

# Gestión y Desarrollo de Proyectos de Software: Metodología Ágil basada en Telecomunicaciones MATE

Marisa Cecilia Tumino<sup>1</sup>, Juan Manuel Bournissen<sup>1</sup>, Claudio Bracalenti<sup>2</sup>, Eric Schlemper<sup>3</sup>, Silvio Kucharski<sup>1</sup>

1. Instituto de Ingeniería del Software. Universidad Adventista del Plata. Libertador San Martín, Entre Ríos. Argentina

2. Universidad Tecnológica Nacional Facultad Regional Santa Fe, Argentina

3. Sanatorio Adventista del Plata. Libertador San Martín, Entre Ríos Argentina

I3@uap.edu.ar

**Abstract**—La Metodología Ágil basada en Telecomunicaciones (MATE) es una propuesta metodológica orientada al progreso de emprendimientos informáticos que permite el desarrollo de software por parte de grupos pequeños con escasos recursos. La metodología contempla el trabajo semi-sincrónico, utilizando los adelantos, y las reducciones de costos asociadas en materia de comunicaciones y de recursos tecnológicos, contribuyendo a la fluidez y a la simplicidad del trabajo en un ambiente totalmente virtualizado y libre de gran parte de los costos fijos comunes a una Software factory convencional. Se han rescatado aspectos exitosos de otras metodologías tales como la reunión diaria de Scrum y la programación en parejas de XP, aunque las mencionadas reuniones se desarrollen en salas virtuales y las parejas de programadores residan en lugares distantes.

**Palabras claves** —Metodología ágil, Teletrabajo, desarrollo semi-sincrónico, Trabajo Freelance.

## I. INTRODUCCIÓN

La filosofía de las metodologías ágiles otorgan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas, mostrando su efectividad en proyectos con requisitos cambiantes o cuando se exige reducir los tiempos de desarrollo, manteniendo la calidad [1].

El presente trabajo tiene como objetivo primario proponer una metodología ágil que facilite el desarrollo de aplicaciones para dispositivos móviles, utilizando, como plataforma de desarrollo, medios que permitan el trabajo a distancia y puedan incluso soportar desplazamientos de los involucrados. El objetivo secundario es el desarrollo de un producto que facilite la gestión de los proyectos, utilizando la metodología propuesta.

## II. DEMANDAS DEL EMPRENDIMIENTO

En el presente, aun más que en el pasado, fundar una empresa puede llegar a transformarse en una actividad que requiera gran inversión de capital. La fiebre de los móviles con pantalla táctil [2] ha revolucionado el mercado de todo buen negocio de las aplicaciones móviles. Y es que estos terminales se han convertido en el ordenador del futuro con el valor añadido de disponibilidad total y de la comodidad y portabilidad.

Los emprendedores recién egresados de la universidad enfrentan dos grandes problemas: (a) las distancias que median entre las residencias de los miembros de equipo de trabajo y (b)

la demanda de capital para fundar una empresa de sede tangible.

Las metodologías más usuales, y desde donde se han adoptado los principios básicos del desarrollo de aplicaciones para dispositivos móviles, tales como SCRUM y eXtreme Programming (XP), sufren limitaciones condicionadas a las localizaciones físicas de los miembros del stakeholders, considerados éstos como quienes están afectados por las actividades de una empresa. Por su parte, para efectuar un desarrollo de software efectivo se requiere un ambiente acondicionado con los equipos tecnológicos y los medios de telecomunicaciones apropiados, por lo que se debe invertir además en los impuestos y servicios que demanda este tipo de emprendimiento.

Sintetizando los recursos necesarios para la puesta en marcha de un emprendimiento de desarrollo de software son los siguientes: (a) recursos humanos, (b) ámbito de trabajo, (c) hardware de desarrollo y comunicaciones, (d) software, (e) capital, (f) tiempo y (g) una metodología apta para el trabajo distribuido.

De la lista precedente resulta evidente que en principio el único recurso abundante con el que puede llegar a contar un recién graduado es el tiempo.

En virtud de las condiciones demandadas por las nuevas modalidades de desarrollo Freelance, donde los miembros de los equipos se encuentran ubicados en diferentes puntos geográficos, se considera necesario elaborar una propuesta idónea que dé respuestas a estas limitaciones y a las expectativas del campo de desarrollo de sistemas. Para ello surgen replanteos para cada uno de los recursos enumerados.

### A. Recursos humanos

La mecánica de trabajo se basa en la existencia de un grupo inicial de pares, es decir de profesionales que en un sentido técnico gozan de similares conocimientos y entre quienes no se encuentran razones para proponer un orden jerárquico. La organización de los involucrados debe darse en forma espontánea, asumiendo cada uno un rol consensuado y basado en la confianza recíproca. Los roles a desempeñar no implican una escala de poder sino, y esto es de capital importancia, una forma de obtener lo mejor de cada uno de los integrantes. Conforme prospere el emprendimiento es lógico que incorpore nuevo personal, pudiéndose establecer en este caso una jerarquía o continuar con el esquema organizativo inicial. En todo emprendimiento debe haber gestión, producción y

servicios de apoyo. La idea es dividir las necesidades en roles y asignar en forma consensuada uno o varios de ellos a cada miembro del equipo.

### B. *Ámbito de trabajo*

Disponer de un lugar común puede resultar más difícil de lo que parece por dos motivos:

- Alquilar o comprar un inmueble comercial requiere comúnmente una gran inversión de capital.
- Coincidir físicamente en una ciudad implica para algunos mudarse, con los consecuentes costos. La idea es entonces virtualizar el ámbito de trabajo, permitiendo a los emprendedores ejecutar sus tareas desde lugares remotos y compartir un universo virtual en el que se desarrollen sus labores.

### C. *Hardware de desarrollo y comunicaciones*

Lejos están los días en que un equipo de computación era asequible sólo para unos pocos. Hoy, por el contrario, en la mayoría de los hogares de alumnos o graduados universitarios las PCs, notebooks, módems y routers simplemente son parte del equipamiento estándar. La ventaja de trabajar en un ambiente distribuido y virtualizado es, justamente, la de poder contar con esta infraestructura preexistente. Sin embargo para esta propuesta también es imprescindible contar con un smartphone o una tableta que corra Android para el correcto testeo de los productos desarrollados, más allá de los emuladores que se encuentran disponibles en otras plataformas como Windows, Mac o las distintas distribuciones Linux. La elección de Android se sustenta en que es la plataforma móvil más extendida del mercado. Muchas de las posibilidades las brinda Android [3].

### D. *Software*

El interés por la calidad en el desarrollo de software crece de forma continua a medida que los clientes se vuelven más selectivos y comienzan a rechazar productos poco fiables o que realmente no dan respuesta a sus necesidades [4]. Si bien la metodología propuesta podría utilizarse en el desarrollo de cualquier tipo de software, y para cualquier plataforma, este trabajo se ha circunscripto inicialmente a la producción para dispositivos móviles y específicamente para aquellos que soportan Android. Se ha proyectado la propuesta sobre la premisa de trabajar preferentemente con productos gratuitos, atendiendo a las razonables necesidades de los destinatarios de esta metodología. En muchos casos implica la aceptación de publicidad pero que representa un costo razonable de la gratuidad.

### E. *Capital*

Las necesidades de capital inicial son mínimas. Las erogaciones necesarias no dejan de ser las mismas que las cotidianas, más allá de la existencia de un emprendimiento productivo. Como ejemplo vale decir que el grupo de investigación pudo trabajar con equipamiento propio al que solo se le debió sumar auriculares con micrófonos y una tableta genérica de muy bajo costo.

### F. *Tiempo*

El tiempo es un recurso que se asume como abundante, considerando que los involucrados están convencidos de las bondades de trabajar para sí mismos. No obstante su disponibilidad, este recurso debe manejarse con criterio pues la

ausencia de una cabeza visible que imponga metas puede conducir a una relajación que conspira contra el alcance del objetivo. Deben fijarse políticas de uso del tiempo, tanto del compartido como del individual, y pautas para verificar el cumplimiento de dichas políticas.

## III. LA METODOLOGÍA MATE

En virtud de las condiciones demandadas por las nuevas modalidades de desarrollo Freelance, donde los miembros de los equipos se encuentran ubicados en diferentes puntos geográficos, es necesario elaborar una propuesta idónea que dé respuestas a estas demandas y se adapte a los potenciales cambios de requerimientos [5].

El diseño de la MATE se alinea a las orientaciones de Alistair Cockburn [6] que desglosa una metodología en elementos. Se describen por lo tanto las funciones y los roles que ella requiere. Con este propósito se respetan las siguientes premisas:

- El ciclo productivo es corto, entregando un producto acotado aunque totalmente funcional cada 30 a 60 días. Este producto representa una Entrega de Valor Agregado (EVA) y está compuesta por al menos una historia del usuario, asumida esta última como la técnica utilizada para especificar los requisitos del software [7].
- Cada versión del producto se entrega debidamente testeada, por lo que cada ciclo productivo contempla el tiempo necesario para todos los tipos de pruebas, incluyendo las betas, es decir aquellas que involucran al cliente sin presencia del equipo de desarrollo.
- El trabajo se realiza a distancia en modalidad semi-sincrónica, con al menos 4 horas diarias de trabajo de programación conjunta, y el resto dedicado al testeo individual y a las pruebas de integración. Cada integrante del grupo de trabajo dispone del equipamiento necesario, tanto para el desarrollo de software como para las comunicaciones.
- El mejor lugar donde puede estar el cliente (de existir) es en su propio ámbito de trabajo. Eventualmente, y de ser posible, un miembro del equipo acude al domicilio del cliente para dialogar sobre los temas más sensibles.

### A. *Funciones y roles*

En la metodología se reconocen los siguientes roles como indispensables para la ejecución de un proyecto de estas características, conformando todos ellos el “stakeholders” en el sentido aceptado del vocablo en el campo de los sistemas (aunque no todos los “stakeholders” conforman el equipo):

1). *El dueño de la idea:* En sí mismo constituye una interfaz que vincula al “cliente” con el “equipo”. El cliente puede ser real y demandante (es decir que puede existir y desear una solución informática ajustada a determinadas funcionalidades y estándares de calidad) o, en el caso de concebir una aplicación sin cliente, es aquél que efectivamente ha tenido la idea y es capaz de cumplir con el rol de interfaz como si el cliente realmente existiera.

2). *El facilitador:* Es el líder de equipo. Comienza y termina las reuniones virtuales, determina el orden del día, enuncia los puntos acordados, fija posiciones con el dueño de la idea, gestiona los recursos y acuerda la extensión de cada EVA en función de lo propuesto por los desarrolladores.

3). *El oráculo*: Este actor gestiona toda la información concerniente al proyecto tal que cualquier integrante tenga un rápido acceso a la misma desde cualquier punto. Transforma el orden del día en una lista de eventos y genera eventos a medida que el orden del día es discutido.

4). *Los desarrolladores*: Los desarrolladores determinan la extensión de las tareas. Son quienes tienen a su cargo el desarrollo del código y su correspondiente testeo y depuración. Trabajan de a pares, compartiendo un escritorio virtual común, al menos cuatro horas diarias. El resto del tiempo testean y depuran el código.

5). *El testeador*: Este actor es el encargado de probar la integración de los componentes. Su función comienza cuando los desarrolladores concluyen las pruebas de unidad. Interactúa con el cliente durante las pruebas alfa y recibe la realimentación de las pruebas beta.

6). *El cliente*: El cliente es la base de la actividad comercial. Puede ser un comitente de existencia real, física, o jurídica, que encomienda el trabajo. En caso de que el equipo perciba una necesidad de mercado, éste pasa a asumir el rol del dueño de la idea.

Un integrante puede ejecutar más de un rol simultáneamente, desde su propio ámbito, incluyendo al cliente (de existir). Es probable que el dueño de la idea programe o que un programador utilice parte de su tiempo en oficiar de oráculo o bien que el testeador oficie de facilitador; dependiendo de las capacidades destacadas de cada miembro.

## B. Herramientas

Existen en el mercado variados productos que podrían brindar un soporte completo a la metodología o hacerlo parcialmente, de disponer de algunas mejoras. La lista de aplicaciones que permiten el teletrabajo es extensa en términos de video chat o video conferencia, captura de escritorios remotos para el desarrollo de a pares y testeo de la aplicación, servidores colectivos de datos para la gestión de la documentación y convertidores de voz a texto para el registro de notas de las reuniones. Asimismo para facilitar la comunicación, la modalidad de interacción puede ser de a pares o de a grupos numerosos.

1). *Evaluación comparativa de herramientas para trabajo de a pares*: Previo a la definición del nuevo planteamiento se han efectuado pruebas sobre un conjunto de productos, tanto pagos como gratuitos, que ofrecen una potencial solución al problema que plantea el trabajo de abordaje distribuido. Para compartir las aplicaciones y el escritorio, y vincularse mediante video conferencia, se probaron las siguientes herramientas: Adobe Connect, TeamViewer, HangOut y Saros, tanto en pruebas individuales como combinadas, salvo con Adobe Connect.

Luego de las pruebas generales y la consecuente selección, se realizaron pruebas con TeamViewer y HangOut para luego evaluar el desempeño conjunto de HangOut y Saros, ambas herramientas corriendo al mismo tiempo en dos computadoras distintas. Una de ellas fue una máquina de escritorio, con altos recursos de hardware, y la otra fue una netbook con limitaciones en tanto a sus capacidades gráficas como de procesamiento.

Se procedió a probar la comunicación por medio de voz sobre ip en las circunstancias mencionadas. Como fruto de las experiencias puede destacarse que resulta más recomendable la utilización de Saros, para compartir la pantalla con el código de programación, y HangOut, que tiene la ventaja de indicar si las

personas involucradas se encuentran conectados a Google+, Google talk o Gmail, para compartir la imagen y la voz. Por su parte se ha prescindido del Adobe Connect por no reunir los requerimientos mínimos que exige esta modalidad de trabajo y ser además relativamente costoso frente a las premisas descriptas con anterioridad. Bajo estas condiciones la programación en parejas opera con eficiencia, dado que la tecnología adoptada permite una comunicación fluida.

2). *Evaluación comparativa de herramientas para el registro de las minutas*: A los efectos de facilitar el trabajo distribuido, durante las reuniones periódicas, se pretende llevar registro de los diálogos mantenidos, en forma parcial o total, entre los stakeholders (integrantes del proyecto). La idea es adaptar un sistema automático de minutas para llevar el registro del diálogo mantenido en las reuniones. Se probaron para este fin, tres sistemas de captura de voz y su conversión a mp3: (a) Camtasia, (b) Voice Recorder y (c) Audacity, todos ellos igualmente recomendables.

3). *Evaluación comparativa de herramientas para la conversión de las minutas a texto*: La conversión se puede facilitar con Talktyper ([www.taltpyper.com](http://www.taltpyper.com)) aunque no transcribe archivos mp3. Esta aplicación web es gratis y dispone de un micrófono que recoge la voz y la transcribe a texto. Para las transcripciones reconoce grabaciones hasta en 18 idiomas, incluyendo el español, el inglés y el francés. Otra herramienta muy útil es Dragon Dictation, una aplicación para móviles, compatible con dispositivos Apple y Android. En la actualidad se está trabajando con Notepad Lite, como conversor de voz a texto, especialmente diseñada para Android con las funcionalidades necesarias para poder trabajar con archivos de texto.

Como producto de las pruebas se adoptaron las herramientas que dieron respuesta a la funcionalidad y rendimiento de la MATe.

## IV. SÍNTESIS DE LA METODOLOGÍA

Tal como se ilustra en la Figura 1, y se describe a continuación, el proceso de cada EVA propone las siguientes fases:



Figura 1: Fases de cada EVA

### A. Ciclo analítico

Dada una idea, propia o de un cliente, se constituye un equipo de análisis formado por el cliente y los desarrolladores, conducido por el dueño de la idea. El Ciclo analítico representa la reunión donde se trata la formulación de la idea global, el listado de las funcionalidades esperadas y la generación de las historias de usuario que se describen en pocas frases y en lenguaje coloquial. Con estas historias se elabora un documento de precedencia de funciones donde se descomponen éstas en tareas y se fijan las prioridades.

### B. Acuerdo VIP (Valor Individual Ponderado de las historias de cada EVA)

Este acuerdo se realiza una única vez por cada EVA, donde se define el “qué” se va a realizar. Se lleva a cabo en forma

previa a cada EVA e intervienen todos los involucrados, incluyendo al cliente de existir. La idea es capturar el espíritu de la aplicación que guiará el proceso de desarrollo, entendiendo “el espíritu de la aplicación” como el conjunto de anhelos que la justifica. En esta reunión el facilitador, en un todo de acuerdo con el dueño de la idea y con los desarrolladores, fijará el tiempo y el alcance de cada EVA. Una vez fijados estos parámetros, cada historia de usuario de la EVA es ponderada, por cada miembro productivo del equipo, a partir de la funcionalidad de la aplicación electrónica para la Metodología Ágil basada en Telecomunicaciones (e-MATE) que trabaja con distintos algoritmos. Esta técnica de planificación es una herramienta útil en la estimación del esfuerzo y de la complejidad de cada historia de usuario y que a su vez permite la estimación de los costos.

### C. Ronda grande

En esta instancia se define el “cómo” se van a realizar las tareas, para lo cual cada par se auto-assigna sus tareas enmarcadas en la EVA. Una vez asignada una tarea no puede ser asignada a otro par, excepto que el par que la posee renuncie a la misma. Intervienen todos los miembros exceptuando al cliente.

### D. Ronda media (semanal o quincenal)

Las Rondas medias se planifican los primeros días hábiles de cada semana e intervienen todos los involucrados, menos el cliente, dirigidos por el facilitador. Se revisa la marcha de las tareas y se ajustan a la realidad imperante.

### E. Ronda diaria

Participa un par productivo y el facilitador en algún momento del día y debe ser muy breve. Se reportan avances y dificultades y se revisan las tareas por hacer hasta la siguiente ronda diaria. El facilitador replica esta reunión con cada par productivo.

### F. Evaluación EVA

Concluido cada EVA se lleva a cabo una reunión que consta de:

1). *Evaluación del producto*: se reúnen todos los miembros del equipo de desarrollo con el cliente, o dueño de la idea, con el propósito de evaluar el EVA resultante.

2). *Evaluación del proceso*: se reúnen todos los miembros del equipo de desarrollo, exceptuando al cliente o dueño de la idea, con el propósito de evaluar el proceso, incluyendo las relaciones entre personas y las formas en que se podría mejorar en el próximo EVA. En esta oportunidad se registra en la e-MATE los tiempos reales que haya consumido cada tarea de cada miembro de equipo a los fines de generar la información referida al rendimiento de cada desarrollador.

La jornada de trabajo no debe superar las 8 hs diarias. Los pares productivos deben poder trabajar al menos 4 hs. conjuntas, por lo que sus locaciones geográficas y costumbres deben permitirlo, donde se trabaja en forma colaborativa y simultánea por captura de escritorio; mientras que el tiempo individual restante se dedica a la prueba y depuración.

## V. FUNCIONALIDADES DE LA APLICACIÓN E-MATE

En esta etapa de la investigación se ha avanzado en el desarrollo de una aplicación que asiste al equipo de análisis en la implementación de la metodología propuesta. Durante este proceso de construcción se probó con rigor la MATE. Para este

propósito el equipo estuvo conformado por desarrolladores radicados en diferentes países, con el mismo huso horario, confirmando el potencial de esta metodología para el trabajo distribuido. En este apartado se describen en forma sintética las funcionalidades de la aplicación.

El sistema e-MATE es una herramienta que ayuda a la organización y planificación de proyectos de desarrollo de sistemas, tanto para el uso de la metodología MATE como Scrum. El sistema brinda las herramientas para la administración de proyectos, administración de usuarios de sistema y las tareas propias de la metodología tales como historias de usuarios, la gestión de los EVA, la generación de estadísticas útiles para el cálculo de fechas de entregas, estimadas al finalizar los EVA, y la gestión del desarrollo.

e-MATE corre sobre plataforma móvil Android y web para permitir su fácil acceso y utilización. Entre las funcionalidades de e-MATE se encuentran:

### A. Gestión de proyectos

e-MATE permite la creación de múltiples proyectos, registrando sus fechas de inicio y de culminación, las horas previstas y la cantidad de horas reales de desarrollo, entre otras variables, facilitando la administración de los roles de usuarios en cada proyecto.

### B. Administración de usuarios de sistema

El alta de usuarios se da de manera general. Los roles son asignados individualmente en cada proyecto. Los usuarios pueden tener más de uno de los siguientes roles: (a) Dueño de la Idea, (b) Facilitador, (c) Oráculo, (d) Desarrollador, (e) Testeador y/o (f) Cliente.

### C. Gestión de historias de usuarios

La aplicación registra las historias de usuarios, ordenándolas en una lista priorizada llamada Realizables. El dueño de la idea es el responsable de priorizarlas. La estimación de la complejidad, del grado de dificultad y de los costos de una historia se define mediante la propuesta del VIP (Valor Individual Promediado), donde los miembros del equipo debaten los resultados por medio de la comunicación propuesta para la MATE.

### D. Gestión de EVA

Las EVA son creadas por el facilitador, tras una reunión de planificación. El facilitador deberá definir la duración y el alcance de la siguiente EVA. Para ello, la aplicación le sugiere la cantidad de historias de usuarios que podrían completar por medio del cálculo de la velocidad asumida del equipo y el ingreso de días laborales de la EVA.

### F. Gestión del desarrollo

Las historias de usuario son tomadas de acuerdo a su orden de prioridad para una EVA. Solo el facilitador podrá quitar o agregar nuevas historias de usuario de una EVA.

Las historias son presentadas en el tablero de la e-MATE (Pizarra Digital) dividido en tres columnas según las referencias: (a) Historias pendientes, (b) Historias en proceso y (c) Historias finalizadas. A su vez cada historia se subdivide en un conjunto de tareas, definidas por los desarrolladores, cuya duración y dificultad son estimadas en el mencionado VIP.

La e-MATE presenta un segundo tablero para las tareas de cada historia. Los desarrolladores pueden adjudicarse una tarea como propia. Por su parte, cada desarrollador debe agregar la

cantidad de horas invertidas en el desarrollo de una tarea, tanto en forma individual como en pareja.

A partir de la posición de las tareas de una misma historia, dentro del tablero, se calcula la posición y el estado de la historia en la Pizarra Digital. En el caso de que todas las tareas de una historia se encuentren Pendientes, la historia estará Pendiente. Si alguna tarea estuviera en Desarrollo, la historia estará en Desarrollo. Si no hubiera tareas Pendientes o en Desarrollo, la historia se encontraría Finalizada. Si alguna historia presentara un impedimento para su realización, el desarrollador podrá marcar la tarea como Impedida y describir sus causas, lo que se mostrará en la e-MATE por medio de una señal en la historia. Es responsabilidad del facilitador solucionar el impedimento y reactivar la tarea.

Una vez que una historia esté completa deberá ser revisada y aprobada por el dueño de la idea a los efectos de ser ubicada en el estado Finalizada.

La e-MATE acompaña la evolución de los procesos mediante la generación de una gráfica de avance que exhibe el volumen del trabajo realizado, el correspondiente al trabajo faltante y una proyección del plazo restante para completar las historias de la EVA. Asimismo, para cada uno de los desarrolladores, la e-MATE almacena datos vinculados con las horas trabajadas, tanto en forma individual como en pareja y la dificultad promedio de las historias y de las tareas en las que trabaja. Partiendo de estos datos, la aplicación genera información referida al rendimiento del desarrollador, según la cantidad y complejidad de cada tarea asignada. De esta manera el facilitador puede obtener, en forma precisa, una reseña sobre el desempeño de los responsables de cada historia y sobre el estado de avance del proyecto en cuestión.

### G. Documentación

En una historia, la e-MATE permite incluir requisitos no funcionales y además admite crear notas retrospectivas que podrán ser consideradas al momento de revisar los resultados de una EVA. Asimismo es factible crear comentarios en las distintas tareas o historias. Los distintos usuarios pueden utilizar estos comentarios para responder a particularidades que requieran una explicación.

La versión web de la aplicación permite exportar el conjunto de fichas de historias de usuarios de un proyecto, así como las gráficas de avance de las distintas EVA y demás información relacionada con las mismas. Una vez completadas las fichas, esta funcionalidad permite exportar la información en un documento más ágil de entregar o evaluar por los interesados o por agentes externos.

### REFERENCIAS

- [1] Letelier, P. y Penadés, M. C. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). Universidad Politécnica de Valencia. Documento recuperado de <http://www.willydev.net/descargas/masyxp.pdf> (2006)
- [2] Blanco, P., Camarero, J., Fumero, A., Werterski, A., Rodríguez, P. Metodología de desarrollo ágil para sistemas móviles: Introducción al desarrollo con Android y el iPhone. Universidad Politécnica de Madrid. Documento recuperado de [http://www.adamwesterski.com/wp-content/files/docsCursos/Agile\\_doc\\_TemasAnv.pdf](http://www.adamwesterski.com/wp-content/files/docsCursos/Agile_doc_TemasAnv.pdf) (2009)
- [3] Ableson, Frank y Sen, Robi. Android: Guía para Desarrolladores (2° Ed.) - Madrid: Anaya Multimedia (2011)

- [4] Scalone, F. Estudio comparativo de los modelos y estándares de calidad del software. Tesis de Maestría. Recuperado de <http://laboratorios.fi.uba.ar/lfi/scalone-tesis-maestria-ingenieria-en-calidad.pdf> (2006)
- [5] [5] Gamma E., Helm, R., Johnson, R. y Vlissides, J. Patrones de Diseño. Madrid: Pearson Educación (2003)
- [6] Cockburn, Alistair, Surviving Object Oriented Projects, Addison- Wesley Object Technology Series (1998)
- [7] Calderón, Amaro, Valverde Rebaza, Sarah Dámaris y Carlos, Jorge. Metodologías Ágiles. Facultad de Ciencias Físicas y Matemáticas, Universidad Nacional de Trujillo, Perú Recuperado de <http://www.seccperu.org/files/Methodologias%20Agiles.pdf> (2007)



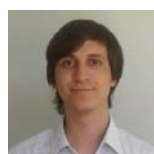
**Marisa Cecilia Tumino.** Ingeniera en Recursos Hídricos, Analista en Informática Aplicada, Magister en Ciencias Computacionales, Doctora en Educación con énfasis en Administración educativa. Desempeño profesional en (a) Instituto Francisco Ramos Mejía, (b) Universidad Adventista del Plata, (c) Instituto Juan Bautista Alberdi- Misiones, (d) Universidad de Montemorelos, México, (e) Herbert Fletcher University, Puerto Rico y (f) Universidad de Chillán, Chile.



**Juan Bournissen.** Doctorando en Tecnologías Educativas: E-learning, y Gestión del Conocimiento en la Universidad de Islas Baleares, España. Master en Ingeniería del Software obtenido en la Universidad Politécnica de Madrid. Magíster en Ingeniería del software obtenido en el Instituto Tecnológico de Buenos Aires, ITBA. Especialista en Entornos Virtuales del Aprendizaje. Profesor Universitario en Sistemas de Información. Ingeniero en Sistemas de Información. Analista Universitario en Sistemas. Profesor universitario de grado y posgrado en la modalidad presencial y a distancia. Administrador de plataformas virtuales en instituciones estatales y privadas. Formador de formadores en e-learning en Argentina y en varios países de Latinoamérica. Director de centro de cómputos. Asesor informático en sistemas de información y en tecnologías educativas y e-learning.



**Claudio José Bracalenti:** Ingeniero en Construcciones (UTN FRSF 1984). Calculista en Agua y Energía eléctrica 1979-1980. Programador en AyE entre 1981 – 1984. Ingeniería de Sistemas en AyE entre 1985 y 1992. Docente de la UTN FRSF desde 1984 habiendo enseñado en varias cátedras. Actual Profesor Titular en Diseño de Sistemas y Proyecto Final desde 1992 a la fecha en la carrera de Ingeniería en Sistemas de Información de la UTN FRSF. Coordinador de Informática en la Universidad Nacional del Litoral entre 1995 – 2000, Consejero Departamental entre 1992 y 2000 y Académico entre 1992 y 2002 en la UTN FRSF. Docente de la Universidad Adventista del Plata desde 1996 habiendo enseñado en varias cátedras. Actual Profesor Titular de Ingeniería del Software II, Administración de Recursos Informáticos e Informática aplicada desde el 2001 a la fecha en la carrera de Licenciatura en Sistemas de Información de la UAP. . Subsecretario de Planeamiento de la UTN FRSF entre 2000-2002.



**Eric Germán Schlemper:** Argentino. Estudios en curso: Licenciatura en Sistemas, Universidad Adventista del Plata (2008 - Presente). Ocupación Actual: Desarrollo de aplicaciones con orientación médica, Sanatorio Adventista del Plata (2013). Intereses: Diseño y desarrollo de aplicaciones

distribuidas. Informática Biomédica. Desarrollo de aplicaciones con metodologías ágiles.



**Silvio Rodrigo Kucharski Zuliani:** Estudiante del último año de Licenciatura en Sistemas de Información en la Universidad Adventista del Plata (UAP). Ponente en el XV Workshop de Investigadores en Ciencias de la Computación (WICC), en Paraná, Entre Ríos, con el título “Metodología Ágil basada en Telecomunicaciones”. Desarrollador de una “Nueva metodología ágil para dispositivos móviles”.