obviar las restantes.
- Priorizar los escenarios al considerarlos bajo contextos específicos (fijar los puntos de variación en una decisión, o emplear un enfoque particular).
- Extender la lista de riesgos para incluir "riesgos de edificabilidad", que existen debido a la necesidad de instanciar la AR en arquitecturas concretas.

Por otro lado, algunas de las extensiones son:

- Valorar la aceptabilidad de la AR, al considerar el tiempo requerido para introducirla a los stakeholders, y su nivel de comprensión como métricas para alcanzar dicha calidad.
- Evaluar la AR *bottom-up* hacia un modelo de referencia de otros autores, que permita estudiar si realmente es posible arribar a la misma AR.
- Estudiar la AR *top-down* hacia arquitecturas concretas, para revelar si la AR soporta las funcionalidades propuestas en concretas, y si puede ser aplicable a casos específicos.

Pese a que estas modificaciones funcionan adecuadamente para un gran número de AR, la existencia de los RF y los AC propios de los APS llama a realizar modificaciones más extensas.

3.1. Adaptación para APS

El impacto de definir los RF y AC tanto para el software como para los modelos, afecta varios aspectos de los APS, incluyendo sus grupos de *stakeholders* [22]. En referencia a esto, Gayialis y Tsiopoulos [31] afirmaron que "[...] *construir este tipo de sistema tiende a ser un proyecto de desarrollo de software que demanda tiempo y presupuesto, y que requiere muchas horas de trabajo de un equipo de programadores de computadoras e investigadores en operaciones*". La cita menciona dos tipos de desarrolladores: los tradicionales que implementan el código del sistema, y especialistas del método de resolución que codifican los modelos para cada problema de planeamiento.

A su vez, cabe destacar al principal usuario de los APS: los planificadores humanos [31, 32, 33, 34]. A menudo poseen gran cantidad de conocimiento sobre los pormenores de la industria, y se pueden beneficiar ampliamente de la implementación de un APS.

Comúnmente, estos *stakeholders* no tienen la misma formación, y sus intereses están centrados en distintas partes del sistema. Esta peculiaridad de los APS, en específico los dos tipos de desarrolladores, presenta una clara necesidad de proponer una modificación más extensiva a los métodos de evaluación [3], en particular a ATAM, que requiere una fuerte presencia de los *stakeholders* durante el proceso.

Entre ellos, los planificadores humanos presentan una mayor dificultad al momento de buscar referentes, ya que su conocimiento está usualmente concentrado en los casos particulares en los que trabajan [33]. Esto afecta negativamente a la evaluación de una AR, ya que se necesita una experiencia más amplia y la capacidad de abstraerse de los casos concretos hacia el dominio [2].

La generalización de resultados se puede alcanzar al usar muestras grandes o participantes con más experiencia [35]. A su vez, las implementaciones de referencia, prototipos y enfoques incrementales pueden ser empleados para validar resultados y comprobación.

Por lo tanto, la metodología propuesta consiste en dos etapas, cada una enfocada a un grupo de desarrolladores diferente. Los participantes convocados deben tener una experiencia notable. Además, las actividades a realizar serán incrementales, y se agregarán extensiones al método que permita evaluar la capacidad de la APS-RA de especificarse para implementaciones concretas, considerando entonces los aportes de los planificadores.

Como consecuencia, las etapas propuestas son:

- **Etapas 1:** Evaluación con Especialistas en Métodos

de Resolución. Debido a que estos *stakeholders* no se enfocan en conceptos específicos de la Ingeniería del Software, no es ideal forzarlos a participar en una evaluación tradicional a riesgo de perder su interés y menguar el aporte que su experiencia puede hacer. Por lo tanto, esta etapa se concentra en analizar cómo interopera el APS con los modelos, los métodos de resolución disponibles, y conceptos exclusivamente ligados a esta relación. Sólo utiliza RF y AC asociados al método de resolución.

- **Etapas 2:** Evaluación con Desarrolladores del Software. Realizan una evaluación más tradicional que considere estilos, tácticas, riesgos, puntos de sensibilidad y *trade-offs* asociados a la arquitectura, con el objetivo de validar que ésta cumpla con los Atributos de Calidad elicitados. Este grupo se enfoca en el APS como un sistema integrado, empleando los RF y AC asociados al sistema.

Esta separación es un muy importante, ya que se asume que diferentes puntos de vista pueden llevar a resultados distintos [35].

4. ATAM-M: Plan de Trabajo en una Cátedra de Grado

La propuesta de adaptación de ATAM para la APS-RA es denominado ATAM-M (siglas de ATAM Modificado), y abarca ambos grupos de evaluación. Sin embargo, el presente artículo se concentra en la Etapa 2, con desarrolladores del software. Por lo tanto, las modificaciones y reformas a ATAM presentadas se abocan exclusivamente a esta fase.

Falesi y colaboradores [35] propusieron trabajar con alumnos, al considerar que muchos de ellos, además de poseer conocimientos específicos sobre el proceso de análisis y desarrollo del software, pueden tener experiencia laboral, en particular si se trabaja con aquellos pertenecientes a los últimos años de la carrera. Esto es importante porque permite incrementar la muestra de participantes generalizando los resultados.

4.1. Organización en Enfoques

Como consecuencia, para poder emular el contexto del mundo real, donde los equipos de desarrollo trabajan en grupos que poseen distintos marcos de acción, se adapta la propuesta de Falesi y otros [35]: diseñar enfoques para los alumnos participantes, los cuales son elegidos voluntariamente por ellos, ya sea por su experiencia personal, preferencias o confianza en el tema.

Considerando la abstracción de la Arquitectura de Referencia, y los aspectos más relevantes de la APS-RA, los enfoques seleccionados son:

- **Administración de Datos (AD):** Centrado en las fuentes de datos, incluyendo su implementación y

mantenimiento. Considera entrada y salida de datos. Se comporta como *database managers*.

- **Administración de Recursos (AR):** Ya que la abstracción de la AR no tener roles acotados al hardware y comunicaciones, este grupo apunta a la disponibilidad, tolerancia a fallos e impacto de los recursos físicos/software del sistema.

- **Desarrollo del Software (DS):** Desarrolladores del sistema, quienes reciben requerimientos, y se abocan al ciclo de vida del sistema, el cual incluye también las modificaciones, puesta en marcha y adaptación de los usuarios. Consideran agregar funcionalidades y adaptaciones del sistema.

- **Integración de Sistemas (IS):** Se centra en cómo el APS interopera con otros sistemas, principalmente el SE. Apunta al intercambio de información y la separación de conceptos.

- **Seguridad y Privacidad (SP):** Se comporta como un administrador de seguridad, priorizando la autenticidad y veracidad de las interacciones y accesos al sistema, estudiando cómo un cambio puede afectar la seguridad, etc.

4.2. Plan de Clases para ATAM-M

ATAM original [29] es una metodología incremental que se realiza en sucesivas iteraciones, refinando las salidas obtenidas previamente. ATAM-M conserva la idea del método original y se organiza para trabajar en sucesivas etapas. En el caso particular de su aplicación con los alumnos, se ha realizado a través de varias clases [5]. Debido a la decisión de emplear enfoques, se proponen dos formas de elaboración:

A. Entregas de Trabajos Prácticos (TP), realizados en grupos. En ellas, los alumnos deben evaluar APS-RA desde el enfoque elegido. Como consecuencia, los resultados e intereses en el sistema varían de grupo a grupo.

B. Evaluaciones Grupales (EG) y *brainstorming*, realizados en horario de clase entre todos los alumnos. Los alumnos deben mantener sus enfoques, pero la presencia de otros grupos abre las situaciones a debate, permitiendo una mirada globalizadora sobre el mismo.

La estructura adoptada para el desarrollo del curso tiene varias etapas, las cuales se detallan la secuencia de tareas propuestas para su aplicación en la cátedra.

Las clases comienzan con una introducción a la APS-RA, orientada a los RF, AC, vistas y puntos de variación, para brindarles a los alumnos el conocimiento necesario sobre el dominio del sistema para que sean capaces de evaluarlo. A partir de esto, se proponen tres entregas de TP, y tres evaluaciones grupales, desarrolladas en tiempo de clase, entre todos los alumnos.

Es importante mencionar que las actividades de

ATAM original pueden ser demasiado extensas para realizarse en un solo TP o EG. Por esto, ATAM-M desagrega dichas actividades entre las dos formas de elaboración posible. Esta decisión facilita la integración de los enfoques con la mirada colectiva a la APS-RA, y simplifica la cantidad de actividades requeridas como entregas.

4.2.1. Primer Trabajo Práctico

En el primer TP, los alumnos deben generar las caracterizaciones de los Atributos de Calidad elicitados para la APS-RA, al proponer preguntas que apuntan a obtener un mejor entendimiento de los AC que afectan al sistema [5]. Las preguntas deben clasificarse en [29]:

- **Estímulo:** eventos que causan que la arquitectura responda o cambie.

- **Respuestas:** cantidades medibles u observables referidas al cambio en la arquitectura.

- **Decisiones:** aspectos de la arquitectura, tales como componentes, conectores, etc., que tienen un impacto directo en alcanzar las respuestas deseadas.

En este punto, y debido a la abstracción de las AR, no todos los AC pueden ser estudiados. Además, las preguntas han de ser acordes al nivel de detalle de la APS-RA, y no incluir números, porcentajes o datos demasiado concretos.

En la segunda parte, los alumnos deben detectar estilos o patrones arquitectónicos, al estudiar los diagramas y puntos de variación que conforman la APS-RA. Estilos y patrones se refieren a soluciones recurrentes que resuelven problemas a nivel de diseño arquitectónico, y proveen un vocabulario común para facilitar la comunicación; además, proveen los medios para razonar los AC y documentar decisiones [36].

4.2.2. Primera Evaluación Grupal

Las actividades a desarrollar se agrupan en tres partes.

En la primera, los alumnos priorizan los AC para cada enfoque basándose en las preguntas generadas en el paso anterior, también detectando aquellos AC que consideran que no pueden evaluar, ya sea por estar fuera del enfoque o depender de detalles específicos de implementación. El objetivo de esta etapa es conocer qué cualidades prevalecen en cada orientación, y delimitar aspectos que están fuera del alcance de una AR [11]. Esta es una breve exposición que debe realizarse grupo a grupo.

Luego de esto, se trabaja en un *brainstorming* grupal, para distinguir los *trade-offs* existentes en los estilos detectados. En Arquitectura del Software, favorecer un Atributo de Calidad conlleva inevitablemente perjudicar otro; esto se conoce como *trade-off* [37]. Detectarlos en la etapa de diseño permite reducir costos durante la implementación y funcionamiento del sistema, además de aumentar la validez del mismo [38].

Para trabajar en esto, se propone emplear la

nomenclatura propuesta por Harrison y Avgeriou [39], debido a su simplicidad y facilidad de comprensión. La propuesta es lograr una lista de *trade-offs* entre todos los alumnos, quienes siempre deben mirar al sistema desde su enfoque. La lista elaborada debe ser visible para todos, para favorecer la discusión.

Finalmente, en la última parte de esta evaluación, los alumnos deben identificar riesgos y puntos de sensibilidad asociados a los estilos identificados. Los riesgos son disposiciones arquitectónicas potencialmente problemáticas, compuestas de decisiones aún no realizadas, la respuesta de un AC y sus consecuencias, y la lógica para el efecto negativo [10]. Por otro lado, los puntos de sensibilidad son propiedades de uno o más componentes (o relaciones entre ellos) que son críticas para alcanzar una respuesta particular de un AC [10].

Esta actividad se realiza como una *brainstorming* entre, todos los enfoques. Como consecuencia, tanto los riesgos como los puntos de sensibilidad logran ser más amplios y abarcar múltiples aspectos de la AR, concentrándose en tantos enfoques como sea posible.

4.2.3. Segundo Trabajo Práctico

En ATAM original, el árbol de utilidad busca generar escenarios que evalúen “cuán buena es” la arquitectura para el sistema y el usuario [29]. Sin embargo, esto requiere de decisiones concretas que no pueden realizarse en una AR [2]. Como consecuencia, algunos trabajos académicos proponen apreciar otras características, acordes a la abstracción de una AR, y que permiten estudiar cómo aplicar los AC en casos concretos [3].

Por esto mismo, se propone que los alumnos realicen dos árboles, cada uno con raíces diferentes, y orientados a distintos tipos de escenarios:

- **Utilidad:** es el árbol original de ATAM, que busca proveer un mecanismo *top-down* para directa y efectivamente trasladar los motores del negocio a escenarios de AC [5]. Con la modificación que se propone para ATAM-M, estos escenarios deben encauzarse en los “escenarios de caso de uso”, los cuales expresan deseos específicos de un grupo de stakeholders en particular [29]. Es importante mencionar que dichos escenarios tendrán un nivel de detalle acorde al de la APS-RA.
- **Edificabilidad:** propuesto en ATAM-M, busca evaluar la capacidad de la AR para instanciarse y especificarse en casos concretos, así como la flexibilidad de sus puntos de variación. Los nodos generados son similares a los “escenarios de crecimiento”, que representan cambios típicos anticipados por el sistema, y a los “escenarios exploratorios”, que empujan al sistema y lo estresan ante situaciones de cambio extremo [29].

La evaluación a través del árbol de edificabilidad se basa en la adaptación sugerida en la literatura [11]:

estudiar la Arquitectura de Referencia desde distintos contextos, y considerando más atributos de la arquitectura como un todo, tales como la capacidad de concretizarse, la aplicabilidad, etc.

Los escenarios del árbol de edificabilidad deben generarse al considerar un contexto en particular: es decir, suponer que un punto de variación toma una forma específica, y generar escenarios al respecto.

Cabe destacar que como APS-RA ya utiliza la terminología “escenarios”, en sus Requerimientos Funcionales y en el Modelo de Vistas “4+1”, se decidió nombrar a los escenarios de los árboles de utilidad y edificabilidad como *nodos*. El objetivo de esto es no agregar más significados a las palabras, y evitar malentendidos entre los participantes de ATAM-M.

En ambos árboles, los nodos se priorizan mediante la evaluación de su rango. Éste se compone de prioridad y dificultad, clasificadas como alto, medio o bajo.

Además, el Segundo TP también incluye la detección y propuesta de tácticas. Los patrones o estilos arquitectónicos empaquetan tácticas, las cuales son decisiones de diseño que influyen el control sobre la respuesta de un AC [40]: son transformaciones que están disponibles para afectar el comportamiento de un diseño con respecto a un atributo de calidad en particular [41].

Dependiendo del enfoque, algunas tácticas podrán ser detectadas en los diagramas o puntos de variación, pero otras deben ser propuestas por los alumnos para ser utilizadas en implementaciones concretas de la APS-RA.

Las tácticas tienen que ser aplicadas a los nodos de los árboles que tengan un mayor rango (prioridad y dificultad alta), con el objetivo de comprobar que las mismas permiten efectivamente realizar dichos nodos.

4.2.4. Segunda Evaluación

Los participantes exponen brevemente las principales tácticas encontradas, y a qué apuntan con las mismas. Dicha exposición es ante el resto de los participantes, favoreciendo un ambiente de preguntas y respuestas desde los otros *stakeholders* hacia el grupo que expone.

Sin embargo, la parte principal de esta evaluación es la generación de escenarios de nivel dos, los cuales serán denominados *situaciones* ya que la APS-RA emplea el término escenarios para conceptos asociados al *planning*.

Las situaciones deben generarse a modo de *brainstorming* y entre todos los participantes, donde cada uno aporta a cada situación una característica propia de su enfoque. Por lo tanto, las situaciones son mucho más amplias y deben comprender aspectos pertinentes a todos, o casi todos, los enfoques.

Es importante destacar que si bien la generación de los árboles facilita comprender cómo se administraron los motores de los AC al diseñar la arquitectura, el análisis de situaciones busca estudiar cómo los *stakeholders* perciben la AR y al sistema en general [29].

Nuevamente, las situaciones deberán ser expuestas de forma que todos los participantes pueden observarlas a medida que se desarrollan.

4.2.5. Tercer Trabajo Práctico

La primera parte de esta entrega inicia el análisis de estilos y tácticas. Por lo tanto, se enfoca en generar preguntas de nivel dos, asociadas a las tácticas detectadas previamente. Dichas preguntas buscan descubrir riesgos, no-riesgos y puntos de sensibilidad asociados a las tácticas. Además, deben ser nuevamente clasificadas en estímulo, respuesta o decisión.

La segunda parte de esta entrega trabaja sobre las situaciones generadas en la Segunda Evaluación. A cada grupo se les asignará una cantidad de votos equivalente al 30% del total de situaciones generadas [29], que son usados para priorizar las situaciones. El criterio empleado debe ser siempre el enfoque del grupo, y lo que los integrantes consideran preferente para el mismo.

Finalmente, la última parte compara las situaciones marcadas como prioritarias con los nodos de los árboles de utilidad y edificabilidad generados en la etapa previa. Aquí se cotejan el contexto que definen tanto la situación como el nodo, y también posibles discordancias en la prioridad y rangos asignados. Cualquier diferencia debe ser reconciliada, al clarificar el significado del nodo o cambiar un rango, o explicada, al comprender el criterio empleado para priorizar y definir un rango [10].

Esta última actividad implica modificar los árboles de utilidad al agregar nodos o cambiar sus rangos. Dichos refinamientos son esperados e incentivados.

4.2.6. Tercera Evaluación

Esta evaluación representa el cierre de ATAM-M y la exposición final entre todos los participantes. Se encuentra dividida en dos actividades principales.

Primero, cada grupo ofrece su visión de cómo asocian los estilos encontrados, mientras que los otros enfoques aportan su propia visión. Esto se elabora en una forma visible para todos los participantes.

Segundo, a través de una discusión entre todos los participantes, se elaborará una lista de riesgos, puntos de sensibilidad y *trade-offs* asociados a las tácticas. Los *stakeholders* pueden generar estos elementos a partir de las preguntas de nivel dos, con el objetivo de estudiar la AR y descubrir elementos importantes que pueden tener efectos adversos en la misma.

5. Aplicación y Evaluación con ATAM-M

Para aplicar ATAM-M se trabaja en la cátedra electiva “Diseño de Software Basado en Arquitectura” de 5º año de la carrera Ingeniería en Sistemas de Información, de la UTN Facultad Regional Santa Fe.

La evaluación se presenta como el TP final de la

materia. Consiste de una clase inicial de exposición de la APS-RA, tres entregas elaboradas por los alumnos y tres evaluaciones grupales. Se incluyen cinco clases de consulta presencial. La duración total es de un mes.

En total, la cantidad de participantes asciende a 10, y aproximadamente la mitad de ellos se encuentra trabajando en empresas orientadas al software; además, dos de los restantes participan de Becas I+D (Investigación y Desarrollo) de pregrado ofrecidas por la Facultad. La evaluación cuenta con tres participantes adicionales: el arquitecto que guía la evaluación, y dos profesores de la materia, que participan en las evaluaciones con el objetivo de favorecer el diálogo y ofrecer puntos de vista diferentes.

El material se gestiona a través del Campus Virtual. Todo lo específico a la APS-RA es introducido como lectura obligatoria, pero también se ofrecen artículos complementarios. Sobre la AR a analizar, se incluye:

- Glosario de términos y definiciones.
- Lista de RF y AC.
- Todos los diagramas de todas las vistas.
- Vista a vista, descripción de todos los elementos y los puntos de variación existentes.

Respecto a esto, a los alumnos se les ofrecen plantillas de hojas de cálculo para facilitarles la generación de algunos pasos de la metodología; el resultado obtenido es considerablemente más ordenado. Además, para trabajar con las EG, se proyecta en una pantalla una otra planilla, que se completa a medida que avanza el *brainstorming*. De este modo, todos los alumnos pueden observar los resultados que produce dicha puesta en común.

Durante la evaluación, los alumnos trabajan los AC, estilos y tácticas, por sus nombres en inglés. Esto se debe a que el material de estudio propuesto por la cátedra incluye apuntes en español y en inglés. Además, respecto a los AC, las traducciones al español suelen ocasionar problemas de comunicación, y por eso se decide continuar con los nombres originales.

Finalmente, al finalizar la evaluación los alumnos completan una encuesta anónima sobre el método, para ofrecer sus comentarios y opiniones.

5.1. Resultados Obtenidos

Todos los enfoques se encuentran representados. Según la encuesta anónima, la mayoría de los alumnos selecciona un enfoque basado en su experiencia personal y/o interés en el área. Esto puede comprobarse con los resultados obtenidos, ya que muchas veces se trata de conocimientos específicos del área en cuestión.

5.1.1. Primer Entrega y Evaluación

Respecto a esta etapa, entre todos los enfoques, los alumnos elaboran 121 preguntas, de las cuales 21 tienen doble clasificación (por ejemplo, estímulo y respuesta, o

estímulo y decisión). La Tabla 1 resume la cantidad de preguntas por clasificación, y el porcentaje del total que representan.

Tabla 1. Estadísticas Generales de Preguntas (Nivel 1).

	# Preguntas	% Preguntas
Total	121	
Estímulo	48	33,80%
Respuesta	69	48,59%
Decisión	25	17,61%
Evaluación	142	100,00%

Un punto importante a destacar es que los alumnos dirigen las preguntas del Primer TP hacia los atributos que, durante la Primera EG justifican como relevantes para su enfoque. Algunas de las preguntas generadas son:

- ¿Existirá algún tipo de redundancia de servidores que permita ofrecer una disponibilidad en caso de fallas? (Decisión).
- ¿Si el sistema es web y el modelo corre en el servidor, pueden dos usuarios diferentes ejecutar el mismo modelo al mismo tiempo? (Estímulo).
- ¿Qué datos manejados en el sistema/canales serán considerados sensibles para el negocio? (Respuesta)

A pesar de que se presentan preguntas sobre *Usability*, es importante mencionar que durante la EG, los alumnos consideran que este atributo debe ser evaluado en mayor detalle por el Analista y Diseñador del Sistema a cargo de cada implementación en particular.

Por otro lado, en la identificación de estilos hay similitudes entre todos los enfoques. Asimismo, durante la Primera EG los alumnos deciden, por iniciativa propia, asociar varios estilos ya que los consideraban variantes directamente atadas a los puntos de variación de la AR.

Entre los estilos más mencionados, se encuentran *Layers*, observado en la Vista Lógica, *Shared Repository*, en la Vista de Desarrollo y Lógica, *Model-View-Controller (MVC)* en la Presentation Layer de la Vista Lógica, *Service Oriented Architecture (SOA)*, en la Vista de Desarrollo, y *Data Abstraction*, en la Vista de Escenarios. Por otro lado, algunas de las similitudes destacadas son *Client Server* junto con *Multi-Tiers* y *Publish Subscribe*, mayormente hallados en la Vista Física. Estos estilos son sólo los más mencionados por los alumnos en la evaluación.

Una de las salidas más importantes de la Primera EG es la obtención de una lista de *trade-offs* para cada estilo, considerando los atributos que afecta, positiva y negativamente, y aquellos en los que no influye.

Como se mencionó previamente, esta etapa emplea la notación de Harrison y Avgeriou [39], donde ++ significa que el AC favorece ampliamente un estilo, y – es el caso opuesto. Sin embargo, durante la discusión los mismos alumnos encuentran que dicha nomenclatura no es suficiente para detallar la influencia de los estilos en las condiciones específicas de las AR. En particular, respecto

a situaciones donde las consecuencias de un estilo dependen de los puntos de variación de la documentación de la APS-RA. Para estos casos, los alumnos proponen agregar la simbología \pm , considerando necesario justificar algunas evaluaciones y situaciones donde no detectan influencia del estilo en un determinado AC.

La Tabla 2 muestra los resultados de los *trade-offs* entre estilos y AC. Las celdas con fondo gris son aquellas que los alumnos consideran necesario explicar.

Hay varias justificaciones, pero la más importante es la evaluación de *Security* y sus sub-atributos con el estilo *Shared Repository*, ya que inicia el debate y la adaptación de la nomenclatura. Los alumnos concluyen que cómo el estilo afecta al AC depende la perspectiva: si hay un solo punto de entrada al sistema se puede concentrar la seguridad y facilitarla, pero si esta es pobre o mal implementada, el sistema en su totalidad es ampliamente vulnerable.

Asociados a los estilos, los alumnos detectan 20 situaciones de riesgo, relacionadas a los mismos. Para no extender excesivamente el artículo, a continuación se detallan sólo algunas, a modo de ejemplo:

- **Layers:** Dependiendo de la implementación, la lógica se puede concentrar en las capas inferiores. Esto puede resultar en un mayor número de transacciones (lo que conlleva demoras), afectando la *Performance Efficiency* y el *Time Behavior* del sistema. A su vez, puede afectar la *Modularity* de las capas inferiores, ya que puede resultar en un acoplamiento indeseado de componentes.
- **SOA:** No se ha definido un nivel de exposición o disponibilidad para los servicios. Por esto, las repercusiones hacia otros componentes no son claras, y se puede afectar la *Interoperability* y *Compatibility* del sistema. La especificación de los servicios debe ser clara para permitir una correcta comunicación entre todos los sistemas del ambiente.

Respecto de los puntos de sensibilidad e intercambio, se detectaron 17, asociados directamente a los estilos; entre ellos, 8 son puntos de intercambio que afectan a más de un AC. Sólo se detallan algunos de los hallados:

- Tiempo requerido por el equipo de desarrolladores para agregar un nuevo Punto de Optimización. **Sensibilidad.**
- Cantidad de conexiones máximas permitidas en un período de tiempo determinado. **Intercambio.**

5.1.2. Segunda Entrega y Evaluación

En esta etapa los alumnos confeccionan los árboles de utilidad y edificabilidad, y luego detectan y proponen tácticas para los nodos con mayor rango.

Respecto al árbol de utilidad, los alumnos generan 87 nodos, y tras la comparación de la tercera entrega sólo modifican 9 nodos.

Tabla 2. Resultado del brainstorming para encontrar los *trade-offs* en los estilos detectados

Atributos de Calidad	Estilos							
	Layers	Pipes & Filters Peer-to-Peer	Shared Repositories	Client Server Multi-Tier Publish Subscribe	MVC	Explicit Invocation	Data Abstraction	SOA
Compatibility				++				
Interoperability			+	++				++
Co-Existence								+
Portability	++	++		+	+		±	
Adaptability	++	++			+		+	
Reliability			-					
Availability	++	++	--	±		±		-
Fault Tolerance	-		--	+				
Maintainability	++		-	+	++	-	++	+
Modularity	++		-		++		++	+
Modifiability	+	++	-		++		++	+
Usability					+	+		
Learnability						+		
Operability						+		
Accessibility						+		
Functional Stability					+		++	+
Correctness		-			+			
Security	-		±	-				--
Confidentiality	-		±					--
Integrity			±					--
Authenticity			±					--
Performance Efficiency	--	-	+	±	-	-	-	±
Time Behavior	--	--		±	-	-	-	
Resource Utilization	--						-	

La Tabla 3 presenta algunos datos estadísticos por enfoque respecto de este árbol.

Tabla 3. Estadísticas del árbol de utilidad por enfoque.

Enfoque	Nodos Totales		Nodos Modificados	
	#	%	#	%
Administración de Datos	14	16,09%	0	0,00%
Admin. de Recursos	22	25,29%	5	55,56%
Desarrollo del Software	18	20,69%	0	0,00%
Integración de Sistemas	19	21,84%	3	33,33%
Seguridad y Privacidad	14	16,09%	1	11,11%
	87	100,00%	9	100,00%

Individualmente, los AC con más nodos en el árbol son *Availability* y *Resource Utilization*. Sin embargo, *Security* es el que más nodos tiene entre todos sus sub-atributos. Algunos de los nodos generados, son:

- Ocurre una excepción no planificada durante tiempo de ejecución, y es atrapada por el sistema. Luego de ser administrada, el sistema restablece al comportamiento normal en X minutos. El sistema no congela la interfaz y se guarda un log de errores. Prioridad alta, Dificultad media.
- Múltiples usuarios acceden a la edición del mismo

modelo y el APS termina informando un mensaje a ambos y bloqueando la edición al que accede más tarde a la acción. Prioridad alta, dificultad media.

En el árbol de edificabilidad, los alumnos escribieron 64 nodos, pero sólo 7 fueron modificados posteriormente a través de la comparación. La Tabla 4 muestra datos estadísticos por enfoque respecto de este árbol.

Tabla 4. Estadísticas del árbol de edificabilidad por enfoque

Enfoque	Nodos		Adaptaciones	
	#	%	#	%
Administración de Datos	6	9,38%	0	0,00%
Admin. de Recursos	9	14,06%	0	0,00%
Desarrollo del Software	14	21,88%	1	14,29%
Integración de Sistemas	19	29,69%	3	42,86%
Seguridad y Privacidad	16	25,00%	3	42,86%
	64	100,00%	7	100,00%

En este árbol, los alumnos se abocaron a evaluar Atributos diferentes. El que más nodos posee es *Interoperability*, seguido de *Modularity* y *Security*. Esto demuestra que al cambiar la perspectiva de análisis del dominio de sistemas, también se modifica la prioridad de los AC. Algunos nodos, son:

- Se decide agregar X servidores de aplicación replicados. Las rutinas o funciones de chequeo de

seguridad (autenticación, autorización, validación de entradas, etc) también son aplicadas en estos servidores. Prioridad alta, dificultad media.

- X cantidad de usuarios conectados al sistema intentan recuperar información almacenada en la fuente de datos del APS. Esta permite el acceso simultáneo de lectura de X usuarios sin generar cuellos de botella ni bloqueos de recursos (memoria, procesador, acceso a disco) del sistema. Prioridad y dificultad media.

En la última parte, los alumnos identifican tácticas para la APS-RA, ya sea en los diagramas, los RF o los puntos de variación. De no poder encontrar tácticas, algo que puede suceder debido a la abstracción de las AR, los alumnos proponen las que consideren aplicables.

Sin embargo, el análisis de las tácticas se inicia a través del estudio de los nodos de rango alto (prioridad y dificultad alta, o prioridad alta y dificultad media). En esta etapa, se analizan 35 nodos y se identifican y/o proponen 97 tácticas. Este número incluye coincidencias, identificando la misma táctica múltiples veces, o distintos grupos concordando en la propuesta. Como consecuencia, de las 97 tácticas, 49 son coincidencias.

La Tabla 5 resume las tácticas que, entre los resultados globales, tienen mayor número de apariciones.

Respecto a los AC, se encontraron 32 de 97 tácticas que pertenecen a *Security*, representando el 32,99% del total. Luego, *Maintainability* cuenta con 24 de 97 identificaciones, representando el 24,74% del total.

Tabla 5. Tácticas con mayor preponderancia en las identificaciones.

Tácticas	Apariciones	
Increase Cohesion: Maintain Semantic Coherence	7	7,22%
Fault Recovery: Preparation and Repair: Active Redundancy	5	5,15%
Control Resource Demand: Bound Execution Times	4	4,12%
Increase Cohesion: Abstract Common Services	4	4,12%
Recover from Attack: Maintain Audit Trail	4	4,12%
	24	24,73%

La segunda EG genera las situaciones (o escenarios de nivel dos). Esto se hace a través de una discusión grupal, completando aquellas encontradas en una hoja de cálculo, proyectada para toda la clase. Las situaciones son más amplias que los escenarios, dado que incluyen aspectos para todos los enfoques. De este modo, se elaboraron 17 situaciones, y en promedio, cada una afecta a 7 AC. Algunas de ellas son:

- Se desea agregar un IDS (*Intermediate Data Source*). Se requiere encriptar canales de comunicación y la fuente de datos. Solo el "Data Layer" es afectado. La cantidad de accesos debe ser

limitada. La conexión no se convierte en un cuello de botella. El IDS ofrece una interfaz para exponer los servicios. Una vez puesto en ejecución, el APS puede acceder a los datos provistos por el IDS.

- Se agrega un nuevo Punto de Optimización (PO) de logística con nuevas conexiones de GPS remotas. Los sistemas interpretan y reciben los datos del GPS. Las conexiones son realizadas a través de HTTPS y los datos son analizados. El sistema sólo recibe los datos de dispositivos conocidos. El tiempo de desarrollo no es mayor a X semanas. Los cambios en la "Data Source" mantienen la consistencia. La disponibilidad del nuevo PO es consistente con la de los otros PO.

5.1.3. Tercer Entrega y Evaluación

En la primer parte del Tercer TP, los alumnos generan preguntas de nivel dos, asociadas a las tácticas. Son clasificadas en estímulo, respuesta o decisión, pero apuntan a encontrar riesgos, no-riesgos, puntos de sensibilidad e intercambio en las tácticas.

En total, se generan 163 preguntas, de las cuales aproximadamente 40 tienen una clasificación doble (por ejemplo, estímulo y respuesta al mismo tiempo, o decisión y estímulo, etc.). La Tabla 6 provee estadísticas generales de las preguntas generadas.

Tabla 6. Estadísticas generales de preguntas (nivel 2).

	Preguntas		Total
	#	%	
Estímulo	47	23,38%	201
Respuesta	60	29,85%	
Decisión	94	46,77%	163
Riesgo	59	36,20%	
No-Riesgo	41	25,15%	
Punto de Sensibilidad	39	23,93%	
Punto de Intercambio	24	14,72%	

Algunas de las preguntas (nivel 2) generadas, son:

- ¿Puede un usuario logueado en el sistema acceder a funciones para las cuales no tiene permisos? (Decisión, busca un riesgo).
- ¿El sistema contempla que solo ciertos desarrolladores con permisos puedan realizar modificaciones en los Data Layer? (Estímulo, busca un riesgo).
- ¿Cuántos componentes/módulos son afectados directamente luego de integrar un nuevo Model Solver? (Respuesta, punto de sensibilidad).

Un aspecto interesante para analizar es la participación de cada enfoque en los resultados generales, y en qué tácticas se fundamentan principalmente. La Tabla 7 resume estos conceptos.

Administración de Datos favorece tácticas de *Maintainability* y *Security*, Administración de Recursos se concentra en tácticas de *Reliability*, Desarrollo

Tabla 7. Tácticas favorecidas por los enfoques en las preguntas (nivel 2).

Enfoque	Táctica	Preguntas		
		# Táctica	# Total	% del Total
Administración de Datos	Manage Resources: Maintain Multiple Copies of Data	3	20	12,27%
	Recover from Attack: Restore	3		
Admin. de Recursos	Increase resources: Maintain multiple copies of computations	5	33	20,25%
	Fault prevention: Process Monitor	4		
Desarrollo del Software	Fault Detection: Exception	5	46	28,22%
	Reduce Coupling: Use Encapsulation	3		
Integración de Sistemas	Manage interfaces: Orchestrate	3	24	14,72%
	Usability: Providing Good Help	3		
Seguridad y Privacidad	Resist Attack: Limit Exposure	8	40	24,54%
	Detect Attacks: Sanitize Data	7		

del Software en *Reliability* y *Maintainability*, Integración de Sistemas trabaja con tácticas de *Usability* y *Compatibility*, y finalmente Seguridad y Privacidad se aboca a tácticas pertenecientes al atributo *Security*.

Por otro lado, en la segunda parte de este TP, los alumnos priorizan las situaciones. Para esto, disponen de 6 votos por enfoque (el 30% de la cantidad de situaciones, redondeado hacia arriba). Varios grupos coinciden en las situaciones seleccionadas, aunque varía la asignación de votos, u otorgan prioridades diferentes. Las tres situaciones con más votos, son:

- Se excede la cantidad de clientes que la APS-DS puede soportar para un determinado servicio. Se produce una caída, pero el sistema continúa funcionando con una replicación. El tiempo de latencia y consulta aumenta, la disponibilidad no baja de X%. La performance recae ligeramente. La seguridad del sistema no es comprometida. El sistema vuelve a funcionamiento normal en menos de Y horas. *5 votos, 16,67% del total.*
- Se cambia la implementación del SE a un servidor físico diferente. El APS no es afectado e interactúa con normalidad. El servidor debe tener seguridad física, requerir autenticación, y encriptación. La performance del APS mejora al tener un servidor dedicado. El traspaso no afecta la disponibilidad del sistema. Dependiendo de la ubicación del servidor, puede aumentar la latencia de respuesta. El impacto en el "Data Layer" es mínimo. Los datos persistidos en las diferentes fuentes de datos no se ven afectados. *4 votos, 13,33% del total.*
- El usuario cancela una operación. Los datos que estaban siendo ingresados se descartan y no se implementan en el sistema. Se permite volver a la etapa previa. No representa una vulnerabilidad (los datos son correctamente desechados, y no pueden ser accedidos). Se hace un rollback en cascada a sistemas que le competen. La operación de volver hacia atrás no consume más de X recursos. La fuente de datos permanece consistente y no se guardan datos indeseados. *3 votos, 10% del total.*

En la última etapa de este TP, los alumnos comparan las situaciones que priorizaron con los nodos de alto rango de ambos árboles, buscando diferencias y similitudes. Para esto, deben clasificarlas como acuerdo cuando coinciden en prioridad y descripción, o desacuerdo cuando proveen situaciones opuestas.

Sin embargo, durante la primera consulta presencial, varios grupos deciden proponer una modificación a dicha nomenclatura, y agregar un acuerdo parcial. Esto representa el caso donde el nodo cubre parcialmente la situación, ya que éstas fueron creadas con una mirada global y los nodos desde los enfoques individuales. Generalmente, el resultado lleva a crear un nuevo nodo.

Al momento de justificar, los alumnos pueden: explicar, que involucra comprender que el criterio usado para priorizar los árboles de utilidad/edificabilidad fue diferente que el empleado para priorizar las situaciones, o reconciliar, que busca clarificar el significado de un escenario o cambiar una de las priorizaciones.

La Tabla 8 muestra datos estadísticos de las comparaciones realizadas, y los resultados que arrojan. Como se observa, la cantidad de acuerdos representa un 54,55% del total de comparaciones. Ambos árboles tienen una comparación igualmente representativa.

Tabla 8. Estadísticas generales de comparaciones.

	Comparaciones	
	#	%
Acuerdo	24	54,55%
Acuerdo Parcial	13	29,55%
Desacuerdo	7	15,91%
Edificabilidad	23	52,27%
Utilidad	21	47,73%

En la tercera EG, los participantes discuten la correspondencia de las tácticas (segundo TP) y los estilos (primer TP). En total, se detectan 67 correspondencias de tácticas a estilos, en las cuales algunas tácticas conciernen a varios estilos arquitectónicos. *Layers* es el patrón/estilo con mayor cantidad de asignaciones, ya que tiene 18 tácticas, que representan el 26,87%. A esto le

sigue *Client Server* y estilos similares, con 15 tácticas equivalentes al 22,39%. Finalmente, *Pipes & Filters* tiene 12 tácticas que constituyen el 17,91% del total.

Esta EG está dedicada a estudiar las tácticas. En total, se logran estudiar 34 tácticas, y se detectan 83 riesgos, 61 puntos de sensibilidad y 31 puntos de intercambio. Algunos de ellos, son:

- **Riesgo:** La cantidad de puntos de intrusión es demasiado alto. El sistema asigna muchos recursos de procesamiento a esta detección, y por lo tanto baja la performance en actividades prioritarias.
- **Punto de Sensibilidad:** Nivel de documentación existente sobre los protocolos de restauración.
- **Punto de Intercambio:** Pérdida en el negocio por no tener disponibilidad en la planificación de la producción (tiempo sobre dinero).

También se estudian los *trade-offs* de las tácticas con los AC. Ya que una táctica puede afectar múltiples atributos, de las 185 correspondencias se reconocen 91 influencias positivas, 54 negativas y 40 mixtas.

El concepto de influencia mixta proviene de la modificación realizada a la nomenclatura para estudiar los *trade-offs*. Se corresponde con aquellos AC cuya influencia depende de un punto de variación, o de decisiones concretas no detalladas al nivel de la AR.

5.2. Salidas Generadas

La aplicación de ATAM-M genera varias salidas, que engloban los resultados provistos por cada enfoque y aquellos realizados a través de un *brainstorming*:

1. Preguntas de nivel 1, asociadas directamente a la aplicabilidad de los AC en la APS-RA.
2. Una lista de estilos y tácticas presentes en la APS-RA, junto con la relación entre ambas.
3. Los *trade-offs* de estilos respecto a los AC.
4. Los árboles de utilidad y edificabilidad, que incluyen los cambios realizados luego de la comparación entre nodos y situaciones.
5. Una lista de riesgos y no-riesgos descubiertos, asociados a los estilos y/o a las tácticas.
6. Puntos de sensibilidad e intercambio presentes en el sistema. Estos deben ser asociados a los estilos y/o las tácticas previamente listados.

Los resultados sólo llevaron a realizar cambios menores en APS-RA, corrigiendo la sintaxis para describir elementos de forma que no se ligaran a una implementación concreta, y clarificando relaciones entre componentes en algunos diagramas.

5.3. Feedback de ATAM-M

Los participantes completan una encuesta anónima sobre ATAM-M, realizada para obtener comentarios sobre la propuesta. La misma indaga por qué

seleccionaron el enfoque, consideraciones sobre la dificultad de las entregas, ideas y conclusiones sobre las EG, aspectos de interés personal, calidad y adecuación de la información provista sobre la APS-RA y el material complementario, además de comentarios generales.

Según los resultados, la mayoría de los alumnos elige el enfoque por interés personal, experiencia o afinidad, si bien dos participantes aclaran haberlo hecho “al azar”.

Respecto a la dificultad de las entregas, la mayoría las clasifica como intermedia. Un comentario repetido es que tienen mayores requerimientos a los trabajos habituales, lo que se suplementa con el tiempo provisto en clase para trabajar el tema. Un comentario considera que el formato de planillas para realizar las tareas provee claridad.

Los alumnos aceptaron la bibliografía provista como suficiente, aunque el tamaño de la APS-RA es muy diferente de los casos de estudio sobre los que suelen trabajar. Por esto, una minoría la considera extensa para el tiempo ofrecido para realizar las entregas.

Hay que destacar que los participantes concuerdan en que las evaluaciones grupales y puestas en común son de gran utilidad. Uno de ellos comenta que “*se lograba claridad y se veía el objetivo del trabajo*”, mientras que otro opina “*Creo que lo más productivo fueron las puestas en común para no quedarnos con un único enfoque y aprender a considerar más de un aspecto*”.

En los comentarios generales varios alumnos destacan que aplicar ATAM-M en la APS-RA les permite clarificar y comprender los AC, aprender a identificarlos y a seleccionar tácticas considerando cuáles se afectan. Además, otro alumno destaca la importancia de las consultas y clases prácticas para realizar el TP en clases, como medio principal para resolver las dudas.

El comentario sobre ATAM-M también es importante desde la mirada docente. Los alumnos muestran interés en la temática, participando activamente en las EG. Los trabajos elaborados denotan conocimientos específicos aplicados a la APS-RA y, en la mayoría de los casos, una lectura detallada de la documentación de la misma.

6. Conclusiones

El presente artículo parte de la necesidad de evaluar Arquitecturas de Referencia (AR), en particular una generada para definir Sistemas de Planeamiento Avanzado (APS), denominada APS-RA. Para esto propone ATAM-M, que es una modificación a ATAM (*Architecture Trade-off Analysis Method*), profundizando y especializando la propuesta de otros autores, que adaptaron ATAM para adecuarlo a las AR.

ATAM-M se realiza en dos etapas, enfocadas a grupos de desarrolladores diferentes: aquellos que generan los modelos de optimización de la planificación a través de la aplicación de un método de resolución, y el equipo de desarrollo del software tradicional. Este artículo se

concentra en la segunda etapa, ya que sus participantes son alumnos de la cátedra electiva “Diseño de Software Basado en Arquitectura” de 5º año de la carrera Ingeniería en Sistemas de Información.

Los alumnos trabajan en grupos asociados a un enfoque: contextos específicos desde los cuales mirar la APS-RA. La evaluación se organiza en tres entregas de Trabajos Prácticos (TP), y tres Evaluaciones Grupales (EG), para mezclar las visiones concretas de los enfoques con la mirada amplia del conjunto de participantes.

Aplicar ATAM-M a la APS-RA produce resultados excelentes desde el punto de vista docente. La actividad propuesta permite que los alumnos resuelvan un problema de un grado de complejidad muy alto que requiere utilizar buena parte de los conocimientos adquiridos durante toda la carrera. En primer lugar se trata de un problema de características reales, cuya elaboración ha requerido un esfuerzo muy grande y un análisis extensivo de la literatura y la industria. En muchas circunstancias a nivel académico se trabaja con problemas muy simples que no logran captar las complejidades de un caso real. En segundo lugar, el tema elegido y la cátedra seleccionada ayudan a que los alumnos se muestren interesados desde el comienzo, llevándolos a integrar conocimientos de toda la carrera, con el objetivo de realizar satisfactoriamente el TP.

Finalmente, la metodología adoptada implica el desarrollo de actividades en un ambiente cercano a la realidad, lo que ayuda a los alumnos a tomar contacto con un contexto cercano al profesional. La interacción en grupos y la discusión de los temas, entre otras actividades, permiten el ajuste mutuo entre propuestas y la búsqueda de enfoques integradores y consensuados.

Por todo lo anterior se puede concluir que, desde el punto de vista docente, es una experiencia positiva, que puede ser mejorada en futuros dictados de la materia a partir de los resultados alcanzados en esta primera aplicación.

Por otro lado, también se logra analizar múltiples aspectos de la AR, sin perder de vista su carácter genérico y abstracto, alcanzando un estudio amplio y exhaustivo. Los participantes concluyen que APS-RA facilita los AC con los que fue generada, y se obtienen nodos y situaciones que refuerzan los puntos de variación presentes en la documentación de la misma.

Surgen varios trabajos futuros. Inicialmente, realizar la evaluación de ATAM-M para la primera etapa que involucra a los especialistas que desarrollan los modelos de optimización, para obtener los resultados completos y generales, comparando de ambas etapas. Por otro lado, generar métricas e indicadores a partir de la lista de puntos de sensibilidad e intercambio que ayudan a estudiar, en cada caso concreto, si una implementación cumple con las respuestas esperadas a sus AC. Finalmente, realizar las extensiones a ATAM para AR, y

evaluar APS-RA *bottom-up* hacia un modelo de referencia propuesto por otros autores, y *top-down* hacia un sistema concreto que no desarrollado con la APS-RA.

7. Referencias

- [1] L. Northrop, «Chapter 2. What Is Software Architecture?,» de *Software Architecture in Practice*, 2nd ed., Boston, MA, Addison-Wesley, 2003.
- [2] R. Cloutier, G. Muller, D. Verma, R. Nilchiani, E. Hole y M. Bone, «The Concept of Reference Architectures,» *System Engineering*, vol. 13, n° 1, p. 14–27, 2010.
- [3] S. Angelov, P. Grefen y D. Greefhorst, «A framework for analysis and design of software reference architectures,» *Information and Software Technology*, vol. 54, n° 4, pp. 417-431, 2012.
- [4] W. Theilmann, J. Happe, C. Kotsokalis, A. Edmonds, K. Kearney y J. Lambea, «A Reference Architecture for Multi-Level SLA Management,» *Journal of Internet Engineering*, vol. 4, n° 1, pp. 289-298, 2010.
- [5] A. Norta, P. Grefen y N. C. Narendra, «A reference architecture for managing dynamic inter-organizational business processes,» *Data & Knowledge Engineering*, vol. 91, pp. 52-89, 2014.
- [6] P. Pääkkönen y D. Pakkala, «Reference Architecture and Classification of Technologies, Products and Services for Big Data Systems,» *Big Data Research*, vol. 2, n° 4, pp. 166-186, February 2015.
- [7] S. Behere, M. Törngren y D.-J. Chen, «A reference architecture for cooperative driving,» *Journal of Systems Architecture*, vol. 59, n° 10, pp. 1095-1112, 2013.
- [8] M. Vidoni y A. Vecchiatti, «Towards a Reference Architecture for Advanced Planning Systems,» de *18º International Conference on Enterprise Information Systems – ICEIS*, Roma, Italia, 2016.
- [9] P. Shanmugapriya y R. M. Suresh, «Software Architecture Evaluation Methods – A survey,» *International Journal of Computer Applications*, vol. 49, n° 16, pp. 19-26, 2012.
- [10] P. Clements, P. Kazman y M. Klein, *Evaluating Software Architectures: Methods and Case Studies*, 1st Edition ed., Addison-Wesley Professional, 2002.
- [11] S. Angelov, J. J. M. Trienekens y P. Grefen, «Towards a Method for the Evaluation of Reference Architectures: Experiences from a Case,» de *Lecture Notes in Computer Science*, vol. 5292, R. Morrison, D. Balasubramaniam y K. Falkner, Edits., Paphos, Cyprus, Springer Berlin Heidelberg, 2008, pp. 225-240.
- [12] M. T. Ionita, D. K. Hammer y H. Obbink, «Scenario-Based Software Architecture Evaluation Methods: An Overview,» de *Workshop on Methods and Techniques for Software Architecture Review and Assessment at the International Conference on Software Engineering*, Orlando, Florida, USA., 2002.
- [13] L. Dobrica y E. Niemela, «A survey on software architecture analysis methods,» *IEEE Transactions on Software Engineering*, vol. 28, n° 7, pp. 638-653, 2002.

- [14] S. J. Martínez-Fernández, C. P. Ayala Martínez, J. Franch Gutiérrez, H. Martins Marques y D. Ameller, «A framework for software reference architecture analysis and review,» de *Memorias del X Workshop Latinoamericano Ingeniería de Software Experimental, ESELAW 2013*, Montevideo, Uruguay, 2013.
- [15] M. Galster y P. Avgeriou, «Empirically-grounded reference architectures: a proposal,» de *QoSA-ISARCS '11 Proceedings of the joint ACM SIGSOFT conference*, Boulder, Colorado, USA, 2011.
- [16] D. Nguyen, K. Usbeck, W. M. Mongan, C. T. Cannon, R. N. Lass, J. Salvage, W. C. Regli, I. Mayk y T. Urness, «A Methodology for Developing an Agent Systems Reference Architecture,» de *11th International Workshop of Agent-Oriented Software Engineering*, Toronto, Canada, 2011.
- [17] S. Martínez-Fernández, C. Ayala y X. Franch, «A reuse-based economic model for software reference architectures,» Barcelona, Spain, 2012.
- [18] H. Unphon y Y. Dittrich, «Software architecture awareness in long-term software product evolution,» *Journal of Systems and Software*, vol. 83, n° 11, pp. 2211-2226, 2010.
- [19] M. Vidoni y A. Vecchiotti, «A systemic approach to define and characterize Advanced Planning Systems (APS),» *Computers & Industrial Engineering*, vol. 90, pp. 326-338, 2015.
- [20] J. M. Framinan y R. Ruiz, «Architecture of manufacturing scheduling systems: Literature review and an integrated proposal,» *European Journal of Operational Research*, vol. 205, n° 2, pp. 237-246, 2010.
- [21] 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.
- [22] K. B. Kallestrup, L. H. Lyngø, R. Akkerman y T. A. Oddsottir, «Decision support in hierarchical planning systems: The case of procurement planning in oil refining industries,» *Decision Support Systems*, vol. 68, pp. 49-63, 2014.
- [23] B. Aslan, M. Stevenson y L. Hendry, «Enterprise Resource Planning systems: An assessment of applicability to Make-To-Order companies,» *Computers in Industry*, vol. 63, n° 7, pp. 692-705, 2012.
- [24] International Organization for Standardization, IEC 25010:2011 Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models, ISO/IEC, 2011, p. 34.
- [25] P. Kruchten, «The 4+1 View Model of Architecture,» *IEEE Software*, vol. 12, n° 6, pp. 42-50, 1995.
- [26] Object Management Group, OMG Unified Modeling Language TM (OMG UML), 2.5 ed., OMG, 2013.
- [27] F. Bachmann, D. Garlan, J. Ivers, R. Little, R. Nord y J. Stafford, «Chapter 9. Documenting Software Architectures,» de *Software Architecture in Practice*, 2nd ed., Boston, MA, Addison-Wesley, 2003.
- [28] P. Clements, F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, P. Merson, R. Nord y J. Stafford, «Chapter 9. Beyond the Basics,» de *Documenting Software Architectures. Views and Beyond.*, Second Edition ed., Addison-Wesley, 2010, pp. 217-260.
- [29] R. Kazman, M. Klein y P. Clements, «ATAM: Method for Architecture Evaluation,» Pittsburgh, USA, 2000.
- [30] M. Klein, «Chapter 11. The ATAM: A Comprehensive Method for Architecture Evaluation,» de *Software Architecture in Practice*, Second Edition ed., Pittsburg, Addison-Wesley, 2008.
- [31] S. P. Gayialis y I. P. Tatsiopoulos, «Design of an IT-driven decision support system for vehicle routing and scheduling,» *European Journal of Operational Research*, vol. 152, n° 2, pp. 382-398, 2004.
- [32] G. Henning, «Production Scheduling in the Process Industries: Current Trends, Emerging Challenges and Opportunities,» *Computer Aided Chemical Engineering*, vol. 27, pp. 23-28, 2009.
- [33] G. Figueira, P. Amorim, L. Guimarães, M. Amorim-Lopes, F. Neves-Moreira y B. Almada-Lobo, «A decision support system for the operational production planning and scheduling of an integrated pulp and paper mill,» *Computers & Chemical Engineering*, vol. 77, pp. 85-104, 2015.
- [34] H.-H. Hvolvy y K. Steger-Jensen, «Technical and industrial issues of Advanced Planning and Scheduling (APS) systems,» *Computers in Industry*, vol. 61, n° 9, pp. 845-851, 2010.
- [35] D. Falessi, M. Ali Babar, G. Cantone y P. Kruchten, «Applying empirical software engineering to software architecture: challenges and lessons learned,» *Empirical Software Engineering*, vol. 15, n° 3, pp. 250-276, 2010.
- [36] P. Avgeriou y U. Zdun, «Architectural patterns revisited: a pattern language,» de *10th European Conference on Pattern Languages of Program.*, Irsee, Germany, 2005.
- [37] L. Bass, P. Clements y R. Kazman, «Understanding Quality Attributes,» de *Software Architecture in Practice*, 3ra ed., USA, Addison-Wesley, 2013, pp. 63-78.
- [38] I. C. Lopes Silva, P. H. S. Brito, B. Neto, E. Costa y H. Rocha, «A Tool for Trade-off Resolution on Architecture-Centered Software Development,» de *Proceedings of SEKE 2014*, Skokie, IL, USA, 2014.
- [39] N. B. Harrison y P. Avgeriou, «Leveraging Architecture Patterns to Satisfy Quality Attributes,» de *Proceedings of the First European Conference, ECSA 2007*, Aranjuez, Spain, 2007.
- [40] L. Bass, P. Clements y R. Kazman, «Architectural Tactics and Patterns,» de *Software Architecture in Practice*, 3ra ed., USA, Addison-Wesley, 2013, pp. 203-249.
- [41] F. Bachmann, L. Bass y R. Nord, «Modifiability Tactics,» Pittsburgh, USA, 2007.