

# Capítulo 5

---

## Tecnologías

### 5.1. Ingeniería de Software

#### 5.1.1. Prospectiva en Ingeniería de Software<sup>45</sup>

El propósito de este documento es realizar un análisis prospectivo de la Ingeniería de Software como disciplina, intentando identificar aquellos aspectos que sean relevantes para el desarrollo de una industria del software y de las TIC en Argentina.

La introducción de este documento hace un análisis de la Ingeniería de Software en el marco más amplio de la industria del software y de las TIC. Primero, explica porqué la Ingeniería de Software es fundamental para el desarrollo de la misma y de las TIC. Es decir, argumenta los motivos por los que un plan de desarrollo nacional donde el software y las TIC ocupen un lugar importante debe estar acompañado de un plan para desarrollar las capacidades nacionales en el área de Ingeniería de Software.

Luego, se argumenta que invertir en el desarrollo de la Ingeniería de Software tendrá un impacto ostensiblemente menor en la medida en que no haya una industria del software y TIC local. Si este fuera el caso, entonces las oportunidades se limitarán al desarrollo de la ingeniería de software como sector vertical de negocios.

El resto del documento está organizado en las siguientes secciones:

- La primera parte presenta un conjunto de propuestas de medidas que creemos que pueden ayudar a promover de manera conjunta el desarrollo de las capacidades nacionales en Ingeniería de Software, la industria del software y de las TIC.
- La segunda parte realiza un análisis de áreas de Ingeniería de Software que se perfilan como oportunidades para la Investigación y Desarrollo.

El análisis de la segunda parte del documento está basado en varios supuestos. El primero es que efectivamente habrá una serie de políticas que apunten al desarrollo general del sector del software y TIC, y que el modelo que tomará el desarrollo de estas industrias en Argentina estará basado en la generación de valor

---

<sup>45</sup> Este documento surge a partir del trabajo, comentarios e interacciones realizadas en el contexto del Foro Prospectiva en Ingeniería de Software en el que participaron Jorge Boria, Victor Braberman, Marcelo Campo, Santiago Ceria, Pablo Michelis, Angel Perez Puletti, Sebastian Uchitel y Daniel Yankelevich

en vez de modelos masivos como los de las Software Factories fundamentadas en ventajas de tipo de cambio o diferencias salariales. El segundo supuesto es que el futuro de la Ingeniería de Software, tanto en Argentina como en el resto del mundo, será mayormente moldeado por fuerzas disruptivas en el sector de las TIC que requerirán que las organizaciones cambien la forma en la que construyen y utilizan software<sup>46</sup>. El tercer supuesto es que la Argentina puede y debe tomar un rol de mayor protagonismo en el área de las TIC buscando invertir en oportunidades y tendencias que se muestran prometedoras en la disciplina de Ingeniería de Software en el nivel mundial y/o que son prometedoras por la ventaja competitiva que la Argentina pudiera tener debido a características locales particulares.

La segunda parte del documento comienza con una discusión sobre las fuerzas disruptivas actuales y de corto plazo en el sector TIC. Luego se discuten algunas de las tendencias y áreas de oportunidad dentro de la Ingeniería de Software y su vinculación con las fuerzas disruptivas.

### **Ingeniería de Software en el Marco de la Industria de TIC**

182 Un trabajo de prospectiva a 20 años en el área de las TIC o más específicamente en software o incluso en Ingeniería de Software necesariamente conlleva una probabilidad de error grande. Al área de las TIC no le faltan predicciones fallidas. Sin embargo, la dificultad de realizar predicciones sobre aspectos tecnológicos en el área de las TIC no impide hacer un análisis sobre el rol fundamental que jugará la Ingeniería de Software en los próximos 20 años tanto en el nivel mundial como en la Argentina.

Una de las razones clave por la cual es difícil realizar una prospectiva en TIC es lo difícil que es predecir la factibilidad técnica y comercial de tecnología en gestación en los laboratorios de avanzada y la dificultad de cuantificar el impacto que tendrán las tecnologías emergentes en el mercado. El área de TIC seguramente sufrirá transformaciones impensadas en las próximos dos décadas, de la misma manera que parte del estado del área era difícil de prever hace 20 años.

Si pensamos al software como una tecnología en sí misma, queda identificada una tecnología que sin duda jugará un rol central en el área de las TIC en las próximas décadas. Los desarrollos tecnológicos actuales y de las próximas décadas serán productos de software, productos que contienen un componente software embebido esencial a su funcionamiento, o el software jugará un rol central en su producción.

Sin duda, un plan de desarrollo nacional con vistas a fortalecer la industria de TIC no podrá prosperar sin un fuerte énfasis en software que es una de las “enabling technologies” clave para desarrollos tecnológicos.

---

46 Barry Boehm. Software Process Disruptors, Opportunity Areas and Strategies. USC-CSE-2005-500, January 2005

Ahora, si pensamos al software como un producto que debe ser construido, ya sea de cero o ensamblado a partir de otros componentes de software, la apuesta, en el contexto de un plan de desarrollo de la industria de TIC, es un país con tecnología para producir software de calidad. Software que podrá terminar en aplicaciones variadas subidas a oportunidades de negocio y/o a la próxima ola de innovación de hardware y comunicaciones. La tecnología que permite la construcción de software de calidad forma parte de lo que se llama Ingeniería de Software.

Según la IEEE, la Ingeniería de Software es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del mismo. El problema central que ataca a la Ingeniería de Software es la complejidad con que actualmente se lo construye. Lidar con esta complejidad necesita de técnicas y métodos concretos que ayuden a producir software de calidad con tiempos y costos controlados.

Las técnicas y herramientas de Ingeniería de Software han permitido extender hoy el límite de lo posible. Actualmente se lo construye de varios órdenes de magnitud mayor que hace unas décadas, satisfaciendo requerimientos de calidad más complejos debido al creciente nivel de sofisticación de la tecnología, negocios y necesidades de la sociedad. Además, el costo y tiempo de construcción de algunas soluciones estándares para la industria era impensable hace algunas décadas. Un ejemplo paradigmático son los bloques de software disponibles para construir y poner en producción un local electrónico en la web en un unas pocas horas de trabajo.

Pero la Ingeniería de Software podría decirse que es víctima de su propio éxito. A mejores herramientas de construcción, mayor complejidad del mismo que se construye y más demandantes los requerimientos que debe cumplir. Es esperable que la tendencia de las últimas dos décadas en cuanto al ritmo de crecimiento de la complejidad del software que se construye y la dependencia mayor que las empresas, el estado y la sociedad en general tienen sobre los servicios provistos por software continuará en las próximas décadas. Con lo cual, se requerirá, al igual que hasta ahora, un continuo avance del estado de la práctica y del arte en la disciplina.

En definitiva, la investigación, transferencia, innovación y capacitación en el área de Ingeniería de Software es fundamental para poder acompañar las necesidades de producción, operación y mantenimiento en los próximos 20 años. Cómo será el área de Ingeniería de Software en dos décadas es, nuevamente, difícil de predecir. ¿Qué tipo de software se construirá? El tipo que se construye cambia notoriamente el foco de los requerimientos y por ende de las técnicas y métodos constructivos para lograrlos. Por ejemplo, los servicios accesibles por la web han generado una renovación importante en los elementos que el ingeniero de software tiene en su caja de herramientas. ¿Qué infraestructura de construcción de software estará disponible? De vez en cuando salen “bloques de construcción” nuevos (lenguajes de programación, frameworks, middlewares, y aplicaciones altamente configurables, etc.) que cambian significativamente la

forma en que se desarrollan sistemas intensivos en software. Algunos ejemplos de estos bloques son los lenguajes de programación orientados a objetos, CORBA, ERPs y servicios web.

A pesar de que predecir los cambios tecnológicos dentro del área del software es extremadamente difícil y susceptible de error, la Ingeniería de Software (o el conocimiento en Ingeniería de Software) sirven para razonar sobre estos cambios de tecnología. Es el conocimiento que debe haber en Argentina a través de los profesionales para entender si vale la pena adoptar el cambio, cómo adoptarlo, cómo aprovechar al máximo sus ventajas y cómo resolver sus deficiencias. Aquí hay un aporte importante para Argentina si es que va a tener una industria de TIC.

En resumen, creemos que las capacidades actuales y futuras en Ingeniería de Software en la Argentina serán definitorias en la capacidad de desarrollo en el área de TIC.

### **Ingeniería de Software como sector vertical**

La Ingeniería de Software puede pensarse como un área de negocios en sí misma, con capacidad de vender y exportar servicios y productos. Esta visión complementa a la de la Ingeniería de Software como sector horizontal, desarrollada anteriormente, que facilita el desarrollo de sectores más amplios como los del software o TIC, o el desarrollo de sectores aplicativos específicos como podría ser productos intensivos en software de procesamiento de imágenes para aplicaciones médicas.

Desde esta perspectiva hay dos posibilidades: la de desarrollar productos y servicios innovadores, probablemente apoyándose en oportunidades generadas por fuerzas disruptivas en el sector de las TIC, y la de desarrollar productos y servicios que, aunque no sean necesariamente innovadores, sean altamente competitivos debido a alguna ventaja generada por circunstancias locales particulares. En la próxima secciones se analizan algunas posibles fuerzas disruptivas en el sector TIC y oportunidades y tendencias en la Ingeniería de Software, ya sean mundiales o por circunstancias locales.

Un sector vertical fuerte en el área de Ingeniería de Software, más allá de su importancia como industria de gran valor agregado en sí misma, aportaría a fortalecer a la industria del software y de las TIC, tanto proveyendo servicios a estos sectores como generando junto a ellos una red de empresas que apunte a la calidad e innovación de procesos productivos. La contraparte de este argumento es que el desarrollo de la Ingeniería de Software local tendrá un impacto ostensiblemente menor en la medida en que no haya una industria del software y TIC local que se nutra de ella.

## 5.1.2. Recomendaciones

Las secciones anteriores argumentan que la suerte de la Ingeniería de Software, la industria del software y de las TIC en Argentina están entrelazadas. Por lo tanto, es importante considerar políticas públicas que intenten fomentar el desarrollo conjunto de las capacidades nacionales en estas tres áreas. Esta sección contiene algunas recomendaciones sobre aquellos temas que deberían ser atacados por políticas públicas.

Es importante notar que las recomendaciones tienen un grado de solapamiento y es posible que puedan combinarse algunas medidas para lograr objetivos comunes.

### Recomendaciones relacionadas con la educación

Creemos que gran parte de las medidas que se deben tomar están relacionadas con la educación. Las siguientes secciones detallan algunas de las recomendadas.

Es fundamental tener en cuenta que la implementación de cambios en el sector educativo necesita para la generación de contenidos y material de soporte una cantidad de tiempo sustancial de personal altamente calificado no sólo en los temas técnicos sino en las formas de transmitirlos. Estos recursos humanos ya están insertos en el ámbito educativo, científico y/o productivo. Un desafío será brindar suficientes incentivos para que aporten a la generación de cambios en el sector educativo. Los subsidios del Estado podrían ser, según su reglamentación, una vía para lograr esto.

185

### Educación Universitaria

- Elaboración de una propuesta de Planes de Estudio y preparación de Material “enlatado” para materias de Ingeniería de Software para carreras de computación. Se espera que éstas cubran una proporción importante del “Software Engineering Body of Knowledge”<sup>47</sup> y que tengan en cuenta los temas mencionados en la prospectiva (ver más abajo) como críticos.
- Propuesta de Planes de Estudio y preparación de Material “enlatado” para materias de Ingeniería de Software para carreras de Ingeniería o Sistemas. Se espera que estas materias estén orientadas a Ingeniería de Software “in the Large” y a temas de gerenciamiento. Ejemplos de materias son Ingeniería de Requerimientos, Arquitecturas y Diseño, Verificación y Validación, Gestión de Proyectos, Calidad y Mejora de Procesos.
- Propuesta de Planes de Estudio y preparación de Material “enlatado” para materias de Ingeniería de Software para carreras cortas del tipo “tecnicaturas”. Por ejemplo, una única materia cubriendo lo básico del ciclo de vida de desarrollo, apuntando primordialmente a Ingeniería de Software “in the small”.

47 Software Engineering Body of Knowledge. <http://www.swebok.org/>



- Implementación de “tracks” en Ingeniería de Software. La idea es que las carreras de Ingeniería en Informática, Computación y Sistemas tengan un “track” de especialización en el área, a través de un conjunto de materias obligatorias y optativas. La iniciativa consiste de proponer currículum estándar para estos “tracks”.

### **Educación de Posgrado – Professional Degrees**

- Implementación de “professional degrees” en Ingeniería de Software para ayudar a la reconversión de recursos actuales y formar una nueva capa de liderazgo para nuestra industria. Ejemplos de carreras de estas características son el MSE (Master of Software Engineering) de Carnegie Mellon y el MSc Ind. (Master in Software Engineering for Industry) de Imperial College de Londres.

### **Educación de Posgrado – Doctorados**

- Promoción de doctorados con orientación en Ingeniería de Software a través de becas competitivas. Debe tenerse en cuenta el diferencial de sueldo al que los doctorandos pueden acceder con un trabajo en industria. Si este diferencial es muy significativo como en la actualidad, difícilmente se pueda fortalecer este sector. Se buscaría dar orientación (no excluyente) a los temas mencionados en la prospectiva.

186

### **Escuelas de Verano en Ingeniería de Software**

- Implementar una conferencia anual con cursos de una semana, becando a alumnos de carreras de todo el país. Esto sería similar las Escuelas de Informática de la UBA (ECI) y de la Universidad de Río Cuarto, pero con un foco en Ingeniería de Software. Una alternativa es implementar un “track” de Ingeniería de Software en una iniciativa existente.
- Implementar una conferencia anual orientada a doctorados con cursos de una semana, becando a alumnos de carreras de todo el país. Esto sería similar a la exitosa experiencia de las escuelas PAV. Si existiera una iniciativa más amplia, implementar un “track” de Ingeniería de Software.

### **Investigación y Desarrollo**

- Fomentar la instalación de “Centros de Excelencia” especializados en alguna temática de Ingeniería de Software. Operados por una universidad y con “visiting scientists” de la industria o centros académicos del exterior. Con subsidios estatales y aportes de empresas privadas. El foco de estos centros debe ser llenar el espacio existente entre investigación, docencia e aplicación de temas de Ingeniería de Software.
- Fomentar el desarrollo de grupos de investigación mediante subsidios que brinden flexibilidad para desarrollar programas de transferencia, co-

laboración nacional e internacional.

- Fomentar las actividades de transferencia de tecnología en Universidades (esto se aplica tanto a profesores como a investigadores). Por ejemplo, aumentando el peso de las actividades de Transferencia de Tecnología en la evaluación de profesores y docentes auxiliares en concursos o en promociones del CONICET.
- Fomentar el ingreso al sistema científico tecnológico de recursos humanos con trayectoria en industria en roles orientados a investigación aplicada y transferencia tecnológica.
- Establecer un Instituto orientado a identificar oportunidades de usar software para mejorar nuestras industrias / empresas de servicios combinado con subsidios para esos proyectos. Un proceso posible sería el siguiente: un grupo de consultores especializados de este Instituto hace una visita a una empresa para conocer sus procesos. Se identifican oportunidades de mejorar la productividad / calidad a través del uso del software. Se buscan subsidios estatales para llevar a cabo los proyectos resultantes. Los consultores preparan los pliegos para estos proyectos. Los consultores asesoran a la empresa durante la ejecución de los proyectos.

## **Otras Iniciativas e Ideas Propuestas**

### **Rol del Estado como Formador de la Industria y Comprador de Tecnología**

- Trabajar con la ONTI (Oficina Nacional de Tecnologías de la Información, dependiente de la Subsecretaría de la Función Pública) en la inclusión de evaluación de capacidades en Ingeniería de Software de las empresas que sean contratadas por organismos del Estado para desarrollos de software. Esto puede ser implementado de varias formas, como por ejemplo el otorgamiento de puntos en licitaciones por certificaciones de calidad o la inclusión en forma obligatoria de capítulos sobre el tema en los Pedidos de Propuesta (RFP) de organismos del Estado. Por ejemplo, la ONTI podría proveer de RFPs “estándar” a los organismos estatales.
- Fomentar que el Estado Argentino compre software de Industria Nacional, y contrate empresas nacionales para sus proyectos de desarrollo. Evitar los desarrollos internos cuando sea posible. Esto puede implementarse otorgando un diferencial de puntos en licitaciones para empresas locales.

### **Subsidios a la “certificación de calidad”**

- Continuar en la línea actual de otorgar aportes no reembolsables a empresas que sigan modelos de calidad y puedan demostrar haber cumplido con un hito importante como por ejemplo una “certificación” o similar.

## Modificaciones a la Ley de Software

- Fomentar la colaboración e inversión directa en centros universitarios o de investigación científica por parte de las empresas beneficiadas por la ley del software.

## Incentivos a la compra de “Software Argentino”

- Fomentar que empresas argentinas compren “software argentino”. Este puede hacerse con subsidios o préstamos blandos, o campañas de promoción. Tal vez el INTI pueda ayudar en estas campañas.

## Incentivos a la Instalación de Centros de Investigación y Desarrollo

- Si bien esto idealmente debería darse tanto para empresas locales como extranjeras, la tendencia hacia la concentración en la Industria del Software hace que sea más realista pensar en acuerdos con los grandes “players” de la industria para que, además de instalar centros de desarrollo en Argentina (como ya ha ocurrido con Oracle por ejemplo), instalen centros de investigación, que podrían trabajar en forma conjunta con las Universidades (esto debería relacionarse con los “Centros de Excelencia” propuestos anteriormente).

## 188 Certificaciones Personales

- Además de las certificaciones de calidad para empresas, es importante no perder de vista que hay una tendencia hacia las certificaciones profesionales de las personas que trabajan en la industria. Creemos que es posible fomentar estas certificaciones a través de mecanismos como aportes no reembolsables del Fonsoft.
- 

### 5.1.3. Prospectiva I+D en Ingeniería de Software

#### Metodología

Nuestro estudio se basa en la identificación y el análisis de las “fuerzas disruptivas” que de alguna forma moldearán, en nuestra opinión, la forma en que se hace software. Las “fuerzas disruptivas” fueron identificadas a partir de reportes académicos<sup>46,47</sup> y de la industria, combinados con la experiencia personal de quienes estamos preparando el reporte. Un comentario importante es que no todas estas fuerzas son novedosas. En muchos casos, el fenómeno que vemos es sólo una profundización de algo que existe hace años o incluso décadas. A su

---

46 Matthias Hözl y Martin Wirsing. State of the Art for the Engineering of Software-Intensive systems. Ludwig-Maximilians-Universität. München, 2008

47 Barry Boehm. Software Process Disruptors, Opportunity Areas and Strategies. USC-CSE-2005-500, January 2005



vez, algunas de estas fuerzas son el resultado de avances en la propia tecnología, generando un círculo que se retroalimenta.

## Fuerzas Disruptivas

### Importancia creciente de Atributos de Calidad

Los atributos de calidad no relacionados con la funcionalidad “de negocio” de una aplicación tendrán una importancia creciente. Los aspectos funcionales de los sistemas continuarán pasando a segundo plano con respecto a aspectos “no funcionales” (también llamados atributos de calidad o “ilities”). Existen cuantiosos ejemplos en donde los primeros productos innovadores funcionalmente no son los que terminan siendo exitosos, sobrepasados por productos que entran al mercado posteriormente pero con un balance de cualidades no funcionales (confiabilidad, usabilidad, flexibilidad, interoperabilidad, etc.) más apropiadas. Algunos atributos de calidad que creemos tendrán una importancia particularmente relevante son:

- **Seguridad.** La tendencia a la digitalización total de información relevante de personas, empresas, gobierno y sociedad en general junto con la tendencia a que todos los dispositivos de cómputo tengan conectividad, está llevando a que casi todos los sistemas tengan acceso en forma directa o indirecta a información sensible. Por lo tanto, atributos de calidad asociados a la categoría de requerimientos denominados de seguridad serán de alta relevancia: Integridad, Privacidad y Confidencialidad. Es importante no tomar una visión simplificadora de los atributos de seguridad pensando sólo en temas específicos como autenticación, encriptación y controles de acceso. La problemática de seguridad va más allá de estas soluciones técnicas. Pensemos por ejemplo en el problema de robo de identidad logrando acceso a información sensible “engañando” al operador en un call-center, de las relaciones de confianza (y sus respectivos controles) necesarias para garantizar que la prepaga médica no comparta datos con la compañía de seguros de vida, o finalmente del extravío de un celular ejecutando un sistema de e-banking que incluye billetera electrónica y manejo de portafolio de inversiones. La tendencia mundial<sup>48</sup> es que los fraudes, ataques y violaciones a la información personal (por ejemplo, robo de identidad) crecen en complejidad y por lo tanto las técnicas destinadas a esta problemática también deben crecer en complejidad y efectividad.
- **Confiabilidad / Disponibilidad.** La informatización de procesos de negocio y de operatoria social en general llevará a una dependencia aún mayor en el software que hace que estas operatorias sean posibles. Por ello, el grado con el cual el software es capaz de realizar sus funciones por períodos ininterrumpidos, más allá de cambios o fallas en su ambiente de ejecución, es crítico. Existen muchos sistemas que,

---

48 Computer Emergency Response Team. <http://www.cert.org>

sin responder a la definición tradicional de sistemas críticos, no toleran interrupciones a la operación por motivos de mercado o sociales. Cada vez más la tolerancia a interrumpir el negocio por fallas en IT se va reduciendo.

- **Precisión.** Ligada a la tendencia a digitalización completa de datos e informatización de procesos está la necesidad de que los sistemas operen sobre datos que sean un reflejo preciso del mundo real que representan, entendiendo a la oportunidad como un aspecto clave de la precisión. La expectativa de los usuarios es, cada vez más, la de un mundo digital que refleje perfectamente su percepción del estado del mundo real. Por ejemplo, la expectativa de que un gasto realizado con tarjeta o un depósito por cajero automático sea reflejado automáticamente en el estado de la cuenta. Este nivel de precisión temporal no puede simplificarse a requerimientos de performance sino que también puede requerir reingenierías importantes de sistemas existentes: ej. eliminación gradual de procesos batch por procesamiento on-line y desplazamiento a eliminación de controles manuales previos al procesamiento de transacciones a controles automáticos o controles manuales post-hoc.
- **Usabilidad.** A medida que las interfaces de usuario se van perfeccionando y el software se va metiendo en temas previamente impensados, alcanzando a cada vez más personas sin acceso previo al “mundo digital”, se va dando una creciente presión sobre el diseño de esas interfaces para que el sistema sea exitoso. Por lo tanto, vamos a ver una presencia creciente de esta problemática en los aspectos de análisis y diseño de una aplicación. Las fallas en la interfaz van a tener costos cada vez mayores.

190

## **Movilidad y Conectividad Permanente**

La tendencia hacia los dispositivos móviles va a seguir creciendo en el futuro, junto con el requerimiento de una conexión permanente que implica la necesidad de entender el contexto en el que está el usuario de ese dispositivo y la “transparencia” en los cambios de conexiones. El uso de dispositivos móviles también tiene implicancias importantes sobre el diseño de interfaces de usuario.

## **Nuevas opciones para el negocio del Software**

Existen múltiples modos de contratar el uso de funcionalidad provista por software. Desde la contratación de la construcción de software donde la empresa contratante es dueña del software desarrollado, pasando por licencias del tipo open source y software libre, hasta la contratación del uso de software como si fuera un componente más en la cadena productiva. La evolución de estos mecanismos nos resulta difícil de pronosticar, pero creemos que los paradigmas que resulten imperantes impactarán sobre la forma que se construye software por la creciente posibilidad de “armar” a partir de otros elementos, en tiempo

de diseño o de ejecución. Más allá de este problema de “licenciamiento” de los sistemas, los cambios en el modelo de negocio (quién paga por el software / cómo se cobra por ese software) involucran otras variables que sin lugar a dudas impactan sobre la forma que tendrá la industria: la tendencia al “software as a service”, en muchos casos impulsada por los grandes proveedores, genera cambios importantes en la estructura del negocio del mismo, pudiendo impactar sobre el ecosistema de organizaciones de IT (roles de las empresas más pequeñas y barreras de entrada al negocio, entre otros factores a analizar).

### **Desafíos del Software como servicio**

Aunque la construcción de sistemas de software complejos mediante la composición de componentes no es novedosa, la tendencia a que estos componentes no estén bajo el control de la organización que construye o manda a construir el sistema sí representa una novedad. Estos componentes en efecto brindan servicios sobre los que se construye un sistema más complejo. La pérdida de control directo sobre los servicios significa exposición, sin preaviso, a pérdida de acceso al servicio, cambios en la calidad del servicio, cambios de versiones y por consiguiente potencial cambio de funcionalidad del servicio, entre otros. Sumado a esto, debido al dinamismo del sector y a que cada servicio es utilizado por múltiples sistemas (de organizaciones distintas), la velocidad con la que estos cambios no controlados ocurren será cada vez mayor.

### **Semántica del contenido**

A medida que todo el conocimiento va estando disponible en formato digital para cualquier persona que tenga acceso a la Web, crece la importancia de entender el significado de ese conocimiento para determinar si es relevante o no en un determinado contexto o para tomar decisiones en cuanto a su distribución (“esta información, dónde tiene que ir?”). Si bien el impacto sobre la ingeniería de software de esta tendencia no es claro, al menos sabemos que nuestros sistemas deberán estar preparados para discernir cada vez más sobre el significado de lo que se recibe y procesa. Esto a su vez puede tener implicancias sobre el proceso de desarrollo (cómo se identifican, analizan, diseñan, construyen y prueban estas nuevas funcionalidades que interpretan la información de manera automática, ya que la revisión manual será imposible).

### **Integración digital**

La tendencia mundial es a una mayor digitalización de los datos y de la información. Cada vez más datos son digitales. Si bien el crecimiento del porcentaje de información digital seguramente es asintótico (y nunca llegará al 100%), permite encarar soluciones, sistemas y procesos que hace pocos años hubiera carecido de sentido pensar. Un caso particular de esta tendencia a mayor digitalización de la información es la convergencia digital: La distinción entre elemento de cómputo para comunicación, entretenimiento y cómputo clásico se hace borrosa. La proliferación de dispositivos hace que portabilidad/familias de

productos/middlewares y arquitecturas sea de mayor criticidad. La problemática del Infotainment, en cuanto a su impacto en las tecnologías y la forma de hacer software, también se incluye en este punto.

### **Poder de cómputo y de almacenamiento ilimitado**

Algunas tendencias hacia la creciente digitalización están siendo posibilitadas por la disponibilidad prácticamente infinita de almacenamiento y de cómputo. Esto abre nuevas oportunidades para la explotación de información o para la ejecución de procesos altamente demandantes de recursos que antes no estaban disponibles. De la misma forma, aparecen proveedores que ofrecen ese almacenamiento o procesamiento a las organizaciones que usan o desarrollan software (por ejemplo Google para almacenamiento y Amazon para procesamiento). Esto puede tener efectos sobre la composición de sistemas y sobre el concepto de “COTS” (comercial off-the-shelf software), además de seguir expandiendo la frontera de dominios que pueden ser alcanzados por el software.

### **Velocidad de Cambio**

El software deberá responder cada vez más rápido a los cambios que provengan del mundo exterior, por la creciente competencia y porque los nuevos dominios en los que el software juegue un papel relevante no permitirán demoras para realizar esas actualizaciones. La creciente interoperabilidad con otros sistemas también hará que los sistemas que no se actualicen inmediatamente queden obsoletos y no puedan seguir formando parte de otros “sistemas de sistemas”. Los métodos de desarrollo que se usen, sobre todo los orientados a la evolución de aplicaciones existentes, deberán tener en cuenta estas crecientes presiones.

192

### **Desarrollo e Integración Global**

A medida que avanzan las posibilidades brindadas por las comunicaciones y la globalización, los procesos de desarrollo cada vez deberán tener más en cuenta la posibilidad de trabajo con equipos distribuidos, donde el reparto de responsabilidades puede ser por tipo de tarea, por alcance funcional o por otros aspectos que todavía no están claros. Esto tendrá también un impacto creciente sobre las herramientas que usemos para dar soporte al proceso de desarrollo. Relacionados con estos temas podemos también tener en cuenta a:

- La escasez de talentos. Desde hace varios años se viene insistiendo sobre la falta de recursos capacitados para todos desafíos que enfrenta nuestra industria. La globalización ha ayudado a cubrir en parte esta falencia, pero esto no parece ser algo que se pueda resolver en el corto plazo.
- La reducción de la Brecha Digital. Tal como se mencionó al hablar de su uso, el universo de usuarios de aplicaciones va a seguir creciendo, presentando nuevos desafíos.

## Oportunidades / Tendencias

### “Ility” Engineering

Aquí podemos tomar prestado un término de Boehm, “Software Quality Attribute Engineering” que puede ser definido como el subconjunto de la ingeniería de software que trata de cómo identificar y especificar atributos de calidad, cómo implementar tácticas que los resuelvan en el nivel de la arquitectura o diseño, y cómo validar y verificar su cumplimiento en una aplicación. Los atributos de calidad pueden ser muy diferentes entre sí (por ejemplo usabilidad vs. flexibilidad vs. performance). El tratamiento específico de cada atributo de calidad como una problemática diferente permitirá avanzar en el camino de la especialización, buscando métodos y técnicas específicas en vez de usar otras generales que no se apliquen o se apliquen pobremente al problema que se está estudiando. Así como ya se usan términos como “Performance engineering”, aparecerán las “engineering” aplicadas a otros atributos de calidad. Habrá consideraciones comunes que entrarán dentro de lo que podríamos llamar “software quality attribute engineering”.

### Líneas de Productos de Software

La tendencia, empujada por factores diversos como ser estrategia comercial, diversidad de dispositivos computacionales y el reuso de componentes de software en sistemas de software complejos, no es a crear un producto para un cliente, entorno y aplicación particular, sino una familia de productos que tienen un núcleo central común pero variaciones puntuales y controladas para adaptarse a, por ejemplo, las necesidades específicas de clientes (variaciones en normativas y leyes según el país, interfaces diversas a sistemas de software legacy del cliente, estrategias-tácticas-modos de operación específicas del cliente), hardware de cómputo específico (por ejemplo, el desarrollo de un juego en red para que ejecute sobre una PC, teléfonos celulares con todas sus variaciones de tamaño de pantalla, y PDAs con distintos estilos de input, complicado aún más por las diversas capacidades de conectividad que cada dispositivo tiene), sistemas compuestos con objetivos diversos (el software de seguimiento y control de navegación para un tractor arando un campo en Entre Ríos tiene similitudes y diferencias a la que usa una flota de buques pesqueros en el Atlántico).

La clave en esta visión de proceso de desarrollo (un producto vs múltiples productos) parte de los requerimientos y tiene fuerte impacto en todas las actividades de desarrollo. La clave es identificar y gestionar un núcleo central de artefactos relevantes para el desarrollo (particularmente en materia de requerimientos y diseño) y sus puntos de variabilidad. Esto implica técnicas, procesos y herramientas que van desde la gestión de requerimientos, pasando por control de versionado, verificación y validación, y a mecanismos novedosos de actualización de versiones en clientes.



## **“Productización” de aplicaciones**

Una problemática asociada al punto anterior es la siguiente: La dinámica comercial de empresas desarrolladoras de software y el grado de experiencia y conocimientos en el dominio de aplicación necesario para construir una familia de productos hace que las empresas que se lanzan a este accionar parten de la venta de uno o varios sistemas a medida, hechos de manera ad-hoc para clientes. El pasaje de los activos (código, casos de prueba, requerimientos, modelos, etc.) de una visión de sistema a una visión de producto es muy compleja (esto sin incluir el cambio cultural a producir en una organización acostumbrada al desarrollo de sistemas y no de productos). Aquí hay oportunidades para la aplicación de técnicas automatizadas y semi-automatizadas para el procesamiento de artefactos del proceso de desarrollo que permitan extraer, abstraer y refactorizar los artefactos disponibles y ayudar en la construcción (primero de la visión y después concretamente) de una familia de productos.

## **Nuevas tendencias en Ingeniería de Requerimientos**

Este tema es viejo, difícil y sigue siendo tan fundamental como siempre. Su aplicación deficiente es la causa principal para el fracaso/bajo-rendimiento de proyectos intensivos en software. La temática corta transversalmente a las problemáticas de familia de productos, productización y software quality attribute engineering mencionados anteriormente así como varios de los que se mencionan más adelante.

194

El tema también puede ser relevante dadas las circunstancias particulares de la Argentina y haciendo un paralelo con Irlanda: El problema de IR tiene un componente técnico muy complejo y otro humano/blando/no-técnico importante en donde la afinidad cultural juega un rol importante. Irlanda viene empujando este tema tratando de aprovechar su posición de cercanía cultural a Europa y EEUU para poder ser el nexo/puente entre éstos y los países que no tienen esa afinidad pero serán potencias de desarrollo de software por sus dimensiones (i.e. India y China). Este rol bien podría cumplirlo Argentina en el plano regional Iberoamericano.

## **Arquitecturas para Sistemas de Software Complejo – Computación autónoma**

Existen una serie de ejes sobre los cuales se espera que la complejidad de los sistemas de software crezca de manera sustancial. Un ejemplo viejo pero aún muy vigente es el de concurrencia y distribución (de cómputo y datos). Sin embargo, a partir de los disruptores enunciados anteriormente, nociones como movilidad, sensibilidad al contexto y autonomía (sistemas “self-\*”, aquellos con la habilidad de ensamblarse, configurarse y adaptarse a cambios del entorno por si solos, así como también sistemas con la posibilidad de responder a fallas propias de manera autónoma) desafían el estado de la práctica y el estado del arte de la ingeniería del software. Central a estos dominios es una ingeniería de software con capacidad de incluir en etapas más tempranas de concepción del

producto/servicio o infraestructura a la arquitectura de la solución, permitiendo construcción con componentes off the shelf (que simplifica la incorporación de tecnología novedosa/disruptiva de manera más simple), y la construcción/análisis o síntesis del glue-code, hasta la gestión (cada vez más compleja) de aspectos como despliegue y mantenimiento.

El desarrollo y utilización de plataformas de construcción de software (arquitecturas conceptuales y de referencia, middlewares, frameworks, etc.) que reduzcan la complejidad de construcción proveyendo abstracciones adecuadas será de suma importancia. La construcción de estas plataformas puede ser una oportunidad interesante porque la identificación de las abstracciones adecuadas y la provisión de los servicios necesarios son no triviales y puede ser la base de servicios y productos de alto valor agregado. El uso de estas plataformas requerirá cierto grado de sofisticación que involucrará un entendimiento importante de la problemática que apunta a desarrollar la plataforma y también de la tecnología en sí misma.

### **Automatización de la Ingeniería de Software**

La sofisticación de herramientas de análisis y transformación de artefactos del proceso de desarrollo de software está comenzando a tener un impacto, largamente prometido, en el tipo de herramientas (de software) que soportan las actividades de verificación, validación y construcción. Combinaciones novedosas de técnicas de análisis de código, model checking, demostración de teoremas, data mining, y síntesis están siendo utilizadas cada vez más para potenciar las relativamente simples técnicas utilizadas hoy por la industria. El impacto de algunas de estas técnicas ya puede verse en V&V como testing, en temas de diseño y programación como refactorización automática y detección de clones. El potencial que tienen estas técnicas en la automatización de actividades de IS tendrá un fuerte impacto en lidiar con la creciente complejidad de la disciplina. Por ejemplo, las problemáticas de líneas de productos y productización ciertamente ofrecen oportunidades para la aplicación de técnicas automatizadas y semi-automatizadas para el procesamiento de artefactos del proceso de desarrollo que permitan extraer, abstraer y refactorizar los artefactos disponibles y ayudar en la construcción (primero de la visión y después concretamente) de una familia de productos.

195

### **Desarrollo basado en Modelos**

Este es un problema clásico de la ingeniería de software: cómo usar modelos de mayor nivel de abstracción para poder generar de manera automática o semiautomática distintas aplicaciones, además de lograr un análisis más oportuno sobre propiedades de lo que se está construyendo y facilitar las tareas de verificación.

A lo largo de la historia de la ingeniería de software se fueron logrando avances en estos puntos, pero en muchos casos los nuevos paradigmas que fueron apareciendo (de división de un sistema en módulos, de centralización o descentra-

lización) causaron retrocesos importantes que provocaron que los avances en productividad, a partir de la automatización, nunca se logaran. Creemos que a medida que se va logrando cierta madurez en la disciplina y cierta estabilidad en estos paradigmas, esta tendencia inevitablemente irá ganando lugar y cambiando los procesos de desarrollo. A esta tendencia se deberá agregar la complejidad de la necesidad de nuevos modelos para los nuevos desafíos presentados en puntos anteriores, sobre todo los relacionados a atributos de calidad. Sin más y mejores modelos, esos avances no se podrán lograr.

### **Desarrollo distribuido y métodos ágiles**

Los modelos de desarrollo global seguirán evolucionando e impactando en el proceso de desarrollo. Por ejemplo, la aparición hace varios años de las llamadas “metodologías ágiles”, que ponen mucho foco en la interacción entre los integrantes de un equipo, tienen un impacto sobre la posibilidad de aplicarlas en lugares con poca compatibilidad horaria. Esto implica una oportunidad para la Argentina, teniendo en cuenta la zona compatible con Europa y Estados Unidos. Por lo tanto, se deberá invertir en especializar procesos de desarrollo que exploten estos beneficios que no pueden ser provistos por zonas más alejadas. También esto tendrá impacto sobre las herramientas que dan soporte a esa interacción más fluida. Este tipo de metodologías seguirán teniendo un alto impacto en los procesos de desarrollo ya que por sus características se adaptan más fácilmente a los entornos dinámicos que se describen en muchos de los puntos anteriores.

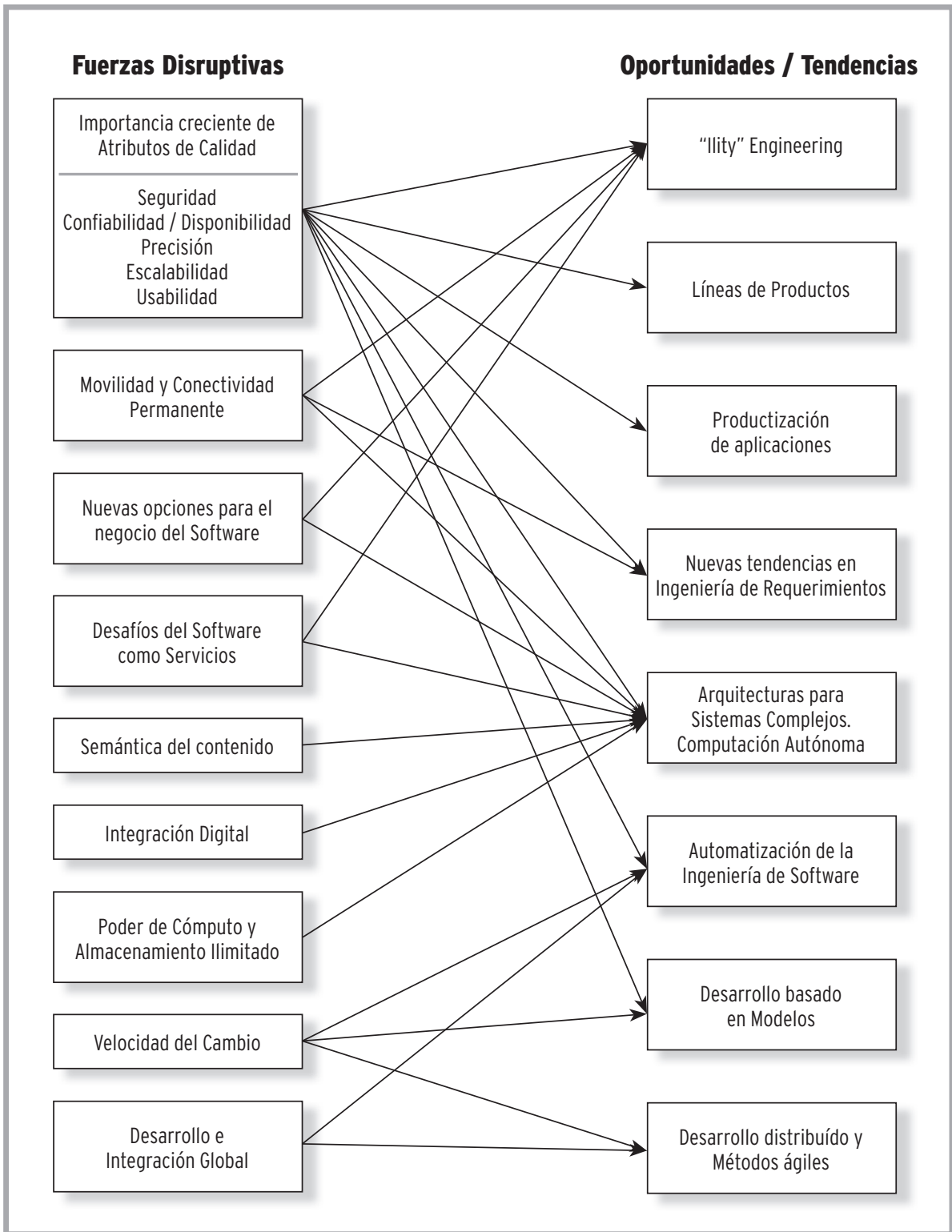
196

### **Temas no detallados**

La lista de temas detallados previamente no pretende ser una exploración exhaustiva de oportunidades y tendencias en Ingeniería de Software. Algunos temas no mencionados que habría que analizar con más detenimiento son “trust”, aplicaciones de data mining a ingeniería de software y temas de gerenciamiento.

### **Relación entre Fuerzas Disruptivas y Tendencias**

El gráfico de la página siguiente muestra algunas relaciones claras entre estos dos puntos.



## 5.2. Señales<sup>49</sup>

### 5.2.1. Áreas tecnológicas de interés

Telecomunicaciones:

- Acceso de banda ancha fijo y móvil y celular (DSL, cable, Wifi, Wimax, BPL, etc) (Wifi, Wimax, etc) (GSM de largo alcance, B3G, LTE, etc)
- Redes autónomas de sensores (manejo de tráfico, optimización de algoritmos de procesamiento en función del tráfico permitido, del consumo de energía, etc)
- Ecuación, rechazo de interferencias, sistemas de transmisión multiusuarios, detección y multi-muestreo
- Enlaces ópticos y optoelectrónica
- Enlaces satelitales – GPS
- Radar – sonar

Otras:

- Procesamiento (compresión) de voz e imagen
- Cancelación de eco acústico
- Bioingeniería (desarrollo de sensores y procesamiento)
- Spectrum Sharing – Cognitive radios – software defined radio

---

<sup>49</sup> Dr. Carlos Muravchik, Profesor Titular, Facultad de Ingeniería, Universidad Nacional de La Plata, Investigador Superior CIC.

Laboratorio asociado: Laboratorio de Electrónica Industrial, Control e Instrumentación (LEICI), Facultad de Ingeniería, Universidad Nacional de La Plata. Capacidades: Procesamiento estadístico de señales y arreglos de sensores. Aplicación en comunicaciones, GPS, radar, biomedicina y control de procesos. Implementación en RF y digital.

Dr. Rubén Milocco, Profesor Asociado, Facultad de Ingeniería, Universidad Nacional del Comahue, Investigador Independiente CONICET

Laboratorio Asociado: Grupo de Control Automático y sistemas. Capacidades: Filtrado estocástico y detección en sistemas de comunicaciones digitales.

Dr. Juan Cousseau, Profesor Titular, Departamento de Ingeniería Eléctrica y Computadoras, Universidad Nacional del Sur, Investigador Independiente CONICET.

Laboratorio asociado: Laboratorio de Procesamiento de Señales y Comunicaciones, Instituto de Investigaciones en Ing. Eléctrica (CONICET). Capacidades: Diseño en RF, DSP y FPGA

Dr. Mario Hueda, Profesor Adjunto, Facultad de Ciencias Exactas, Físicas y Naturales, Universidad Nacional de Córdoba. Investigador Asistente CONICET.

Laboratorio asociado: Laboratorio de Comunicaciones Digitales (LCD). Capacidades: Procesamiento de señales para comunicaciones. Diseño e implementación de sistemas de comunicaciones digitales de alta velocidad (DSP, FPGA)

Dra. Cecilia Galarza, Profesor Adjunto, Facultad de Ingeniería, Universidad de Buenos Aires. Investigador Adjunto CONICET.

Laboratorio asociado: Grupo de Procesamiento de Señales, Departamento de Electrónica Capacidades: Diseño de algoritmos para comunicaciones multiusuario

Dr. Emmanuel Jaffrot, Profesor, Escuela de Postgrado, Universidad Nacional de San Martín.

Laboratorio asociado:

Capacidades: Comunicaciones digitales, procesamiento estadístico de la señal, modelación del canal de propagación, sistemas inalámbricos y de banda ancha.



### Objetivos: (asociados a las áreas tecnológicas de interés):

En la última década del siglo XX, las telecomunicaciones protagonizaron un gran avance a nivel mundial, haciendo real el concepto de “mundo globalizado” que conocemos hoy. El avance tecnológico a nivel mundial se realizó en base al esfuerzo cooperativo, o sinergia, entre investigadores tecnológicos, desarrolladores de productos, y usuarios. Por diversas razones históricas, en nuestro país no se incentivó este proceso de innovación y desarrollo tecnológico. Como resultado, en la actualidad no se cuenta con planes de fortalecimiento del área en ningún concepto, educación, entrenamiento, industria.

En la Argentina actual, el ámbito industrial concentra sus actividades alrededor del suministro de equipamientos importados y del mantenimiento de los mismos. Existe poca transferencia tecnológica para la creación de soluciones originales e innovadoras. Estructurar la investigación académica en el área de las telecomunicaciones representa el primer paso para construir la innovación industrial en este mismo ámbito.

Una lista de objetivos urgentes a un plazo mediano son las siguientes:

- Desarrollar una fuerza de trabajo educada y competente (Educación, capacitación, entrenamiento, acceso a la tecnología)
- Dominar el proceso de innovación (Investigación básica y producción de PhDs, investigación aplicada, desarrollo de polos y clusters tecnológicos)
- Desarrollar nuevos servicios y productos para el mercado global
- Generar una masa crítica de profesionales capaces de fomentar el desarrollo de la industria local de las telecomunicaciones a través de sus desarrollos innovadores
- Desarrollar el acceso a conectividad (telefonía e infraestructura)
- Participación abierta e institucional en los procesos de estandarización de servicios.
- Estimular, facilitar y gestionar los medios necesarios para el intercambio con grupos de trabajo a nivel internacional.
- Desarrollar una política “agresiva” de publicación académica al nivel internacional
- Incentivar la producción de patentes nacionales e internacionales.

### **5.2.2. Procesamiento de Señales para sistemas de comunicaciones futuros**

#### **Síntesis**

A continuación se discutirán varios aspectos relacionados con la enorme actividad prevista, fundamentalmente para la próxima década, en tecnologías de comunicaciones móviles inalámbricas, por un lado y sistemas de comunicacio-