

## Experiencia sobre desarrollo con Genexus siguiendo prácticas ágiles

Loraine Gimson, Gustavo Gil

CIDIA, Facultad de Cs Exactas, Universidad Nacional de Salta

[loraine@cidia.unsa.edu.ar](mailto:loraine@cidia.unsa.edu.ar); [gdgil@unsa.edu.ar](mailto:gdgil@unsa.edu.ar)

### Resumen

Este trabajo busca describir la experiencia realizada con alumnos del último año de la Licenciatura en Análisis de Sistemas de la Universidad Nacional de Salta combinando el desarrollo basado en conocimiento (DBC), utilizando Genexus, con un desarrollo ágil. Se pretendió brindarles una práctica real de aplicar los principios ágiles a los alumnos combinando su aplicación con Genexus a dos proyectos concretos.

**Palabras clave:** desarrollo basado en conocimiento, Genexus, prácticas ágiles.

### Introducción

#### Metodologías ágiles

Hace casi dos décadas que se comenzó a buscar una alternativa a las metodologías formales o tradicionales que estaban sobrecargadas de técnicas y herramientas y que se consideraban excesivamente “pesadas” y rígidas por su carácter normativo y fuerte dependencia de planificaciones detalladas previas al desarrollo.

Las metodologías ágiles conllevan una filosofía de desarrollo de software liviana, debido a que hace uso de modelos ágiles. Se considera que un modelo es ágil o liviano cuando se emplea para su construcción una herramienta o técnica sencilla, que apunta a desarrollar un modelo aceptablemente bueno y suficiente en lugar de un modelo perfecto y complejo.

Existen actualmente una serie de metodologías que responden a las características de las metodologías ágiles y cada vez están teniendo

más adeptos. Aunque los creadores e impulsores de las metodologías ágiles más populares han suscrito el manifiesto ágil y coinciden con sus postulados y principios, cada metodología ágil tiene características propias y hace hincapié en algunos aspectos más específicos.

#### Bases de Datos del Conocimiento

Actualmente se pretende poder desarrollar software en el menor tiempo posible y con el menor costo. Para tratar de reducir el tiempo de programación, la solución no está relacionada tanto en mejorar más todavía los lenguajes de programación sino en la programación en sí. En los desarrollos de sistemas tradicionales se desarrolla y se realiza el mantenimiento con programación manual. Si se "describe" en vez de "programar", se pueden maximizar las descripciones declarativas y minimizar las especificaciones procedurales, haciendo desarrollo basado en conocimiento (DBC) y no en programación. Esta pretensión constituye un cambio esencial de paradigma e implica un choque cultural.

Los sistemas basados en conocimiento (SBC) tratan de mantener una gran cantidad de conocimiento y aportan mecanismos para manejarlo. La representación es declarativa: se separa el conocimiento del dominio de los mecanismos de deducción. Esto permite reutilizar tanto la Base del Conocimiento como los mecanismos de razonamiento [1]. En este trabajo, el paradigma de DBC al que se hace referencia, consta de dos partes principales: la Base de conocimiento y el motor de inferencia o proceso de razonamiento, donde el objetivo es obtener por inferencia la base de datos y los programas para manejar los objetos descritos por el usuario. Pero, no busca emular capacidades de un dominio experto. Este

paradigma está orientado más a los sistemas de gestión. Permiten partir de la descripción de las visiones de todos los usuarios del sistema a desarrollar y genera la base de datos y los programas necesarios para satisfacer dichas visiones.

La Base del Conocimiento inicialmente tiene asociado un conjunto de mecanismos de inferencia y contiene reglas generales que son independientes de cualquier aplicación particular. Al describir la realidad del usuario objeto, se almacenan las descripciones en el Modelo Externo. El sistema, automáticamente, captura todo el conocimiento contenido en el Modelo Externo y lo sistematiza, agregándolo también a la Base del conocimiento. Adicionalmente, sobre el conocimiento anterior, el sistema infiere lógicamente un conjunto de resultados que ayudan a mejorar la eficiencia de las inferencias posteriores. En este tipo de desarrollo el foco está en ocuparse únicamente del Modelo Externo (el “qué”) y abstenerse de tratar la Base del Conocimiento, que lo contiene y lo mantiene, (y que forma parte del “cómo”).

### Genexus

Genexus es una herramienta de desarrollo de software basada en conocimiento (Knowledge-based Development Tool), orientada principalmente a aplicaciones de clase empresarial para la web, plataformas Windows y Smart Devices producida en Uruguay por la empresa Artech. Según los propios autores Genexus es, esencialmente, un sistema que permite una buena administración automática del conocimiento de los sistemas de negocios [2]. Parte de las visiones de los usuarios; captura su conocimiento y lo sistematiza en una base de conocimiento. A partir de esta última, Genexus es capaz de diseñar, generar y mantener de manera totalmente automática la estructura de la base de datos y los programas de la aplicación, es decir, los programas necesarios para que los usuarios puedan operar con sus visiones. [3] El desarrollador describe sus aplicaciones en alto nivel (de manera mayormente declarativa), a partir de lo cual se

genera código para múltiples plataformas. Genexus incluye un módulo de normalización, que crea y mantiene la base de datos óptima (estructura y contenido) basada en las visiones de la realidad descritas por los usuarios utilizando un lenguaje declarativo, ver Figura 1. [4].

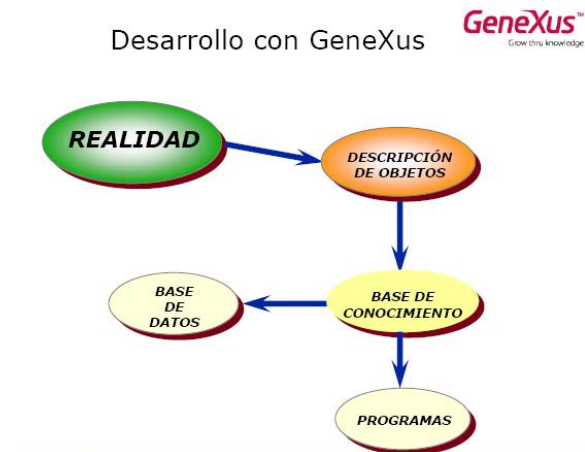


Figura 1: Desarrollo con Genexus [5]

Genexus [3] es una herramienta de especificación de sistemas de información basada en la aplicación de un modelo con rigurosa fundación matemática que permite capturar e integrar las visiones de los usuarios en bases de conocimiento a partir de las cuales genera, mediante ingeniería inversa y procesos de inferencia, bases de datos normalizadas y aplicaciones completas; para diferentes plataformas de destino a partir de una misma especificación básica, pudiendo mantenerlas en forma automatizada ante cambios en los requerimientos [6] [7] [8].

Una parte considerable del total de software producido en Uruguay se elabora con esta herramienta creada y mantenida por una empresa uruguaya, Artech y en Salta existen varias empresas del ámbito público y privado que también las utilizan. También está creciendo su uso en noroeste argentino y también en Bolivia y Paraguay.

### Contexto de la experiencia

En la carrera de la Licenciatura en Análisis de Sistemas de la Universidad Nacional de Salta

existe una cátedra del último año llamada *Desarrollo Basado en Conocimiento* donde se forma a los alumnos en este tipo de desarrollos utilizando la herramienta Case Genexus. La Universidad tiene un convenio con Genexus y los alumnos al finalizar el cursado pueden rendir su certificación en Genexus. Esta materia viene dictándose hace 3 años y recién en 2015 se planteó realizar prácticas reales combinando DBC con prácticas ágiles.

Los lineamientos ágiles para combinar las prácticas ágiles con Genexus fueron formulados en el marco de un trabajo de tesis de maestría de Ingeniería de Software de la Universidad Nacional de la Plata [13]. A través de esta propuesta, de un conjunto de prácticas y lineamientos, se buscaba una manera de lograr que los equipos de trabajo de DBC puedan acercarse de una manera simple y sin implicarles demasiados cambios drásticos al desarrollo ágil inclusive trabajando en más de un proyecto de desarrollo simultáneamente. Posteriormente, podrían ir incorporando mayores conocimientos sobre metodologías ágiles, obtendrán experiencias y estarán en condiciones de adaptarlo de la mejor manera a su equipo de trabajo y proyectos a desarrollar.

### **Propuesta ágil**

La propuesta planteada por Gimson [13] se divide en tres secciones principales, como lo plantea la mayoría de las metodologías ágiles: Iniciación ágil, Producción y Ritual de finalización del proyecto. La primera busca compartir una visión del proyecto a realizar, clarificar los objetivos y razones por las que se realiza el proyecto, la segunda se centra en el desarrollo en sí del producto esperado y por último se encuentra el cierre del proyecto, que incluye la entrega final del producto. La propuesta también contempla una serie de reuniones a realizar con diferentes objetivos y define una serie de roles y artefactos a utilizar.

Como toda metodología ágil plantea: fomentar la interacción entre los miembros del equipo y con el cliente, sobre todo a través de

reuniones, estar abiertos a cambios de requerimientos y de planificación, y enfocarse más en entregar al cliente software que funcione más que una documentación completa.

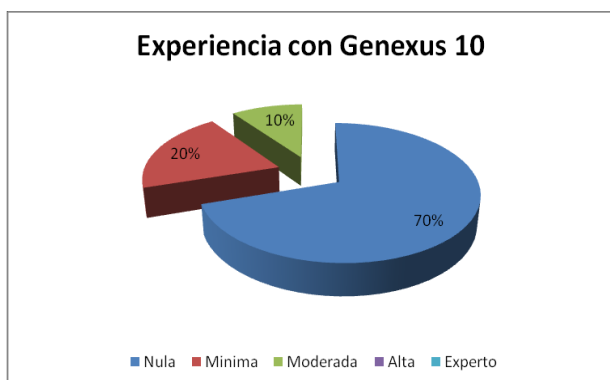
### **Alumnos participantes**

La cátedra *Desarrollo Basado en Conocimiento* contó con 10 alumnos durante el 2015 y fueron divididos en dos equipos (Grupo 1 y Grupo2) de cinco integrantes cada uno, para que cada equipo trabaje de manera independiente sobre los mismos proyectos.

Se creó en la plataforma educativa *Moodle* un espacio para poder compartir información desde la cátedra con los alumnos y donde también cada grupo pudiera compartir entre ellos. La información vertida en la plataforma ayudaría también al seguimiento de la experiencia.

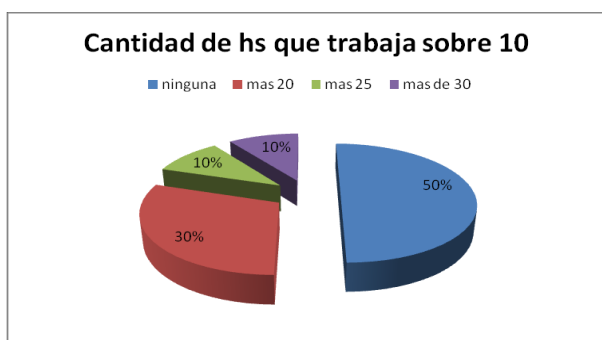
Los alumnos involucrados en esta experiencia básicamente se habían formado en el desarrollo tradicional, Proceso Unificado. Si bien, en una cátedra de Análisis y Diseño se les describió en qué consisten, brevemente y de manera superficial, este tipo de desarrollos, su única experiencia en el ámbito académico era con metodologías tradicionales. En el ámbito universitario, ninguno realizó previamente alguna actividad de desarrollo siguiendo una metodología ágil.

Respecto a la experiencia previa con Genexus antes de iniciar la asignatura, un 10% de ellos consideraba que tenía experiencia moderada y venía trabajando con Genexus hace un año (Ver Figura 2). Otro 20% consideraba que tenía conocimientos mínimos siendo la experiencia de uno de ellos de un año de trabajo y del otro menos de un mes. Pero la cátedra consideraba que todos tenían los conocimientos mínimos para realizar esta experiencia.



**Figura 2:** Experiencia previa con Genexus dentro del grupo de alumnos.

El 50% de los alumnos de la experiencia se encontraba trabajando, y algunos de ellos acababan de ser contratados. En Figura 3 se muestra la cantidad de horas dedicada por cada alumno al trabajo.



**Figura 3:** Cantidad de horas dedicadas al trabajo dentro del grupo de alumnos.

Para determinar los miembros de cada grupo, se buscó que en cada uno participara uno de los alumnos con mayor experiencia en el DBC con Genexus, quienes llevaban un año trabajando con este tipo de desarrollo. Dicha división fue planteada desde la cátedra, una vez que los principales conocimientos sobre el DBC estaban afianzados en los alumnos.

No se debe perder de vista ciertas restricciones en la experiencia realizada:

- Los alumnos participantes de la experiencia no fueron alumnos que estuvieran muy familiarizados con las metodologías ágiles, como se mencionó anteriormente, sino más bien estaban formados en metodologías tradicionales.

- Los alumnos fueron introducidos al DBC recién al estar cursando esta asignatura, si bien un 30 por ciento ya se encontraba trabajando fuera del ámbito universitario con DBC, solo un 20 por ciento tenía un año de experiencia.
- Los alumnos tenían otras obligaciones, otras materias, trabajos por lo que el tiempo de dedicación al desarrollo fue durante las horas de la materia (10 hs semanales), y otras horas que pudieran dedicar (mayormente durante los fines de semana). Se solicitó, al iniciar la experiencia, que por lo menos dediquen otras 10 horas adicionales.

### Capacitación y armando de equipos

Mediante una charla se planteó a ambos grupos la propuesta de trabajo a seguir. Se expuso claramente las tres etapas de la propuesta de desarrollo, dejando bien en claro los objetivos de cada una. Se explicaron todos los tipos de reuniones involucrados en cada etapa. Se detallaron los diferentes roles y se mencionaron los artefactos a utilizar. Se entregó un documento con todos los detalles de la propuesta y se les realizó un coloquio la siguiente clase para asegurar que todos comprendieran la forma de trabajo antes de iniciar. Todos los alumnos parecieron estar entusiasmados por esta propuesta práctica de combinar lo aprendido de DBC y las Metodologías Ágiles. Esto se vio reflejado en que ninguno desaprobó el coloquio, significando así que dedicaron tiempo de calidad a entender cómo se pretendía trabajar.

La idea que los requerimientos pudieran variar a lo largo del desarrollo del proyecto no fue simple de comprender, creían imposible poder llegar a un desarrollo aceptable bajo estas condiciones. La falta de necesidad de documentar formalmente los requerimientos y diseños del software a desarrollar fue vista con mejor agrado por otros. La necesidad de realizar todas las reuniones detalladas en la propuesta tampoco fue simple de entender, incluso algunos pensaban que podían llegar a

ser opcionales. Pero, luego de las aclaraciones necesarias, los miembros de los dos grupos estuvieron en condiciones de iniciar la experiencia práctica.

Cada grupo pudo distribuir los roles según su propio criterio sin intervención externa. Cada grupo definió los estándares a seguir, si bien estaban basados en patrones. Cada grupo definió las actividades a considerar en el tablero de trabajo incluyendo la Definición de Terminado (DoD) de cada una.

No se planteó como obligatorio el uso de ninguna herramienta específica, más bien se les dio libertad a cada grupo de buscar y decidir cuáles utilizar. Aunque se les exigió que utilicen por ejemplo alguna pizarra, sea física u on-line.

Solo los clientes fueron personas externas que representaban las necesidades de cada negocio a implementar. Estos clientes no eran alumnos de la cátedra, sino personas con necesidades reales en las áreas planteadas para el desarrollo.

Durante todo el tiempo que se realizó la experiencia se brindó asesoramiento sobre la propuesta metodológica a seguir y sobre el trabajo con Genexus para realizar el DBC. Pero, se dejó que cada equipo trabaje libremente.

## **Descripción de la experiencia**

En esta sección se procederá a detallar algunas características del trabajo de los grupos, sin profundizar en detalles específicos ya que no son en sí el objeto del presente trabajo. Para realizar este análisis se tuvieron en cuenta los informes redactados en la plataforma, la información en el tablero de tareas, la encuesta realizada al final y otros recursos de distribución utilizados dentro de la plataforma como por ejemplo wikis y foros.

A lo largo de la experiencia se buscó respetar todas y cada una de las etapas y reuniones propuestas en la tesis de Gimson y también

cubrir por lo menos los roles esenciales allí planteados. Al principio los miembros del equipo estaban temerosos de participar en las reuniones, por ser una forma diferente a cómo realizaron todas sus actividades de desarrollo a lo largo de la carrera. Poco a poco las reuniones se hicieron más fluidas y cada uno de los integrantes del grupo participaba libremente expresando sus opiniones y estas eran respetadas por el resto.

En toda la experiencia se buscó tener un rol muy pasivo, como espectador, para simplemente recabar información y aclarar dudas o aconsejar en situaciones especiales.

### **• Auto-organización del grupo**

Antes de iniciar específicamente la experiencia los grupos debieron definir los roles que desempeñarían cada uno, los lineamientos y estándares a seguir, las herramientas a usar y la forma de estructurar la pizarra de tareas definiendo las actividades asociadas a cada uno y posteriormente, el DoD de cada actividad. Los alumnos no estaban acostumbrados a tener tanta libertad a la hora de definir la forma en que iban a trabajar. Ambos equipos realizaron adaptaciones en la forma de trabajo a lo largo de la experiencia. Todos admitieron haber realizado modificaciones a la manera que habían determinado trabajar, ya sea por cambiar ciertas herramientas o, por ejemplo, por modificar lineamientos (como establecer ciertos horarios para que todos estén conectados para trabajar en conjunto). Ambos equipos reconocieron que es importante y enriquece el trabajo, poder tener un equipo auto-organizado.

La definición de roles realizada en esta propuesta permitió que todos entendieran perfectamente cada uno de los roles que podían desempeñar. Dentro de la reunión de auto-organización cada grupo asignó los roles de común acuerdo dentro de sus integrantes.

## Reuniones

Se realizaron todas las reuniones en ambos equipos, si bien algunos admitieron haber cambiado horario o día de algunas reuniones por situaciones específicas. No hubo uniformidad a la hora de comentar cuál consideraron que fue la reunión más importante (ver Figura 4). Pero, vale la pena recalcar que, ninguno consideró que alguno de los tipos de reuniones planteadas fuera innecesario.



Figura 4: Reunión más importante

Las reuniones con los clientes se realizaban para cada grupo el mismo día, una después de la otra. Lo que se buscó desde la cátedra fue alternar el grupo que tenía la reunión en primer lugar. A medida que los alumnos ganaron más confianza en estas reuniones, se los notó más cómodos al interactuar con los clientes.

La priorización de las historias de usuario no fue sencilla para ninguno de los clientes. Pero, los analistas recordándole la visión previamente definida, los ayudaron a realizar esta tarea, como se pudo observar de manera directa.

Algunas reuniones del equipo se realizaron de manera virtual si bien todas las reuniones con el cliente fueron cara a cara.

Dentro de las reuniones realizadas por el equipo vale la pena recalcar aspectos

observados en las reuniones de reflexión interna. Estas reuniones permitieron:

- Analizar las razones de demoras y problemas ocasionados por el uso de una herramienta en particular, decidiendo finalmente reemplazarla por otra y modificando la manera de trabajar.
- Comentar que no resultaba tan simple trabajar en dos proyectos simultáneos, pero veían que se lograba avanzar bastante rápido con la implementación.
- Valorar que las pruebas sean realizadas por otra persona ya que permitían detectar errores y decidieron esforzarse más en la realización de las pruebas.
- Determinar buscar mejoras visuales en los sistemas que guíen mejor al usuario.
- Plantear situaciones que afectaban la productividad del equipo (en uno de los grupos): el abandono de uno de los integrantes y el poco cumplimiento de otro. La persona cuestionada por su bajo rendimiento explicó las razones, se disculpó y volvió a asumir el compromiso de trabajar las horas pactadas. Para avanzar con el proyecto a pesar del abandono de uno de los integrantes, consensuaron en tratar de comprometerse un par de horas más cada uno.
- Reiterar el compromiso de trabajar responsablemente cada miembro del equipo. Alentarse a seguir trabajando para concluir los proyectos.
- Un grupo aprovechó para destacar los logros alcanzados, aún trabajando con menos personas que el otro equipo y también resaltar el compromiso realizado de cada miembro del equipo, alentando a seguir trabajando de igual manera.
- Otro grupo aprovechó para analizar la falta de tiempo dedicada al proyecto y concluir que esta se debió al tiempo dedicado a la preparación de los exámenes del turno extraordinario. Todos se comprometieron a avanzar lo máximo posible.

Analizando los resultados de cada una de las reuniones de evaluación y reflexión interna se



pudo observar que hubo lugar para alentar a los integrantes, lugar para resolver problemas técnicos y proponer mejoras, lugar para buscar mayor compromiso de cada uno. Por todo esto se considera que realmente esta reunión realmente necesaria permitió a los alumnos trabajar como equipo, escuchándose, alentándose y exigiéndose entre ellos.

En otro tipo de reuniones el cliente tenía posibilidad de aceptar o solicitar cambios a lo desarrollado, modificar el orden de prioridades de los requerimientos e inclusive incorporar nuevos. Al respecto de los cambios solicitados en esta reunión, se consultó en la encuesta final si el equipo aceptó los cambios de requerimientos realizados por el cliente obteniendo una respuesta afirmativa de un 89% y el 11% restante no contestó.

Y respecto a la complejidad para realizar los cambios solicitados se puede observar según la Figura 5 que:

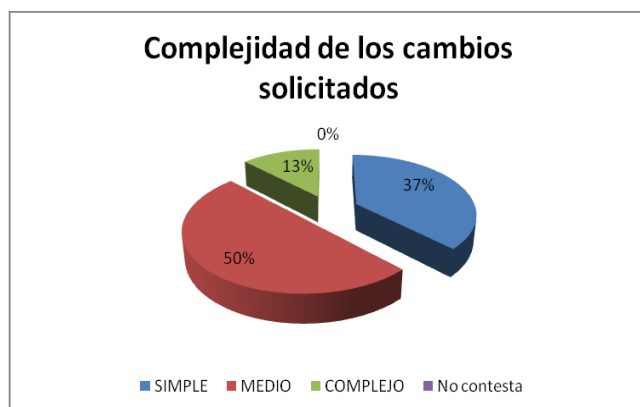


Figura 5: Complejidad de los cambios solicitados.

- el 37% consideró a la complejidad baja debido a que el analista logró entender bien los cambios y expresar de manera concreta al grupo los mismos, también a que Genexus se adapta fácilmente a la metodología ágil y porque no fueron grandes cambios.
- el 50% consideró una complejidad media, aduciendo algunos por cuestiones de estar trabajando con un trial, otro adujo que para ciertas funcionalidades se debió profundizar ciertos conocimientos de Genexus y otro por ejemplo destacó que

los cambios solicitados tenían cierta complejidad de diseño.

- un 13% consideró que fue complejo debido a que no se contaba con todo el conocimiento necesario sobre Genexus.

### Reutilización

Se hizo mucho hincapié en la reutilización, sobre todo al inicio de la experiencia. Como los alumnos no tenían demasiada experiencia y objetos desarrollados previamente se sugirió que también en un desarrollo era válido reutilizar objetos creados por otros y se sugirió buscar en la nube de GXServer. Esto también causó una reacción de sorpresa, dado que se les estaba permitiendo usar algo no desarrollado por ellos. Pero al final de la experiencia el 89% concluyó que es más rápido aunque en algunos casos no tanto como hubieran imaginado. Para el 11% restante es más rápido solo en algunos casos, y consideró más rápido desarrollar desde cero que modificar propiedades y atributos.

Caso especial fueron las reutilizaciones de objetos que no necesitaron adaptarse y se reutilizaron sin modificaciones, como por ejemplo aquellos objetos involucrados con la seguridad del sistema.

### Reuniones de evaluación final

Las reuniones de finalización de proyecto realizadas con cada uno de los clientes y luego reflexionando internamente cada equipo sobre el trabajo realizado vale la pena tratarlo en un ítem aparte.

En estas reuniones finales con los clientes, se puede destacar el entusiasmo observado al presenciar la reunión con uno de ellos. En la reunión con cada grupo, comentó haber estado accediendo al sistema desde su trabajo y que otra persona de su empresa también había navegado por el sitio fina mostrándose muy conformes con lo obtenido. Otra cuestión a destacar es que un cliente hizo un comentario al grupo que había tenido la deserción de uno de los integrantes, valorando su esfuerzo y

resultados obtenidos. Esto permite constatar que las reuniones cara a cara realmente generan un vínculo entre el equipo de desarrollo y el cliente. Posteriormente, cuando ese grupo estuvo sin la presencia del cliente remarcaron ese comentario, y reconocieron haber aprendido muchas cosas.

Después que cada cliente se retiró de la reunión, cada grupo buscó aprender y extraer cosas positivas y negativas de lo que había ocurrido. Algunos de los comentarios surgidos en estas reuniones, relevados de las encuestas fueron los siguientes:

- el equipo no quedó del todo conforme con algunos detalles de los productos, por las limitaciones de haber trabajado con una versión trial de Genexus
- el equipo estuvo muy conforme con la forma de trabajar en equipo
- el equipo comentó diferentes cosas que “se podrían haber hecho” para considerar a futuro.
- el equipo enfatizó la reutilización de objetos para el proyecto que seguía en desarrollo (opinión surgida en la reunión de reflexión final para el proyecto de la librería)

Analizando estos puntos se puede decir que hubo un cierto aprendizaje basado en las experiencias vividas que enriqueció cada grupo y este era el objetivo de esta reunión. Lograron extraer comentarios positivos y negativos del proyecto y en el último punto descripto, donde se sugería maximizar la reutilización, incluso llegaron a proponer “mejoras” al proceso de desarrollo.

### Dificultades en la experiencia

Para poder obtener mayor información de la experiencia de cada alumno, se solicitó en la encuesta nombrar las tres dificultades mayores enfrentadas indicando para cada uno el nivel del impacto, pudiendo optar por: Muy Bajo, Bajo, Moderado, Alto, Muy Alto.

Se pueden nombrar seis que fueron comentarios repetitivos entre los alumnos.

- El uso de GxServer
- Poco conocimiento de Genexus
- La coordinación de horarios
- Utilización de una versión trial en la nube
- Los tiempos para realizar las entregas
- Posibilidad de compartir lugar de trabajo

Se muestra en la Figura 6 cada una de las dificultades mencionadas, sus porcentajes de referencias (mencionado) y nivel de impacto definido para cada uno.

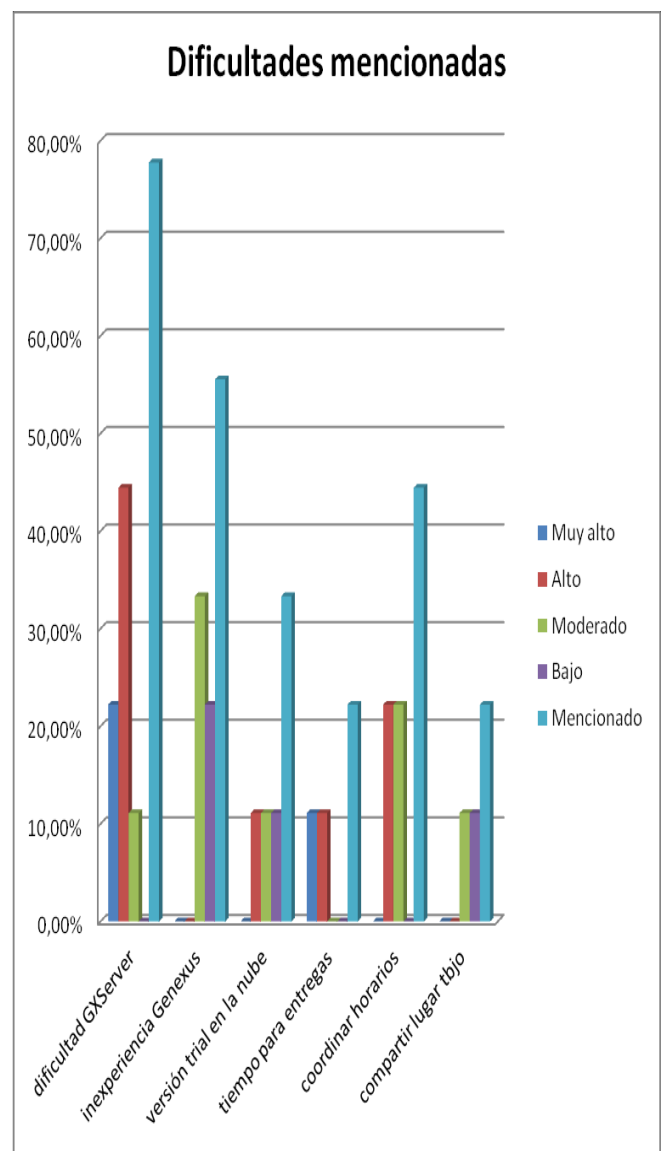


Figura 6: Dificultades mencionadas por los alumnos.



Otros tres puntos fueron resaltados por uno sólo pero vale la pena nombrarlos:

- Tiempo para documentar (Muy alto)
- Hacer que todos los integrantes trabajen relativamente iguales (Alto)
- Trabajo simultáneo en diferentes sistemas (Bajo)

En un proyecto real, trabajando con la versión oficial de Genexus muchas de estas dificultades desaparecerían. Actualmente se está buscando trabajar con otra licencia de Genexus, a través de un nuevo convenio con la Facultad donde no tengan limitaciones con la herramienta.

Por otro lado, el poder trabajar en un lugar común, como se mencionó anteriormente reduciría muchas complicaciones a la hora de compartir la Base del Conocimiento, aún sin utilizar GXServer. El coordinar horarios comunes de trabajo es más probable cuando existe una relación laboral y no en una experiencia universitaria, donde queda a responsabilidad de cada uno debe buscar cumplir con los horarios que demanda el grupo.

Siempre existirá una presión para tratar de entregar lo máximo posible en la menor cantidad de tiempo, independientemente del proyecto y metodología a utilizar. Esto es común en cualquier proyecto de desarrollo de software, no solamente en esta experiencia. Pero, al poder ir entregando software funcionando al cliente, se puede ir cumpliendo con sus necesidades más urgentes para su negocio y esto en la mayoría de las situaciones siempre será valorado. Además se va entrenando al alumno en tratar de cumplir con los clientes en tiempos pactados.

Y por último, en todo equipo siempre habrá personas más comprometidas que otras, pero a través de las reuniones de seguimiento se puede persuadir a aumentar el compromiso. De esta manera todos trabajarán dando lo mejor de cada uno. Por lo tanto si bien se

planteó la dificultad de hacer que todos los integrantes trabajen relativamente iguales puede aparecer en cualquier metodología, brindando esta propuesta, basándose en prácticas ágiles, las herramientas para tratar de minimizar la falta de compromiso. Esta práctica permitió que los alumnos dentro de los grupos se exijan mutuamente para cumplir con sus responsabilidades, logrando así también una madurez profesional muy importante a la hora de ejercer su profesión.

## **Conclusión de la experiencia**

En términos generales, según la experiencia se podría decir que la propuesta cumplió su objetivo de introducir a quienes están desarrollando con DBC al desarrollo ágil. El problema de trabajar con una versión gratuita desvió un poco la atención de aplicar la propuesta a lograr implementar con Genexus ciertas historias de usuario.

Costó al principio hacer que los alumnos participen en las reuniones pero luego los resultados fueron muy buenos en cada tipo de reunión diferente. Se pudo constatar que no es siempre posible contar con los stakeholders en todas las reuniones, se debería proponer alternativas dentro de la propuesta en estos casos. Se plantea entonces, utilizar encuestas para obtener cierta retroalimentación o realizar visitas al lugar de trabajo.

La experiencia realizada para aplicar la propuesta será realizada nuevamente el año próximo a los alumnos que estén cursando la asignatura. Esto se debe a que, como resultado de la misma, los alumnos no solo entendieron más sobre DBC utilizando Genexus, sino que ellos mismos reconocieron haber logrado percibir cómo es la filosofía detrás de las metodologías ágiles. Algunos de los alumnos que participaron, consideraron que la experiencia ameritaba ser una asignatura completa, para poder realizar una experiencia más profunda y debatir más sobre los conceptos ágiles utilizados. El responsable de la materia también manifestó su conformidad a los resultados, sobre todo por haber logrado un

gran compromiso y participación por parte de los alumnos. Los mismos alumnos comentaron a otros sobre la experiencia y hay muchas expectativas para el próximo año entre ellos.

## Bibliografía

[1] Alonso Betanzos A. Ingeniería del conocimiento: Aspectos metodológicos, Pearson Education. ISBN: 9788420541921 Año 2004.

[2] Gonda, B., Breogán, Jodal, N. GeneXus: Filosofía. Artech Consultores S. R. L. Última actualización: 2010.

[3] Sitio oficial de Genexus. Sitio:<http://www.genexus.com/>. Accedido: 05/07/2014.

[4] Visión General de Genexus. Artech Consultores S.R.L. Uruguay. Octubre 2008. Sitio:[http://www.genexus.es/wp-content/uploads/2010/06/vision\\_general\\_gx\\_es\\_p2009.pdf](http://www.genexus.es/wp-content/uploads/2010/06/vision_general_gx_es_p2009.pdf). Accedido:05/08/2013.

[5] Gonda, B., Jodal, N. Desarrollo Basado En Conocimiento - Filosofía Y Fundamentos Teóricos De Genexus. Artech. Mayo de 2007. Sitio:[http://www.genexus.es/wp-content/uploads/2010/06/desarrollo\\_basado\\_en\\_el\\_conocimiento.pdf](http://www.genexus.es/wp-content/uploads/2010/06/desarrollo_basado_en_el_conocimiento.pdf). Accedido: 16/09/2011.

[6] Márquez Lisboa D., Fernández C. Genexus Rocha – Episodio Uno. Editorial Grupo Magro. Montevideo, Uruguay. 2007

[7] Latorres E.,Salvetto P., Larre Borges U., Nogueira, J. Una herramienta de apoyo a la gestión del proceso de desarrollo de software. IX Congreso Argentino de Ciencias de la Computación (CACIC 2003). La Plata, Argentina.

[8] Salvetto P. Modelos Automatizables de Estimación muy Temprana del Tiempo y Esfuerzo de Desarrollo de Sistemas de Información. Tesis de Doctorado. Univ. Politécnica de Madrid. Doctorado conjunto en ingeniería informática UPM-ORT. Año: 2006. Sitio: [http://oa.upm.es/367/01/PEDRO\\_SALVETTO\\_LEON.pdf](http://oa.upm.es/367/01/PEDRO_SALVETTO_LEON.pdf). Accedido: 16/09/2011.

[9] Gil, G., Gimson, L., Ramírez, J., Aballay, P., Ortega, V., Torres, M. Metodologías ágiles y desarrollo basado en conocimiento. (WICC 2011). RedUNCI Universidad Nacional de Rosario (UNR) - Rosario – Santa Fe - 5 y 6 de mayo de 2011, Rosario.

[10] Gil, G., Arias Figueroa, D., Gimson,L., Ramírez, J., Silvera, J. Metodologías ágiles y desarrollo basado en el conocimiento, evaluación cuantitativa de F/OSS para la reutilización, normas ISO y su aplicación en centros educativos. (WICC 2012) Universidad Nacional de Misiones (UNM) - Posadas – Misiones – 26 y 27 de abril de 2012, Posadas.

[11] Gil, G., Gimson, L. Silvera, J. Desarrollo basado en conocimiento siguiendo prácticas ágiles. (WICC 2014). RedUNCI Universidad Nacional de Tierra del Fuego (UNTDF) – Ushuaia – Tierra del Fuego – 7 y 8 de mayo de 2014, Ushuaia.

[12] Gimson, L., Gil, G., Arias Figueroa, D. Desarrollo basado en conocimiento siguiendo prácticas ágiles. (WICC 2015). RedUNCI Universidad Nacional de Salta (UNSa) – Salta Capital – Salta – 16 y 17 de abril de 2015, Salta.

[13] Gimson, L. Desarrollo Basado en conocimiento siguiendo prácticas ágiles. Tesis de Maestría en Ingeniería de Software. Facultad de Informática. Universidad Nacional de La Plata (UNLP). 2015. Sitio: <http://sedici.unlp.edu.ar/handle/10915/50431>. Accedido: 01/03/2016.

[14] Rasmusson, J., The Agile Samurai: How Agile Masters Deliver Great Software. PragmaticBookshelf. Año 2009.

[15] Beck, K., et.al, Manifesto for Agile Software Development, Sitio: <http://agilemanifesto.org/>.

[16] Parasuraman Narayan - Understanding and Managing Agile Transitions. Julio 2011. Sitio: <http://www.modernanalyst.com/Resources/Articles/tabid/115/articleType/ArticleView/articleId/1926/Understanding-and-Managing-Agile-Transitions.aspx>. Accedido: 15/12/2011.