

# Un escenario de Role Playing para la enseñanza de la Gestión de Proyectos de Software \*

Osiris Sofia<sup>\*\*</sup>, Esteban Gesto<sup>\*\*\*</sup>

Universidad Nacional de la Patagonia Austral  
Lisandro de la Torre 1070  
CP 9400. Río Gallegos, Santa Cruz, Argentina  
Tel/Fax: +54-2966-442313/17  
{osofia;gesto}@unpa.edu.ar

## Resumen

En este trabajo se describe una experiencia de enseñanza de Ingeniería de Software en la Unidad Académica Río Gallegos de la Universidad Nacional de la Patagonia Austral que integra conocimientos de la asignatura “Ingeniería del Software” y asignaturas precedentes a través de la técnica conocida como *Role Playing*, mediante una actividad integradora definida como *Trabajo de Campo* que se realiza durante todo el semestre y prepara a los alumnos para encarar exitosamente la realización del Trabajo Final de Carrera, denominado “Proyecto de Software I” y que se efectúa bajo la supervisión de un tutor.

**Palabras claves:** Ingeniería de Software, Role Playing, Trabajo en Grupo, Proyecto Final de Carrera.

## 1. Introducción

La Unidad Académica Río Gallegos de la Universidad Nacional de la Patagonia Austral ofrece las carreras de Analista de Sistemas (3 años) y Licenciatura en Sistemas (4 años) orientadas a formar profesionales con un perfil de Ingeniero de Software. Los planes de ambas tienen una estructura curricular que requiere que los alumnos desarrollen un proyecto final que consiste en un desarrollo de software para un cliente real, permitiendo a los egresados la posibilidad de salir al mercado con una experiencia concreta de desarrollo, basándose en los principios de la Ingeniería de Software. Estos espacios curriculares, denominados “Proyecto de Software I” y “Proyecto de Software II” tienen la característica de trabajo tutorado. Sin embargo, no existe en los planes

mencionados un espacio curricular previo previsto para que los alumnos puedan integrar los conocimientos técnicos y de gestión adquiridos en las asignaturas precedentes para poder encarar con éxito el Trabajo Final de Carrera. La modificación del plan de estudios se presentaba como una tarea altamente dificultosa, debido a que los planes de estudio pertenecen al ámbito de toda la Universidad, por lo que una modificación implica la coordinación con el resto de las Unidades Académicas.

Teniendo en cuenta estas falencias, se implementó en el espacio curricular denominado “Ingeniería de Software” una práctica integradora grupal, a través de la técnica de Role - Playing, denominado internamente “*Trabajo de Campo*”.

El Role Playing es una técnica a través de la cual se simula una situación que se presenta en la vida real. Al practicar esta técnica se adopta el papel de un personaje concreto y se crea una situación como si se tratara de la vida real [16].

El diseño e implementación de esta actividad considera las recomendaciones del Comité Curricular de RedUNCI [10], de incorporar asignaturas dedicadas a la integración de conocimientos en aplicaciones concretas.

## 2. Antecedentes

En la Unidad Académica Río Gallegos de la Universidad Nacional de la Patagonia Austral (UARG-UNPA), en el año 1995 [11, 12] es aprobado el plan de estudios de la carrera “Analista de Sistemas”, como reemplazo al plan de “Analista Programador Universitario”, que se encontraba en vigencia desde el año 1991. En el nuevo plan se establece el requisito para obtener el título, de acuerdo a la tendencia nacional e internacional de ese momento:

*Para obtener el título de Analista de Sistemas se deberá llevar adelante un proyecto de desarro-*

\* Este trabajo fue financiado por la Universidad Nacional de la Patagonia Austral, Santa Cruz, Argentina’

\*\* Profesor Adjunto, cátedra Ingeniería de Software, UNPA-UARG

\*\*\* Auxiliar Ayudante de Docencia, cátedra Ingeniería de Software, UNPA-UARG

*llo de software en forma integral dirigido por un tutor y evaluado por una comisión Ad-Hoc. Para iniciar el trabajo el tutor determinará los requisitos previos y aceptará el problema seleccionado por el alumno. Se estima que para el desarrollo del proyecto, el alumno deberá afectar una carga horaria de 80 horas en el segundo cuatrimestre, teniendo que tener todas las asignaturas aprobadas para su acreditación final.*

Dentro del plan de estudios se establecen además las asignaturas a cursar de las diferentes áreas: programación, hardware y software de base, sistemas. Sin embargo la totalidad de las asignaturas están destinadas a abordar temas específicos, no existiendo una asignatura destinada a que los alumnos integren los conocimientos impartidos en el resto de los espacios curriculares y los prepare para encarar el desarrollo de un proyecto software en su totalidad. De esta manera los primeros estudiantes que culminaron el cursado de las asignaturas del plan se encontraban con enormes dificultades para concretar el Trabajo Final de Carrera, extendiéndose mucho más allá de los tiempos previstos para el desarrollo del mismo.

Esta situación fue detectada poco tiempo después de implementado el nuevo plan de estudios [9, 6, 2]. Sin embargo, en la UNPA los planes de estudio pertenecen no a una Unidad Académica en particular sino a todo el sistema, eligiendo las distintas Unidades Académicas qué ofertas académicas realizar en cada año. La carrera Analista de Sistemas se ofrece también en las Unidades Académicas de Caleta Olivia y Río Turbio. Esto implica que ante la necesidad de una modificación al plan de estudios es necesario consensuar entre todas las Unidades Académicas involucradas, lo que naturalmente conlleva grandes esfuerzos y plazos muy dilatados.

### **2.1. Ingeniería de Software**

En el año 1997 se dicta por primera vez el espacio curricular denominado “Ingeniería de Software”, cuyos contenidos representan la enseñanza de la Gestión de los Proyectos de Software. Esta asignatura se encuentra ubicada en el 2º cuatrimestre de 3º año, por lo que es una de las últimas materias a cursar por los alumnos. Cuenta con 6 horas semanales de cursado. En el año 1998 se hace cargo de la asignatura el actual equipo de cátedra. Al producirse los primeros egresos y detectar las dificultades para cumplimentar el Proyecto de Software I, se evalúa la posibilidad de incorporar, dentro de las atribuciones del responsable de la cátedra, alguna actividad que permita al menos mitigar la falencia observada.

En el mismo año, 1998, se decide por lo tanto

incorporar, además de las tradicionales actividades prácticas para cada tema o módulo dictado, un trabajo integrador, continuación de un trabajo similar comenzado en la asignatura precedente, “Análisis y Diseño de Software” para su realización en forma personal ó en pequeños grupos de 2 ó 3 personas como máximo. La implementación de trabajos integradores no es nueva en la enseñanza del desarrollo de software [8, 1, 4]. Tomando en cuenta antecedentes de otras universidades, esta experiencia se desarrolla durante los años 1998 y 1999, pero no brinda buenos resultados, ya que el cúmulo de actividades para cada alumno es excesivo y en la mayoría de los casos no se alcanzan los objetivos planteados.

Finalmente en el año 2000 se implementa la actual metodología de trabajo, consistente en un Trabajo de Campo integrador que se realiza durante toda la cursada, con un *único* grupo de desarrollo donde cada alumno asume un rol específico que mantendrá durante todo el semestre. [7]

## **3. Metodología**

Para poder obtener la aprobación de la cursada, los alumnos de la asignatura Ingeniería de Software deben cumplimentar los siguientes requisitos:

- Asistencia a un mínimo del 80 % de las clases
- Aprobación de la totalidad de los trabajos prácticos
- Aprobación de las guías de lectura de la bibliografía obligatoria
- Aprobación de los parciales o sus respectivos recuperatorios
- Entrega y aprobación por parte del cliente del Trabajo de Campo

Para este último ítem, los objetivos principales son:

- Que el alumno logre un aprendizaje significativo e integrador de los contenidos.
- Aplicar los conceptos y técnicas estudiados durante la asignatura en el desarrollo de un proyecto de software concreto.
- Favorecer el trabajo en equipo potenciando las capacidades de comunicación y coordinación.

Para el Trabajo de Campo, el equipo de cátedra propone un problema real en el que desempeña el papel de cliente. Los alumnos asumen los distintos roles de un grupo de desarrollo de software

y aplican sucesivamente los métodos y técnicas abordados en la asignatura así como en asignaturas precedentes. El trabajo de campo es una actividad que realizan los alumnos formando un único grupo de trabajo. En dicho trabajo los alumnos aplican todos los conceptos de la asignatura y de asignaturas precedentes a un sistema real. Esta actividad se lleva a cabo durante todo el cursado de la asignatura. Para realizar el desarrollo de un sistema real, los alumnos actúan como un grupo único, adoptando los distintos roles necesarios (Role-Playing) para este tipo de grupo técnico y exponiendo periódicamente los avances y dificultades encontradas. En este contexto el equipo de cátedra asume el rol de cliente, propone el problema a solucionar y paralelamente realiza la guía de las actividades desarrolladas. Así los alumnos proceden a realizar diversas tareas relacionadas fundamentalmente con la gestión del proyecto, brindando menor importancia a tareas técnicas de análisis y diseño de sistemas, continuando en cierta forma con el trabajo de campo desarrollado en la materia “Análisis y Diseño de Sistema”. Se realiza una exposición breve semanal de los progresos alcanzados y las dificultades presentadas. Esto permite realizar un seguimiento personalizado de cada uno de los integrantes del grupo.

### 3.1. Actividades del Trabajo de Campo

Al comenzar el cursado se asignan los roles adecuados al tipo de grupo de trabajo seleccionado. Como mínimo se debe contar con 9 (nueve) alumnos para poder realizar el TC, aunque se ha realizado la experiencia con una menor cantidad de alumnos, donde los mismos asumen varios roles simultáneos, siempre que no sean incompatibles [5].

- Jefe de proyecto (2)
- Analista / Diseñador (2)
- Programador Jefe (1)
- Programador (1)
- Verificador (1)
- Documentador (2)

En el caso de contar con una matrícula mayor, los alumnos restantes se incorporan a los roles ya expresados, pudiendo incrementar las funcionalidades del software a desarrollar. En este caso particular se seleccionan dos jefes de proyecto

para evitar una posible acefalía del mismo. A pesar de la asignación de roles, todos los alumnos participan en la totalidad de las etapas de desarrollo del sistema. Una vez presentada la declaración de propósito y asignados los roles comienzan las actividades propias del proyecto, de acuerdo a las técnicas seleccionadas. Una posible secuencia resumida de tareas se refleja a continuación de acuerdo a la metodología estructurada de desarrollo de software [14]:

1. Estudio de Factibilidad
2. Gestión de Proyecto
3. Análisis de Requisitos
4. Diseño
5. Pruebas
6. Código
7. Manual de Usuario
8. Gestión de Calidad

De acuerdo al problema seleccionado, se pretende centrar la atención y esfuerzos de los alumnos en los puntos 1, 2, 7 y 8. Los restantes son realizados sólo en forma parcial, o el problema a solucionar es sencillo para permitir un rápido avance en estas etapas, las que fueron desarrolladas en detalle en la asignatura “Análisis y Diseño de Sistemas”, “Programación I” y “Programación II”.

### 3.2. La organización del grupo

La primer tarea del grupo es la división de roles, donde se definen el Project Manager, analistas, diseñadores, programadores y documentadores. Esta separación se realiza principalmente pensando en los gustos personales de cada alumno, la actividad con la que se tuviera una mejor práctica y, en el caso del Project Manager, la relación que esa persona tiene con el resto de los integrantes y sus características particulares. Se cuenta con un espacio en la red Novell de la Universidad para el alojamiento de los documentos del proyecto, al cual se puede acceder en cualquier momento desde todas las computadoras de la Institución. Se dispone de un tiempo en cada clase para trabajar en el proyecto y además el grupo se reúne frecuentemente fuera de horario para realizar tareas en conjunto o controlar el trabajo individual.

### 3.3. La gestión del proyecto

Luego de definidos los roles de cada integrante, se encarar las primeras tareas concernientes a la gestión del proyecto. Cada alumno debe definir la forma de trabajar en su rol: normas, estándares, plantillas, software y metodología de trabajo. Esto se ve acompañado en las clases teóricas por una revisión de las tareas involucradas en esa etapa y la función de cada uno a lo largo de todo el proyecto. Luego se realizan presentaciones al resto de los integrantes en el horario de clase, para explicar las herramientas y formatos a utilizar. En una reunión plenaria se define grupalmente el ciclo de vida más adecuado y la planificación temporal, la cual dado que el proyecto se debe realizar y completar durante el desarrollo de la cursada, se realiza contemplando una fecha límite.

En las primeras semanas se dictan temas teóricos de planificaciones, estimaciones, configuraciones y calidad, de manera de encarar las tareas de análisis, diseño y codificación con todas las herramientas necesarias.

### 3.4. Las tareas de Captura de Requisitos, Análisis y Diseño

Para la captura de requisitos, se realizan entrevistas a los docentes, los cuales actúan como clientes. Para la confección de las entrevistas se tienen en cuenta distintas técnicas y consideraciones impartidas en clase. Ya en la etapa del análisis se sigue consultando con el cliente, e involucrándolo con el desarrollo a través de la presentación de prototipos, los cuales son de gran utilidad para comprender la visión que el cliente tiene del sistema a implementar. En el segundo año de la carrera se cursa la asignatura Análisis y Diseño de Software, y dado que se acometen muchos trabajos prácticos a lo largo del año que dura la materia, no surgen muchos inconvenientes en la realización de esas tareas.

### 3.5. La codificación

Si bien para evitar demoras innecesarias se elige un lenguaje de programación conocido para el desarrollo, anteriormente en la carrera no se integran las tareas de diseño con la programación, y llegado este punto se detectan algunas dificultades para la implementación de los requerimientos tal y como se detalla en el diseño. En lo posible, si lo permite el modelo de ciclo de vida elegido por el grupo, se incentiva la posibilidad de avanzar tempranamente en la codificación de módulos del sistema.

### 3.6. La entrega del proyecto

Para la entrega del proyecto se realiza una presentación en la cual se le permite al cliente realizar

una prueba de la version beta del software, y se le entrega la documentación realizada: manuales de instalación y de usuario.

## 4. Evaluación

La aprobación del Trabajo de Campo tiene un requisito fundamental: el proyecto debe completarse indefectiblemente en su totalidad. Esto representa una clave para la concreción del éxito de la cursada. Sin embargo también representa un posible conflicto para el grupo si existen integrantes que por diversas razones no cumplen con sus tareas asignadas. Para mitigar el factor individual, la cátedra ha implementado un sistema de penalizaciones a través de tarjetas amarillas y rojas [3, 17]. La tarjeta roja equivale a la expulsión del grupo de desarrollo y consecuentemente a la desaprobación de la actividad y por ende de la cursada. Estas tarjetas pueden aplicarse por situaciones excepcionales, ya sea por decisión de la mayoría del grupo o por asignación por parte del profesor responsable. Para evitar que los factores que influyen en el retraso del proyecto sean detectados demasiado tarde, el equipo de cátedra solicita en forma individual a los integrantes del grupo que realicen presentaciones semanales de avance de las tareas de cada rol.

## 5. Resultados

Desde el año 1998 a la fecha se ha trabajado con 10 cohortes. En los dos primeros años, donde no se aplicó el concepto de grupo de desarrollo único, los resultados fueron escasos: sólo el 50 % de los alumnos logró aprobar el cursado de la asignatura. Si bien este resultado se repite en el año 2000 (primer año en que se aplica el concepto de desarrollo único), la situación cambia radicalmente a partir del año 2001, donde se observa claramente un enorme crecimiento de las tasas de retención y aprobación. La evolución de los alumnos aprobados y desaprobados para esta asignatura puede observarse en la Fig. 1. En la Fig. 2 puede observarse una estadística similar, pero incluyendo los alumnos identificados como *ausentes*, lo cual significa que son alumnos inscriptos en el cursado pero que no han tenido movimientos académicos. (los escasos tres alumnos inscriptos en 2007 abandonaron la cursada al poco tiempo de comenzar. Esta situación se da principalmente por el cambio de plan de estudios que se lleva adelante actualmente).

Al finalizar la asignatura se solicita a los estudiantes que completen un formulario anónimo de *Valoración de la Cátedra*, donde se solicita a

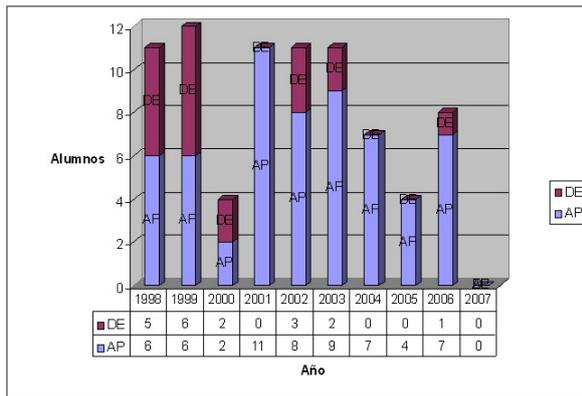


Figura 1: Estadísticas de Cursado 1998 - 2007

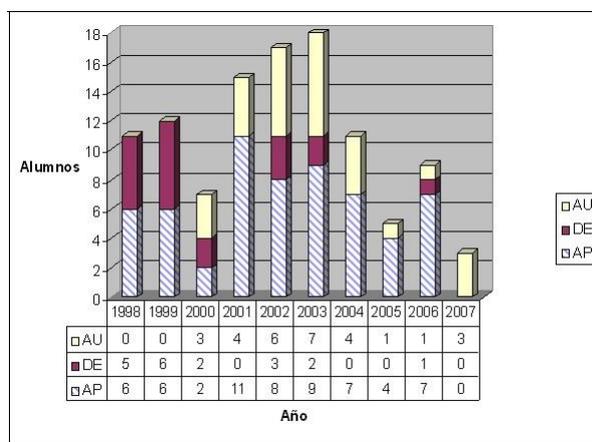


Figura 2: Estadísticas de Ausentes 1998 - 2007.

los mismos su parecer con respecto al contenido y organización del curso, material de estudio y trabajo, docentes, calificación global del curso y sugerencias. Mayoritariamente los alumnos realizan una valoración muy positiva de todos los aspectos indicados. La más baja valoración se obtiene siempre en lo referente al número de horas disponibles para realizar las actividades propuestas. Este punto ha sido la base para proponer en el nuevo plan de estudios la creación de un *Laboratorio de Desarrollo de Software*, que reemplazará a esta actividad con entidad de espacio curricular.

Para finalizar se realiza una sesión oral de evaluación de la asignatura con la totalidad de los alumnos y el equipo de cátedra, evidenciándose otros resultados importantes, que si bien corresponden a una evaluación subjetiva, son fácilmente apreciables en el trato directo con los alumnos, y corresponden a variados aspectos:

Los alumnos logran una mayor participación personal, tanto en las decisiones como en las consultas en clase, presentación de avances, etc.

Es notable la motivación observada en los alumnos, animándose unos a otros cuando se en-

cuentran con un inconveniente que puede hacer peligrar el desarrollo del proyecto, o realizando controles en forma cruzada para que esto se logre, etc.

Se genera un intenso ambiente de trabajo en grupo. Por supuesto esto siempre depende de la cohorte, pero en general luego de una primera etapa conflictiva el grupo logra un ritmo de trabajo adecuado. El principal problema observado en este aspecto es la *comunicación*, ya que el avance del proyecto requiere un intenso trabajo fuera del horario de cursada a través de reuniones de grupos, reuniones plenarias, entrevistas, comunicación electrónica, etc.

Con el avance del proyecto resulta fácil detectar las condiciones de liderazgo.

Los alumnos logran identificar en la práctica los problemas y errores más comunes en el desarrollo de software.

## 6. Trabajos Futuros

En el año 2005 se comenzó la ardua tarea de modificar el plan de estudios y la oferta académica del área. Luego de dos años se llegó a un consenso con las Unidades Académicas Caleta Olivia y Río Turbio, y a partir del año 2007 ha comenzado la implementación del nuevo plan [11, 13]. Para el diseño de los espacios curriculares del área “Ingeniería de Software, Bases de Datos y Sistemas de Información” se ha tenido en cuenta esta experiencia. Como en la misma se ha identificado la necesidad de asignar mayor carga horaria a un trabajo integrador, se ha propuesto la creación de un espacio curricular denominado “Laboratorio de Desarrollo de Software”, cuatrimestral de 6 horas en el 6° cuatrimestre. Esta asignatura comenzará a dictarse en el año 2009.

## 7. Conclusiones

Se ha presentado en este trabajo un enfoque para la enseñanza de Ingeniería de Software que según nuestro parecer ha dado excelentes resultados para brindar a los alumnos las bases necesarias para llevar adelante el Trabajo Final de Carrera de Analista de Sistemas, debido a la falencia existente para integrar los conceptos de las distintas asignaturas técnicas y de gestión del plan de estudios. La implementación de un trabajo integrador a lo largo de todo el cuatrimestre con un único grupo de trabajo donde cada alumno asume un rol específico como desarrollador ha permitido mejorar la retención, aumentar significativamente el porcentaje de alumnos aprobados y mejorar su preparación para el trabajo en equipo.

## Referencias

- [1] Using large projects in a computer science curriculum. In *SIGCSE Bulletin*, volume 32, 2000.
- [2] Alejandra Cechich, Informe de Consultoría Externa. Proyecto FOMECEC 857. Incrementar el rendimiento académico en las carreras de pregrado y grado del Informática de la UNPA, Universidad Nacional de la Patagonia Austral. 1996.
- [3] Alejandro Teruel. Penalizaciones y Bonificaciones en Equipo. Universidad Simón Bolívar, Venezuela. <http://www ldc.usb.ve/teruel/ci4712/clases2000/tarjetas.html>.
- [4] Marcela Jerez Alejandro Gimenez Guillermo Feierherd, Armando De Giusti. Análisis del enfoque de la asignatura laboratorio de software en la sede ushuaia de la unpsjb. In *VIII Ateneo de Profesores Universitarios de Computación, CACIC 2000*, 2000.
- [5] Watts Humphrey. *Managing the Software Process*. Addison Wesley Publishing Company, 1990.
- [6] Juan Manuel Luzuriaga, Informe de Consultoría Externa. Proyecto FOMECEC 857. Incrementar el rendimiento académico en las carreras de pregrado y grado del Informática de la UNPA, Universidad Nacional de la Patagonia Austral. 1996.
- [7] Osiris Sofía, Página de la Asignatura Ingeniería de Software. <http://www.unpa.edu.ar/sistemas>.
- [8] G. Gaetán P. Yañez. El laboratorio de prácticas de desarrollo de software: Un enfoque para el entrenamiento de estudiantes en ingeniería de software. In *VIII Ateneo de Profesores Universitarios de Computación, CACIC 2000*, 2000.
- [9] Proyecto FOMECEC 857. Incrementar el rendimiento académico en las carreras de pregrado y grado del Informática de la UNPA. Universidad Nacional de la Patagonia Austral. 1996.
- [10] Red de Universidades Nacionales con Carreras en Informática. Propuesta de Currícula para Carreras de Grado en Informática. 2001.
- [11] Resolución Consejo Superior, Universidad Nacional de la Patagonia Austral. Res. CO-SU N° 176/07-CS-UNPA-R, 2007.
- [12] Resolución Consejo Superior, Universidad Nacional de la Patagonia Austral. Res. CO-SU N° 116/00-CS-UNPA, 2000.
- [13] Resolución Consejo Superior, Universidad Nacional de la Patagonia Austral. Res. CO-SU N° 147/95-CS-UFPA, 1995.
- [14] Ian Sommerville. *Ingeniería del Software 7ma Edición*. Pearson Addison Wesley Publishing Company, 2006.
- [15] Janine LaFrance Tyson R. Henry. Integrating role-play into software engineering courses. *Journal of Computing Sciences in Colleges*, 22(2):32–38, 2006.
- [16] Yellow Card / Red Card Mechanism. Effective Projectwork in Computer Science: EPCoS. <http://www.cs.kent.ac.uk/national/EPCoS/>.