

# TOLUCA

## La Programación de Juegos para la Enseñanza del Paradigma Orientado a Objetos

P. Santa Cruz, A. Alliana, A. Feltes, L. Cernuzzi

Universidad Católica “Nuestra Señora de la Asunción”, Paraguay  
lcernuzz@uca.edu.py

### ABSTRACT

The paper presents the experience of teaching Object Oriented Programming, based on Java, by mean of the TOLUCA project. The project aimed to develop an on-line version of the well known card game Truco, usually played in different countries of South America. The project has followed a collaborative approach and was developed as part of a selective course in which participated 20 students from different disciplines and with different knowledge levels. In general, the experience was very positive both from the perspective of the obtained product, both from the perspective of the learning process.

**Keywords:** Object Oriented Programming, Java, Games, Collaborative Development, Teaching methodologies

### RESUMEN

El trabajo presenta la experiencia de enseñanza de conceptos de Programación Orientada a Objetos basada en el lenguaje Java a través del proyecto TOLUCA. El proyecto ha permitido el desarrollo de una versión on-line del juego de Truco, un juego conocido y jugado en varios países de América del Sur. El proyecto se ha caracterizado por un enfoque colaborativo y ha sido realizado en el marco de una materia optativa en la que han participado 20 alumnos de diferentes niveles y carreras. En términos generales la experiencia puede ser clasificada muy positivamente tanto desde la perspectiva del producto obtenido como, sobretodo, desde la perspectiva del proceso de aprendizaje.

**Palabras Clave:** Programación Orientada a Objetos, Java, Juegos, Desarrollo Colaborativo, Metodologías de Enseñanza.

### INTRODUCCIÓN Y MOTIVACIÓN

Los juegos han jugado siempre un papel importante en la enseñanza de conceptos básicos de lenguajes de programación en general [1].

Una de las razones principales por la que esto sucede es la cuota de motivación extra que el diseño y desarrollo de un juego tiene para el alumno. Otra de las razones es el nivel de abstracción que se debe poseer para modelar correctamente con los elementos que hay en los distintos lenguajes, tanto las reglas de juego, como los algoritmos que permiten que un computador pueda participar en el juego.

Varios juegos han sido utilizados para la enseñanza de la orientación a objetos. Uno de ellos es un proyecto de IBM que enseña lenguaje Java y sus conceptos de orientación a objetos en una manera divertida a través de la creación de robots que batallan en una arena unos contra otros [2].

El “Truco” es un juego originario de España pero que se juega y es extremadamente popular principalmente en la región del Río de la Plata: Paraguay, Uruguay y Argentina. Está basado en la baraja Española y puede ser jugado en grupos de 2, 4 ó 6 jugadores enfrentados siempre en dos equipos.

Jugar bien al Truco supone un conocimiento profundo de sus reglas básicas, de estadísticas y teoría de juegos que permita siempre maximizar la cantidad de puntos que se puedan obtener en una ronda. Además de esto, como se trata de un juego basado en retos, también hay una cuota importante de conocer o estudiar al contrincante para tener una mayor posibilidad de suceso. Por todo esto, es un juego ideal para fomentar las destrezas de los jugadores en estas áreas mencionadas.

Desarrollar una versión de este juego que pueda ser jugada en línea por jugadores conectados a través de una red utilizando protocolo TCP/IP (posiblemente Internet) presentó fuertes desafíos en diversas áreas de tecnologías de la información como ser: redes, bases de dato, interfaces de usuario y arquitectura cliente/servidor. Esto dio la posibilidad de que los alumnos lidiasen con los problemas que acompañan la integración de todas estas tecnologías; y la orientación a objetos brindó las respuestas necesarias para hacer un trabajo ordenado y documentado al nivel necesario para que el proyecto sea manejado con suceso en un grupo de considerable tamaño en cantidad de personas.

La idea de este tipo de juegos fue inspirada en el sitio de juegos de una de las compañías líderes en portales de Internet, YAHOO Inc. [3]

Tanto en el análisis y diseño documentados con UML [4], como en la programación, la orientación a objetos demostró ser capaz de modelar, con un nivel de abstracción adecuado, todos los componentes de un juego de cartas en línea según el modelo cliente/servidor. En efecto, existe una correspondencia casi directa de la mayoría de los elementos del juego con Clases, por ejemplo: una mesa, una carta, una

sala de juegos, lo que hace que inicialmente se requiera de un nivel de abstracción abordable por parte de los alumnos.

El resto del presente trabajo se estructura como sigue. En la sección 2 se presenta brevemente el juego del Truco con sus reglas fundamentales. En la Sección 3 se describe la arquitectura del sistema así como la plataforma de desarrollo y se presentan algunas consideraciones con respecto a su relación con la orientación a objetos y el lenguaje Java en particular. A continuación, en la sección 4, se presentan diversas consideraciones acerca del proyecto en término del trabajo grupal, de la aplicación desarrollada, del proceso de aprendizaje y del enfoque del proyecto comparándolo con el de otras materias relacionadas. Finalmente, en la última sección se presentan algunas conclusiones y se trazan posibles trabajos futuros.

## EL JUEGO DEL TRUCO

El Truco se juega con un mazo completo de 40 cartas españolas y en el que pueden participar dos equipos de uno, dos o tres jugadores cada uno. El Diccionario de la Real Academia Española postula que el Truco es una variedad del Truque, juego popular en algunas regiones de España.

El objetivo del juego es sumar puntos. Normalmente la partida se juega a 30 puntos divididos en dos partes iguales, la “mala” y la “buena”. El Truco es un juego que involucra suerte, habilidad, experiencia y capacidad de mentira.

Para jugar al Truco existen varias modalidades: la individual o por parejas, siendo esta última modalidad la más usual y la única válida en campeonato. En el juego por parejas, se rota la condición de líder del equipo: éste coordinará el desarrollo del juego de la pareja, mediante contraseñas, generalmente verbales y preestablecidas entre ambos.

El transcurso del juego se desarrolla en "manos" La mano comienza cada vez que se dan tres cartas para cada jugador en sentido contrario a las agujas del reloj (Figura 6). Es en esta fase del juego cuando el capitán de cada equipo recibe por señas, antes de comenzar, las cartas de su compañero, para luego indicarle qué carta debe utilizar o que debe decir durante el desarrollo del juego. Estas direcciones se realizan normalmente utilizando una jerga particular al juego (cántale, rétale, etc.).



Figura 6. Juego de Truco

El juego consta principalmente de tres partes o juegos: el envío y la flor, que pueden o no disputarse, y el Truco, que

se juega una vez terminado los dos primeros. El juego de la flor anula el juego del envío.

## Reglas del Truco

### Envío

El envío se disputa antes de que se resuelva la primera baza de Truco de cada mano. El jugador que envía debe hacerlo antes de jugar su primera carta. El envío puede incluso no disputarse durante una partida, ya que los jugadores no están obligados a enviar, si consideran que no llevan suficientes cartas de envío para hacerlo con éxito o simplemente porque quieren esconder sus cartas para el Truco. Por tanto, si llega a disputarse la primera baza sin que nadie haya enviado, al final de esta mano no se otorgarán puntos por el envío.

La apuesta al ganador que se acumula al pronunciar la palabra ‘envío’ son 2 puntos. Ningún jugador puede enviar cuando antes ya lo haya hecho otro: tendrá que limitarse a responderle. Las respuestas posibles son:

- No querer, en cuyo caso el punto puesto en juego se lo anotará el jugador que envió;
- Querer el envite, en cuyo caso se otorgarán 2 puntos al que mayor puntuación de envío tenga.
- No sólo querer, sino aumentar la apuesta. En este caso, el turno de hablar vuelve al jugador que envió.

Las reglas dependen de la región donde se realice el juego..

### La flor

La flor (3 cartas del mismo palo) es un componente muy importante del juego, ya que introduce una fuerte variable aleatoria. Un equipo agraciado por las cartas en forma de abundantes flores, puede llegar a ganar el partido, sin desplegar un juego excesivamente brillante. Por cada flor, el equipo poseedor de la misma se anota tres puntos.

De poseer una, cada jugador debe “cantar” flor antes de jugar su primera carta. A diferencia del envío, en la flor si dos jugadores del mismo equipo cantan flor los puntos obtenidos se acumulan.

### Truco

Es la suerte definitiva del juego. En él triunfa aquel jugador que hace 2 de las 3 bazas de la mano. Cabe destacar que el valor de las cartas en el Truco no es igual a su valor en el envío. Para ganar una baza ha de jugarse una carta de mayor valor que la del adversario. Esto es así, aunque no siempre es necesario tener mejores cartas que el adversario para ganar el Truco: en algunas ocasiones el Truco se gana haciendo creer al adversario que se posee cartas mucho mejores que las que realmente se posee. Aquí es donde radica la dificultad del juego, que se aleja bastante de la tarea mecánica que sería mostrar las cartas en cada una de las tres jugadas de la mano y ver quien ha triunfado en dos de estas tres jugadas.

Al igual que en el envío, en el Truco existen una serie de “retos” para intentar ganar más puntos o intentar persuadir al

rival de retirarse y perder los puntos en juego. Una descripción completa de las reglas se encuentra en [5].

### TOLUCA: TRUCO ON-LINE DE LA UCA

El curso de programación Orientada a Objetos nació como una idea para fijar los conceptos de Orientación a Objetos en una aplicación concreta, y poder hacer uso de una de las principales ventajas de la OO, que es aplicarla en las etapas de Análisis y Diseño, además de la de implementación en un lenguaje de programación.

El curso se realizó como una materia optativa de las carreras de Ingeniería Electrónica e Informática de la Universidad Católica “Nuestra Señora de la Asunción” (Paraguay), teniendo como principal propósito el desarrollo en el lenguaje Java. La carga horaria del mismo ha sido de 6 horas semanales durante dos semestres.

Se inició con algo más de 20 alumnos, con el requisito mínimo de haber aprobado alguna materia previa referente a Introducción a los Lenguajes de Programación. Los alumnos participantes en el curso han sido divididos en diferentes grupos de trabajo permitiendo así aplicar el principio de modularidad y separación de los conceptos para facilitar la posibilidad de alcanzar resultados significativos en términos del sistema global.

### Plataforma de Desarrollo

La naturaleza del proyecto hace que se tenga que contar con distintos tipos de herramientas y software de base. En efecto, más allá del lenguaje de programación, sería inadecuado pensar en el desarrollo y la implementación de un proyecto colaborativo y de una cierta complejidad intrínseca, como el TOLUCA, sin involucrar herramientas de modelado y diseño, así como de entorno de desarrollo integrado y de administración de desarrollo grupal entre otras. Además, siendo que un requisito de la aplicación es que corra en Internet, ha sido necesario contemplar un servidor Web y plataformas anexas. En el mercado existen varias alternativas de herramientas específicas y varias de estas han sido analizadas previa a la elección de la siguiente plataforma de desarrollo:

- Lenguaje de Programación: JAVA 2 Software Development Kit – 1.4.2\_x [6]
- Entorno de desarrollo Integrado: Eclipse 2.x y 3.x [7]
- Base de Datos: Firebird [8]
- Manejo de procesos de desarrollo en grupo: CVS (Concurrent Versions System) [9]
- Herramienta de Diseño y Modelado UML: Poseidón CE [10]
- Contenedor de Servlets: Apache Tomcat [11]
- Servidor WEB para la aplicación: Apache Web Server [12]

Todas las herramientas comparten diferentes características, que jugaron un papel muy importante en su elección. Una primera característica es la portabilidad. El equipo de

desarrollo utilizaba indistintamente plataformas basadas en MS-Windows ó basadas en Linux para el desarrollo.

Además, todas las herramientas escogidas son de Código Abierto. Con alguna diferencia en el tipo de licencia con que se otorga, todas las herramientas escogidas ponen a disposición de los desarrolladores el código fuente de las mismas, lo cual facilita el aprendizaje y la re-implementación de funcionalidades, así como el rastreo de bugs de los sistemas.

Finalmente, las herramientas son de libre distribución. También con ciertas diferencias en las licencias, todos estos programas escogidos permiten ser utilizados sin el pago de costosas licencias por los mismos.

### El Diseño del TOLUCA

El diseño fue iniciado por los profesores de la materia, quienes hicieron el modelado inicial de Clases y las relaciones entre las mismas. Dicho diseño ha sido distribuido a los diferentes grupos de trabajo.

Se eligió el lenguaje UML [4], por la facilidad que el mismo provee para este tipo de tareas, y para la comunicación entre los grupos de desarrollo.

### Arquitectura General del Software

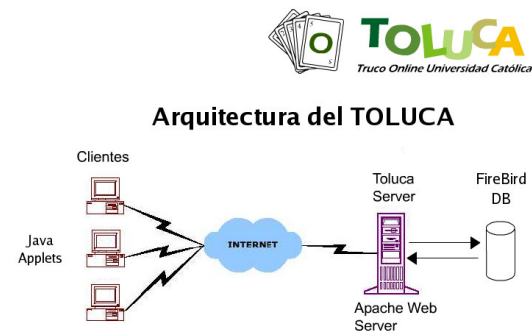


Figura 7. Arquitectura del TOLUCA

En la Figura 7 se aprecian la dos partes principales del TOLUCA: el cliente y el servidor.

El cliente consiste de un Applet Java que es descargado por el navegador de los clientes que ingresan al sitio WEB. Mientras que el servidor es una aplicación Java que está escuchando en un puerto TCP que tiene una conexión a una Base de Datos (Firebird) mediante JDBC, en la cual se encuentra el registro de usuarios y los registros y resultados de los partidos desarrollados.

La comunicación entre el cliente y el servidor se lleva a cabo mediante un par de clases Java que utilizan el protocolo TCP/IP para comunicarse entre sí, lo cual permite su utilización en Internet.

Como parte de la arquitectura interna (ver Figura 3), tanto el cliente como el servidor disponen de tres clases que abstraen el juego y se comunican entre ellas y a su vez con el servidor como lo indica la siguiente figura:

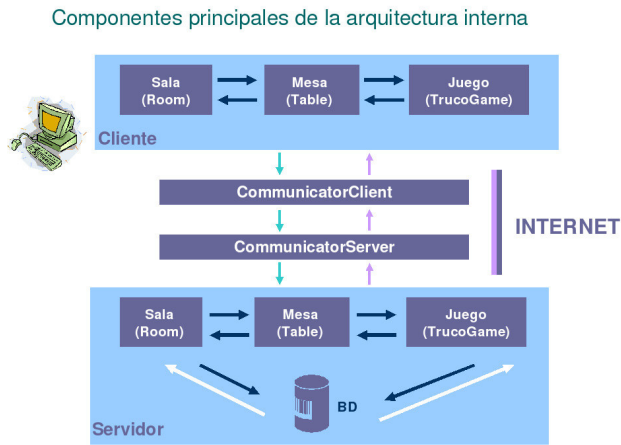


Figura 8. Arquitectura interna

La clase Room (sala) abstrae el concepto de una sala, en donde los jugadores ingresan y se encuentran con otros conectados al mismo tiempo. Dentro del Room un jugador puede elegir crear un Table (mesa) y a su vez, dentro de un Table, puede elegir crear un TrucoGame (juego de Truco). La comunicación se da entre las clases relacionadas (Figura 9). Se creó para eso una jerarquía para cada una de ellas, así tenemos que de la clase Room descenden las respectivas clases RoomServer y RoomClient, y lo mismo para Table y TrucoGame.

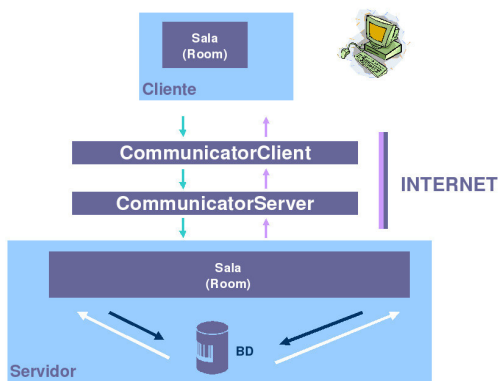


Figura 9. Comunicación entre componentes

#### Protocolos de Comunicación entre los Componentes

Para la comunicación interna entre los componentes se escogió el modelo de eventos de Java (Figura 10. Modelo de eventos), uno de los componentes principales del lenguaje. Inclusive, existen autores y programadores que suelen decir que Java es un lenguaje orientado a Eventos, por el uso intenso que tienen los mismos en gran parte de la API del lenguaje.

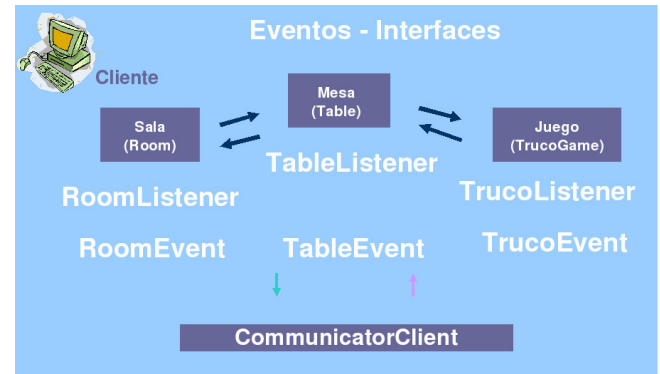


Figura 10. Modelo de eventos

La comunicación remota entre los componentes (ver Figura 6) es manejada por la clase Communicator, y sus respectivas extensiones CommunicatorClient y CommunicatorServer.

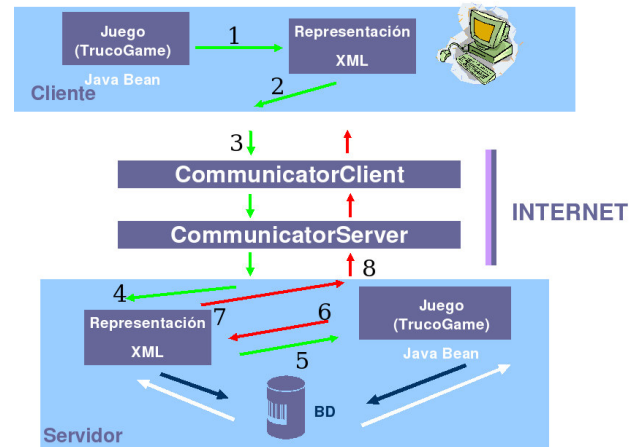


Figura 11. Esquema de comunicadores

Éstas se encargan de hacer todo el paso de mensajes entre el lado cliente y el lado servidor del Toluca, por medio de un protocolo basado en XML. Cada clase a ser transportada fue modelada como un Java Bean, siguiendo las especificaciones de Sun Microsystems para los mismos [13]. Cada Bean a ser transportado genera una representación en XML del mismo, y la misma se pasa a la clase Communicator para que lo envíe a través de la red.

#### La Orientación a Objetos y el Lenguaje Java en el Proceso de Desarrollo

La orientación objetos jugó un papel fundamental en el proceso de desarrollo. En primer lugar apoyó a la fase de análisis y diseño. Aquí se pudieron abstraer todos los componentes visibles y obvios del juego como los jugadores, las mesas, las cartas, las manos, las jugadas, así como los componentes no tan directos como el servidor y el administrador de reglas, entre otros.

El *UML* (*Unified Modelling Language*) fue clave en la documentación de todos los componentes y apoyó notoriamente la comunicación intergrupar necesaria para la integración exitosa de todo el sistema. Dentro de la gama de diagramas presentes en *UML* fueron principalmente útiles en este proceso de desarrollo los diagramas de clase, diagramas de objetos y diagramas de secuencia.

El lenguaje Java también fue fundamental. Su sencillez, robustez y buena documentación sirvieron para poder llevar el proyecto a un término en donde los objetivos fueron plenamente cumplidos. La riqueza y claridad del *API de Java* bajó mucho los tiempos de desarrollo porque tiene componentes altamente probados en materias de programación de redes, manejo de *XML* y desarrollo de interfaces gráficas (*Java Swing*) que fueron parte integral del juego.

El proyecto final cuenta con aproximadamente 140 clases y 30000 líneas de código de lenguaje Java.

### **ANÁLISIS DE LA EXPERIENCIA DEL PROYECTO COLABORATIVO**

Enseñar orientación a objetos a través de la producción de un juego desarrollado entre todos los integrantes del curso como trabajo final, fue una experiencia muy productiva para todos los actores involucrados en el proyecto: universidad, alumnos y profesores.

El curso tuvo un doble enfoque teórico-práctico, muy recomendable debido a que se necesitan ambas cosas para tener una formación integral en el área.

El objetivo final del curso en términos de “productos” fue presentar a la Facultad de Ciencias y Tecnología, un sistema que permita a usuarios conectados en línea a través de una red TCP/IP (podría ser Internet) jugar al mencionado juego de naipes “Truco”, entre dos, cuatro o seis jugadores.

Lograr este objetivo final requirió de mucho esfuerzo de parte de todos.

#### **Metodología de enseñanza**

El curso tuvo una duración de un semestre con una carga horaria de cinco horas semanales. Los alumnos tenían como requerimiento haber pasado un curso de introducción a la programación.

Los alumnos fueron evaluados en aspectos teóricos mediante pruebas parciales y en aspectos prácticos, mediante trabajos prácticos. Queremos destacar uno de ellos, el Robocode [15]. El Robocode es un proyecto original de IBM cuyo objetivo es la enseñanza del lenguaje java. En este trabajo práctico se requirió a cada alumno crear un robot que compitió contra los robots de otros alumnos en una batalla virtual en modalidad de torneo puntuable que fue parte de la evaluación final de cada uno.

Además del curso de programación, se dio una introducción al análisis y diseño orientado a objetos con UML. Esto permitió introducir al proceso de aprendizaje exitosamente herramientas de apoyo a las distintas fases de la ingeniería de sistemas. El análisis y diseño del proyecto global fue hecho por los profesores, mientras que la definición en detalle de los componentes del sistema fue hecha por los distintos grupos de trabajo formados por alumnos y profesores.

Una vez que los alumnos decidieron el grupo de trabajo de su preferencia, se brindaron clases especializadas para que cada grupo tenga una mejor orientación y especialización en cada componente. Esto fue particularmente interesante, ya que permitió a que cada grupo se enfoque en el área particular del problema a ser atacado.

#### **Trabajo Grupal**

El primer gran desafío fue manejar un grupo considerablemente heterogéneo en cuanto a nivel de formación, con poco conocimiento básico de orientación a objetos en general.

El curso estaba formado por alumnos que cursaban desde el 2 hasta el 5 año de distintas carreras (Ingeniería Electrónica, Ingeniería Informática y Análisis de Sistemas) en la Universidad Católica “Nuestra Señora de la Asunción”.

Como ya mencionamos, el conjunto de alumnos del curso fueron divididos en diferentes grupos cuya conformación y objetivo principal se describen brevemente a continuación.

#### **Grupo de Interfaz de Usuario**

Se encargó principalmente del desarrollo del cliente del juego, es decir de la parte visible que es utilizada a través de un *Applet* de *Java* corriendo en un navegador que tenga soporte para esta tecnología. Este grupo estuvo integrado inicialmente por 8 integrantes del curso, que dividieron las tareas en la realización de: interfaz de inicio o preparación del juego, interfaz de juego, y panel de *chat*.

#### **Grupo de Servidor**

Se encargó del desarrollo del servidor del juego. El servidor se encarga de representar todos los componentes de manera centralizada para poder coordinar la comunicación entre los clientes conectados remotamente. En el grupo han participado inicialmente 3 integrantes.

#### **Grupo de Redes**

El grupo se hizo cargo del intercambio de datos e información entre los clientes (interfaz de usuario) y el servidor con comunicación a través de sockets *TCP/IP*. Los paquetes de intercambio fueron codificados en *XML* (*eXtensible Markup Language*) para simplificar y documentar de una manera clara y sencilla el protocolo de comunicación, y además para poder aprovechar el excelente soporte de herramientas de análisis de *XML* que vienen incorporadas en el mismo *API* del lenguaje *Java*. El grupo contó con 5 integrantes.

#### **Grupo de Base de Datos**

La información persistente del juego se encargó al grupo de Base de Datos. Fue integrado por un único alumno ya que la tarea de desarrollo e implementación de este grupo se limitaba a la autenticación dentro del sistema para la parte de seguridad (manejo de usuarios y contraseñas), así como la implementación de un sistema de bitácora para los partidos y mantenimiento de un *ranking* de los mejores jugadores del sistema. Para esta última parte, se utilizó la misma fórmula adoptada por la Federación Internacional de Ajedrez para mantener los *Elos* de los jugadores afederados. Los *Elos* son puntajes que determinan una posición general a cada jugador, y se calculan en base a los *Elos* actuales de los participantes de las partidas.

#### **Grupo de Reglas de Juego**

Este grupo de 3 integrantes, se encargó de proveer los componentes del sistema a cargo de manejar las reglas del juego de Truco. Estas reglas sirvieron para manejar la secuencia y los turnos del juego, así como para validar la validez de las jugadas realizadas en el sistema y contabilizar los puntos de cada uno de los equipos participantes en el juego.

Luego de un curso de aproximadamente seis meses donde se trataron temas relacionados a orientación a objetos, usando como herramienta principal de enseñanza al lenguaje Java, se procedió a formar un grupo de trabajo para implementar el juego como proyecto final de curso.

Este proyecto requirió de mucho esfuerzo de los alumnos y se llevó a cabo con la participación de la mayoría de los participantes de la formación previa de seis meses. Algunos alumnos, sin embargo, quedaron auto excluidos del proyecto debido a la inversión de tiempo necesaria para poder estar en sintonía con el resto de los alumnos y por ende poder realizar un efectivo trabajo responsable en equipo.

En general los alumnos que participaron llenaron las expectativas de los profesores, colaborando tanto en el desarrollo del juego como en la integración de los componentes y posteriores pruebas exhaustivas. Esta combinación permitió que el trabajo pudiera ser concluido en el tiempo establecido y con los parámetros de calidad propuestos en los documentos de definición.

Además de los alumnos propios del curso, se obtuvo colaboración en cierta medida de parte de alumnos y profesores de la carrera de Diseño Gráfico, quienes hicieron aportes en la línea de interfaces, dibujos, presentación e interacción en general.

El trabajo en grupo cercano con otros compañeros del mismo nivel de formación, con otros compañeros de distinto nivel de formación (por encontrarse en cursos superiores o inferiores) y con tres profesores, supuso una experiencia en donde hubo una simbiosis en la que todos aprendieron de la experiencia de los demás y pudieron entender las ventajas y

complicaciones que tiene trabajar en grupos de desarrollo de tamaños considerables, teniendo en cuenta que en las demás materias, los trabajos son individuales o en grupos pequeños de trabajo.

#### **El sistema resultante**

El resultado final del primer curso está disponible al público en general en el sitio oficial del proyecto Toluca: [www.truco.com.py](http://www.truco.com.py). En este periodo de funcionamiento se pudo comprobar que el sistema es bastante robusto y estable, y el servidor ha sido probado con varias conexiones en simultáneo sosteniendo varios juegos al mismo tiempo. Evidentemente, la posibilidad de contar con la visibilidad de una aplicación pública en Internet ha sido y sigue siendo un factor de grande motivación para todos los participantes en el proyecto, empujando así un consistente esfuerzo en pos de una calidad del juego. Inclusive, esto se vio reflejado en el diseño de las interfaces y las formas de interacción para lo que se ha contado con el apoyo de un diseñador gráfico.

Al finalizar el proyecto, y como parte del proceso de evaluación, se realizó una presentación interna en la Universidad donde el trabajo tuvo una excelente aceptación por la comunidad educativa compuesta por profesores, directivos y alumnos que no habían participado en el proyecto. También se hicieron varias presentaciones en otras Universidades y congresos nacionales, donde el trabajo presentado causó una muy buena impresión en todo sentido.

#### **Consideraciones acerca del Proceso de Aprendizaje**

El proceso de aprendizaje ha sido muy importante en la formación de los alumnos. Han adquirido no solamente conocimientos y experiencia en análisis, diseño y programación orientada a objetos sino que también formaron parte de un grupo de desarrollo relativamente complejo, donde tuvieron que lidiar con problemas como la integración y prueba de funcionamiento correcto de componentes, creación de marcos de prueba independientes, integración de código y documentación grupal.

En este sentido, la Orientación a Objetos demostró plenamente su razón de ser y por qué es uno de los paradigmas predilectos hoy día para desarrollo de sistemas y diseño de nuevos lenguajes de programación. La integración con relativa facilidad gracias a las interfaces bien definidas de comunicación entre objetos, el bajo acoplamiento, las herramientas de documentación y la reutilización de códigos y componentes, pusieron en evidencia las enormes ventajas que este paradigma presenta al ser comparado con otros. En este punto, el lenguaje Java con las ventajas ya mencionadas, fue una excelente elección para poner en práctica todos estos nuevos conceptos aprendidos.

Los alumnos además se encontraron con el desafío extra de unir en un solo sistema los conocimientos adquiridos en otras materias curriculares de la carrera como Redes, Sistemas Operativos, Bases de Datos, Ingeniería de Software, Lenguajes de Programación, entre otras. Esto por sí solo



resultó muy motivador puesto que para muchos fue la primera experiencia en la que pudieron darse cuenta de los relacionadas que están todas las áreas de estudio de la ciencia informática y en qué podría aplicarse en un sistema real cada una de ellas.

Sin duda, las distintas tareas y el trabajo asumido han sido llevados adelante con responsabilidad por parte de los alumnos involucrados. Seguramente en este aspecto ha influenciado notablemente el fuerte grado de motivación del grupo. En la motivación han influido varios factores, algunos de los cuales han sido previamente mencionados, como ser el desafío de construir una aplicación entretenida y “visible”, la fuerte cohesión del grupo, la posibilidad de aprender nuevos conceptos y destrezas en un ambiente de cooperación propicio, la posibilidad de ver los nexos y relaciones entre varias disciplinas de la carrera, y la oportunidad de demostrar el potencial profesional de nuestros alumnos y egresados. En este último aspecto vale la pena mencionar que los tres profesores involucrados en el proyecto son egresados de la carrera de Ingeniería Informática de nuestra Universidad. Todo esto permitió que los alumnos que asumieron el compromiso del proyecto final lo llevaran adelante sin abandonos. Más aún, un aspecto que se podría destacar es como, en términos generales, la interacción y ayuda entre pares ha facilitado el proceso de aprendizaje. También en otras materias normalmente acontece algo parecido, sin embargo, en el caso de un proyecto con un objetivo común y en el que el trabajo de cada uno es indispensable para alcanzarlo, marca una diferencia con respecto a los procesos de estudio de las demás materias.

Para los profesores también todo el proceso ha sido muy enriquecedor, habiendo tenido la oportunidad de probar teorías de aprendizaje en base al desarrollo de juegos y coordinar las actividades de los distintos grupos de trabajo involucrados.

#### **Comparación con Otras Experiencias en las Carreras**

El proceso normal en otras materias dictadas en la misma Universidad, se basa normalmente en una evaluación final con un trabajo único que tiene que ser entregado de manera individual por cada alumno o bien en grupos reducidos de alumnos (2 o 3 integrantes). La única otra materia que contempla parcialmente modalidades similares es Ingeniería de Software, utilizada como materia integradora en una etapa intermedio-alta de la carrera de Ingeniería Informática.

En particular se ha marcado una fuerte diferencia con respecto a las demás materias relacionadas con la programación, donde el enfoque se centra en el desarrollo de uno o varios mini-proyectos por parte de un grupo muy reducido de alumnos. Inclusive, en algunas de dichas materias, para evaluar la calidad del trabajo se adoptan enfoque competitivos, es decir, se ponen a competir los software desarrollados por cada grupo otorgando una mejor puntuación a los que ganan el juego.

En este caso, el proceso de desarrollo de un juego completo con capacidad de juego en línea, fue único y desarrollado por todos los participantes del curso en forma colaborativa.

Esto brindó la experiencia de trabajo grupal anteriormente mencionada y un trabajo que pudo ser publicado en redes públicas para la utilización de usuarios en general. Además, el tipo de proyecto sobrepasa ampliamente los límites del contenido principal: la orientación a objetos a través del lenguaje de programación Java. Más bien, se constituye como una experiencia integradora de conocimientos de diversas disciplinas y materias.

#### **CONCLUSIONES Y TRABAJOS FUTUROS**

En este artículo explicamos las razones por las cuales la programación de juegos en general es un problema adecuado para la enseñanza de Orientación a Objetos. Además hemos presentado el juego del Truco en particular, y nuestra experiencia en la enseñanza del lenguaje de programación Java para la programación de un proyecto de envergadura considerable por un grupo grande de alumnos.

Los juegos son adecuados para la enseñanza del paradigma Orientado a Objetos, ya que existe una correspondencia casi directa de la mayoría de los elementos del juego con Clases, lo que hace que inicialmente se requiera de un nivel de abstracción abordable por parte de los alumnos.

Otro factor relevante es la motivación. Es más fácil mantener a los alumnos motivados en los proyectos finales de curso, utilizando problemas didácticos. Los juegos ayudan a resolver este problema, ya que por su naturaleza, los alumnos hacen la verificación de la calidad de la aplicación por diversión.

Relacionado con la motivación está el enfoque colaborativo del proyecto. Esto ha permitido el fortalecimiento de un espíritu de grupo y al mismo tiempo ha facilitado el desarrollo de destrezas y aptitudes comunicativas y de colaboración que son destrezas transversales muy apreciables en cualquier profesional.

Una ventaja adicional de proyectos como el TOLUCA es que los juegos pueden ser utilizados en varios niveles de la enseñanza. Particularmente, en nuestra experiencia, las tareas de programación de las reglas del juego, guardar el estado de juego y recuperarlo y la programación de la interfaz gráfica son tareas que pueden ser realizadas por alumnos de materias introductorias a la programación. Alumnos de cursos avanzados de Inteligencia Artificial pueden realizar proyectos sobre la programación de los algoritmos de juego y aprendizaje sobre ellos.

El área de la enseñanza del paradigma orientado a objetos mediante la programación de Juegos seguirá siendo materia de exploración. Actualmente se están llevando a cabo dos investigaciones nuevas en esta línea. La continuación de desarrollos de juegos en línea con la inclusión del juego de “Escoba de 15” en el proyecto TOLUCA y el desarrollo de un

framework de plugins orientados a objeto con C++ para la implementación del juego Mastergoal. Mastergoal ha sido creado por una persona que reside en Paraguay y apunta al desarrollo de habilidades de inteligencia espacial, de estrategias, de lógica y otras destrezas mediante un juego de tablero basado en conceptos del fútbol ([www.mastergoal.com](http://www.mastergoal.com)).

#### **AGRADECIMIENTOS**

A todos los alumnos que participaron en las materias.

#### **REFERENCIAS**

1. Huang, T. Strategy Games Programming Projects. Department of Mathematics and Computer Science. Middlebury College. Middlebury. Mayo de 2001.
2. <http://www.alphaworks.ibm.com/tech/robocode>
3. <http://games.yahoo.com>
4. Grady Booch, Ivar Jacobson, James Rumbaugh. Unified Modeling Language User Guide (Object Technology) 1998. Addison Wesley.
5. Fuentes, J. El Truco: Historia de una Tradición. SUMUNTÁN, Revista de Estudios de Sierra Mágina, n.º 14. (2001).
6. <http://java.sun.com>
7. <http://www.eclipse.org>
8. <http://www.firebird.sourceforge.net>
9. <https://www.cvshome.org/>
10. <http://www.gentleware.com/>
11. <http://jakarta.apache.org/tomcat>
12. <http://www.apache.org>
13. <http://java.sun.com/products/javabeans>
14. <http://www.apache.org>
15. <http://robocode.sourceforge.net/>