

## Competencias en Programación Orientada a Objetos

Lucas Spigariol<sup>1,3,4</sup>, Nahuel Palumbo<sup>1,2,3</sup>, y Nicolás Passerini<sup>1,2</sup>

<sup>1</sup> Universidad Nacional de San Martín, San Martín, Provincia de Buenos Aires, Argentina

<sup>2</sup> Universidad Nacional de Quilmes, Bernal, Buenos Aires, Argentina

<sup>3</sup> Universidad Tecnológica Nacional - FRBA, CABA, Buenos Aires, Argentina

<sup>4</sup> Universidad Tecnológica Nacional - FRD, Campana, Buenos Aires, Argentina  
{lspigariol, nahuel.palumbo, npasserini}@gmail.com

**Resumen.** La tendencia actual de encarar los desarrollos curriculares desde una perspectiva de competencias constituye una valiosa oportunidad de repensar prácticas docentes. En ese sentido, se presenta un estudio de caso que consiste en la sistematización de una práctica docente. Es una reflexión sobre el proceso de llevar adelante un trabajo por proyectos en el marco de asignaturas de programación, en particular de programación orientada a objetos, a partir de una revisión crítica de la planificación anterior desde una mirada basada en competencias. La propuesta central fue un trabajo en equipo con consigna abierta, que consistió en el desarrollo de un videojuego, utilizando la herramienta Wollok Game, que es una biblioteca para facilitar el desarrollo de interfaces gráficas interactivas que forma parte del software educativo Wollok<sup>1</sup>. La experiencia se llevó a cabo en cuatro asignaturas de diferentes universidades públicas que comparten objetivos, materiales, herramientas y enfoques pedagógicos.

**Palabras clave:** Enseñanza de la programación, Competencias, Programación Orientada a Objetos, videojuego, Wollok.

## Pedagogical competences in Object Oriented Programming

**Abstract.** The current tendency of facing curricular developments from a pedagogical competence perspective constitutes a valuable opportunity for rethinking educational activities. Taking this into account, we present a case study about one teaching practice systematization. It is a reflection about the process of carrying out project based works in the programming courses frame, particularly Object Oriented Programming, based on a critical review of the previous planification from a pedagogical competence point of view. The main proposal was the elaboration of an integration project that was held along the whole course. This integration project consisted in coding a videogame using the Wollok Game tool, which is a library that allows the interactive graphic interfaces development in Wollok, an educational software. This experience took

---

<sup>1</sup> <https://www.wollok.org/>

place in three public Argentinian universities, during eight courses which shared objectives, resources, tools and pedagogical perspectives.

**Keywords:** Teaching programming, Competencies, Object Oriented Programming, videogame, WolloK

## 1 Introducción

Pensar los itinerarios educativos desde una perspectiva de competencias presenta un desafío para revisar las prácticas concretas.

Aún reconociendo la variedad de interpretaciones teóricas y sobre todo implementaciones concretas en planificaciones y planes de estudios que apelan a esta categoría pedagógica, y evitando caer en posiciones extremas que van desde pensar que es una revolución que viene a refundar el sistema educativo o que es una moda pasajera que consiste en maquillar las formalidades sin alterar las prácticas, el desafío abordado consiste en analizar un caso concreto.

Como afirma Coll, “tal vez el riesgo principal del enfoque basado en competencias sea similar al que han tenido que afrontar en el pasado otros enfoques, con éxito casi siempre más bien escaso o moderado: el de presentarse y ser presentado como una solución a los males, problemas e incertidumbres que aquejan la educación escolar en la actualidad. Las aportaciones de los enfoques basados en competencias son muy valiosas, pero definitivamente tampoco son un remedio milagroso.” [1]

En primer lugar, siguiendo a Tobón, se entiende a las competencias como “Procesos complejos de desempeño con idoneidad en determinados contextos, integrando diferentes saberes (saber ser, saber hacer, saber conocer y saber convivir), para realizar actividades y/o resolver problemas con sentido de reto, motivación, flexibilidad, creatividad, comprensión y emprendimiento, dentro de una perspectiva de procesamiento metacognitivo, mejoramiento continuo y compromiso ético, con la meta de contribuir al desarrollo personal, la construcción y afianzamiento del tejido social, la búsqueda continua del desarrollo económico-empresarial sostenible, y el cuidado y protección del ambiente y de las especies vivas.” [2]

De los múltiples aspectos que encierra la definición, cobran especial importancia el reconocimiento de la complejidad de los procesos de aprendizaje, como así también el sentido de reto, motivación y creatividad.

Recuperando el contexto de la sociedad del conocimiento y considerando el ámbito tecnológico del presente trabajo, en particular en el campo de la informática, se reafirma la necesidad de articular el pensamiento con la acción: “Las propuestas de enseñanza y aprendizaje en el marco de la sociedad del conocimiento deberán integrar un sistema educativo cuyo objetivo sean las operaciones de pensamiento, pero no en el marco del pensamiento lógico tradicional, sino dentro de un modelo donde las operaciones de pensamiento puedan expresarse en competencias de acción que sean com-

petencias complejas en las que se mezcla conocimiento abstracto con experiencia, al cual definimos como pensamiento tecnológico." [3]

En este sentido, la disciplina de programación se caracteriza por tener una base conceptual densa que a la vez puede ser entendida como herramienta para resolver problemas concretos. Sin entrar en las siempre presentes discusiones acerca de los límites y alcances de cada terminalidad en la que la programación tiene un lugar importante en el ámbito universitario, ya sea en una ingeniería, licenciatura o tecnicatura, ya sea bajo el paraguas de sistemas de información, informática, o ciencias de la computación, asumiendo que hay matices, cabe reconocer que todas ellas adquieren un marcado enfoque instrumental: se trata de un conocimiento teórico que permite construir software que funcione, que tiene sentido en un contexto determinado para resolver problemas concretos.

De alguna manera, si bien esta interrelación fecunda entre teoría y práctica es compartida con otras corrientes educativas, en particular remite a la perspectiva crítica que interpreta la relación práctica/teoría desde la acción/reflexión: "Si los hombres son seres del quehacer esto se debe a que su hacer es acción y reflexión. Es praxis. Es transformación del mundo. Y, por ello mismo, todo hacer del quehacer debe tener, necesariamente una teoría que la ilumine. El quehacer es teoría y práctica. Es reflexión y acción. No puede reducirse ni al verbalismo ni al activismo." [4]

Metodológicamente, asumiendo en cierta medida el doble rol de los autores en tanto parte también del equipo docente en cuestión, se recuperan los aportes de la investigación-acción participativa, [5] que ayudan a conservar criterios de distancia epistemológica a la vez que valora la posibilidad de investigar sobre cierta problemática social, en este caso educativa, mientras se busca incidir sobre dicha realidad.

Uno de los elementos importantes que se consideraron fue la motivación de los estudiantes a participar de una actividad que los desafíe e integre, basándose en el concepto de vínculo educativo, diagramado por el triángulo de Herbart. En particular, se aventura la hipótesis de que no siempre se genera espontáneamente un deseo en los estudiantes que los motive a moverse al objeto de conocimiento. Este vínculo educativo no viene dado y se debe construir. "Requiere de un trabajo de transmisión por parte del agente y de un trabajo, también, de apropiación, de adquisición por parte del sujeto de la educación." [6]

## 2 Metodología y descripción de la experiencia

Como recorte temático y espacio temporal, se asume la temática de la enseñanza de la programación orientada a objetos en el marco de carreras universitarias de informática de diferentes casas de estudios durante el año 2019. En todas ellas se utilizaron las mismas herramientas tecnológicas, material bibliográfico y una propuesta pedagógica similar. Si bien los equipos docentes son diferentes, existe una fluida comunicación entre ellos, teniendo algunos profesores que trabajan simultáneamente en diferentes universidades y una historia compartida en cuanto a elaboración de materiales y

herramientas didácticas. Como referencia, también se toman algunos aspectos de lo sucedido en años anteriores.

Las materias y sus correspondientes universidades son:

- “Algoritmos I” (cuatrimestral) en la Universidad Nacional de San Martín.
- “Objetos I” (cuatrimestral) en la Universidad Nacional de Quilmes.
- “Paradigmas de Programación” (anual) en la Universidad Tecnológica Nacional, Facultad Regional Delta.
- “Paradigmas de Programación” (anual) en la Universidad Tecnológica Nacional, Facultad Regional Buenos Aires.

Las denominaciones y extensiones de las carreras donde se encuentran son diferentes, en algunos de los casos las materias incluyen además otras temáticas, pero todas tienen en común la enseñanza inicial del paradigma de programación orientado a objetos, con el objetivo de formar un profesional informático.

El punto de partida fue que revisar la forma en que se enseñaba la programación orientada a objetos, prestando especial atención a las competencias que se busca desarrollar como eje organizador de los contenidos y metodologías a aplicar. Se constata la importancia de una serie de criterios que estaban presentes y surgen algunos aspectos que despertaron interrogantes e inquietudes entre los docentes.

Entrando en mayor detalle, se confirma la importancia de saber desarrollar software en términos del paradigma de objetos. Los contenidos clásicos del paradigma de objetos, tales como mensajes, polimorfismo, clases, herencia, son entendidos como herramientas que tienen sentido en la medida que se cuenta con la habilidad para poder aplicarlos en situaciones nuevas, que resultan útiles para resolver problemas y que permiten desarrollar software robusto. Los diversos instrumentos de ejercitación y evaluación responden a ello.

También se confirma lo pertinente de articular con otras disciplinas, concretamente con las referidas a diseño y análisis de sistemas, que son necesarias para poder comprender los elementos de un problema y modelar una solución, y útiles para pasar de dicho planteo a implementar el programa que lo resuelve.

Se constata que muchas de las actividades propuestas tradicionalmente, como trabajos en equipo y presentaciones orales, apuntan a promover otro conjunto de competencias relacionadas con la capacidad de comentar y explicar determinadas soluciones, programas o diseños, relacionándolos con los conceptos teóricos vistos previamente.

Por último, otra de las convicciones que reafirma el equipo docente es entender la tarea de programar, no como una acción mecánica que concluye con el programa escrito, sino asumiendo la escritura como una tarea más dentro de una serie de actividades que son propias de la dinámica del desarrollo de software profesionalmente, tales como realizar *tests*, organizar el desarrollo con sistemas de versionado, el uso de un entorno integrado de desarrollo y otras herramientas. Este camino llevó a la incorpo-

ración de *Wollok* como lenguaje y entorno de programación que acompaña el proceso de enseñanza.

Con todo esto, una posición cómoda hubiera sido quedarse tranquilo que se están haciendo las cosas bien, pero tomando la perspectiva de competencias como un desafío crítico a repensar las prácticas concretas, surgieron nuevos interrogantes, búsquedas y propuestas de acción.

Los principales aspectos que se detectaron para trabajar fueron:

- La contextualización de los programas realizados de manera que puedan ser utilizados por un usuario final, habilitando que no sólo el autor del trabajo, sino también otros estudiantes y el resto de la comunidad pueda interactuar con sus producciones de manera más fluida.
- Un replanteo metodológico del hecho de tener siempre los requerimientos determinados por los docentes, incorporando la posibilidad de que el estudiante defina sus propias metas y objetivos para favorecer de esta manera la iniciativa e incentivar mayores niveles de creatividad y autonomía.
- La motivación y el entusiasmo durante el transcurso de la cursada de la materia y la búsqueda de mayor participación y compromiso con la tarea, en un clima distendido.

Para lograrlo se puso en marcha una propuesta pedagógica que, manteniendo el esquema general de cursada, introdujo una nueva modalidad de trabajo práctico integrador tendiente a abordar, ponderar y fomentar otras habilidades de los estudiantes, sin descuidar los aspectos que se consideran valiosos del enfoque anterior.

Para dar cuenta del resultado de esta propuesta de trabajo se tuvieron en cuenta indicadores cualitativos y cuantitativos, combinando diferentes instrumentos y estrategias de recolección de datos. Se recuperaron datos básicos de inscripción, conformación de grupos y aprobación de las materias y también se construyeron estadísticas específicas respecto de la intensidad y volumen de los trabajos realizados. El corpus de trabajos prácticos elaborado fue segmentado para su análisis que va desde la búsqueda de algunas variables generales hasta el desmembramiento de una muestra específica para hacer un estudio más detallado, agregando también una lectura interpretativa de algunos casos particulares que se consideraron significativos. La perspectiva de los estudiantes fue incorporada a partir de su propia voz, en tanto protagonistas del proceso de aprendizaje. La mirada docente estuvo presente de diferentes maneras: por un lado, expresada directamente mediante su opinión y reflexión sobre su rol así como su percepción sobre el seguimiento de los trabajos prácticos y las interacciones que fueron teniendo con los estudiantes en su realización; y por otra parte mediada por las correcciones y calificaciones escritas que hicieron de los trabajos en su presentación final.

## 2.1 Los juegos como propuesta de trabajo

A partir de las nuevas perspectivas generadas por los análisis realizados, se diseñó una actividad diferente a la que se venía empleando: un trabajo práctico grupal de desarrollo de un videojuego a elección de los estudiantes, que debía tener como única regla que se usen las herramientas presentadas en la materia.

La consigna de trabajo fue la misma en todos los cursos: "hacer un juego". El alcance y la funcionalidad la determinan los estudiantes dentro de márgenes razonables dados en el seguimiento docente, para evitar caer en una trivialidad que no sirva para aprender o pretender tareas de demasiada complejidad que terminen generando frustración. La forma de trabajo mantuvo unidad de criterio, con diferencias sutiles en cada curso: mayores o menores plazos de realización, con grupos más numerosos o pequeños, y con matices en la forma de seguimiento del trabajo propia del estilo y experiencia de cada docente a cargo de cursos y sus ayudantes.

Cabe reconocer que una preocupación que estuvo presente en las conversaciones docentes al momento de decidir este cambio fue el riesgo de que por abocarse a estas nuevas actividades los estudiantes descuidaran la carga teórica de la materia y la correcta aplicación de los conceptos. En otras palabras, que con tal de lograr que el juego sea más lindo o divertido, o que funcione de cierta manera, recurran a malas prácticas de programación, dejando de lado los principios del paradigma de la programación orientada a objetos presentados a lo largo de la cursada. Conscientes de ello, se procedió a llevar adelante la propuesta, aclarando a los estudiantes que el criterio de evaluación no sería sólo ver el juego funcionando, sino que se analizaría cómo fue construido el código desde un punto de vista conceptual.

La herramienta a usar fue el lenguaje *Wollok*, que ya se venía utilizando como recurso pedagógico para enseñar a programar en objetos, y en particular una biblioteca llamada *Wollok Game*. Se trata de una interfaz que permite construir juegos y aplicaciones gráficas interactivas de menor o mayor complejidad, diseñada para enseñar los diferentes conceptos del paradigma. Se basa en la asociación de los objetos presentes en el código con imágenes y posiciones en la pantalla de manera que la ejecución del programa sea visible, a la vez permite al estudiante interactuar con un usuario vinculando ciertos eventos de entrada/salida con su código. También incluye facilidades propias de los videojuegos tales como desplazamientos y colisiones de elementos visuales. La primera versión de *Wollok Game* fue desarrollada por estudiantes como proyecto final de ingeniería y en sintonía con la dinámica colaborativa y filosofía de software libre que tiene el mismo proyecto *Wollok*, se incorporó luego al lenguaje como una herramienta más.

## 3 Desarrollo y Resultados

Inicialmente, se identificaron los grupos que no lograron presentar su trabajo en los tiempos pautados o que lo hicieron pero de forma muy incompleta. Independiente-

mente de cómo siguió en cada curso la situación académica de los correspondientes estudiantes, en cuanto a las instancias de recuperación, la reubicación de sus integrantes en otros equipos o eventualmente el abandono o desaprobación de la materia, se trata de trabajos que no alcanzaron los objetivos planteados y que se decidió apartarlos de las instancias posteriores del estudio. En esta situación se contabilizó un total de 9 grupos, lo que en promedio se aproxima a uno por curso, y que constituye una proporción compatible con los indicadores históricos.

De esta manera, la primera aproximación al territorio de estudio, y para tener una referencia que permita dimensionar el alcance, muestra un total de 77 juegos, que se distribuyen en distintas asignaturas y universidades como informa la Tabla 1.

**Tabla 1.** Distribución de los cursos y juegos analizados.

Universidad (Facultad)	Cantidad de cursos	Cantidad estudiantes	Cantidad juegos
UTN (FRD)	1	41	12
UTN (FRBA)	5	154	32
UNSAM	2	49	15
UNQ	2	60	18
<b>Total</b>	<b>10</b>	<b>304</b>	<b>77</b>

Para viabilizar y organizar el análisis de todo el universo de todos los juegos presentados y establecer relaciones, se procedió con diferentes criterios de selección y formas de abordaje.

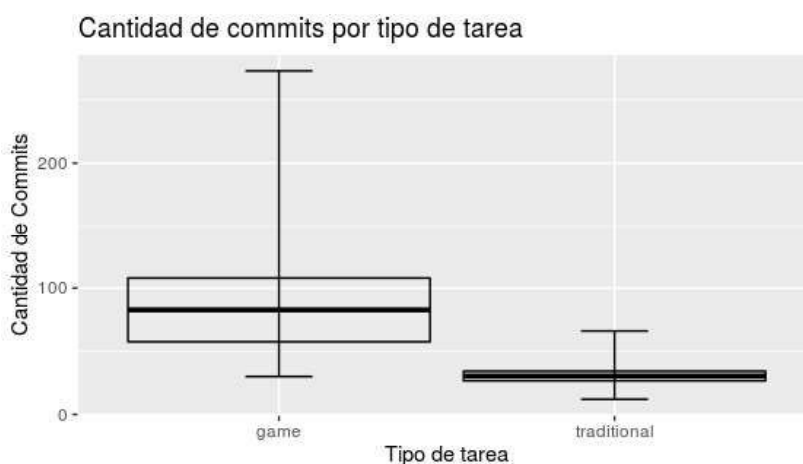
En una primer recorrida por todos los juegos, se constata la enorme variedad de estilos, recursos e ideas que surgieron. Hubo juegos totalmente originales de lucha, de carreras, de búsquedas y de persecución, con planteos totalmente originales. Otros hicieron su propia versión de videojuegos clásicos, como el "pacman", "arkanoid" o "space invaders", por nombrar algunos. De alguno de ellos, por ejemplo el "bomberman" o "battle city", hubo varias implementaciones y todas con alcances y aspectos diferentes. También estuvieron presentes los juegos de cartas y otros juegos de mesa, desafíos de preguntas y respuestas, juegos de ingenio, entre otros. Juegos con una única pantalla o juegos que al ir subiendo de nivel van cambiando de escenario, personajes y desafíos. Simples o complejos. Para jugar un par de minutos o para quedarse horas hasta ganar o superar el récord anterior.

### 3.1 Volumen e intensidad del desarrollo

Sabiendo lo imposible de relevar el tiempo real invertido por los integrantes de cada grupo en el desarrollo de los juegos, se definieron algunos indicadores para cuanti-

ficar el volumen de trabajo realizado y a partir de allí inferir el esfuerzo y compromiso con el trabajo.

El primer indicador es la cantidad de interacciones que tuvieron los grupos con su trabajo, a partir de los *commits* realizados en los repositorios de código donde se versionan los trabajos prácticos. Teniendo en cuenta que, más allá de los diversos estilos de trabajo, se entiende que el *commit* es la unidad de trabajo con la cual va evolucionando un proyecto de software, ya sea para agregar una funcionalidad puntual, resolver una falla o mejorar cierta porción del programa, su cantidad sugiere la intensidad del trabajo realizado, el nivel de interacción, como así también los cambios sucesivos que se fueron haciendo hasta poner a punto el software.



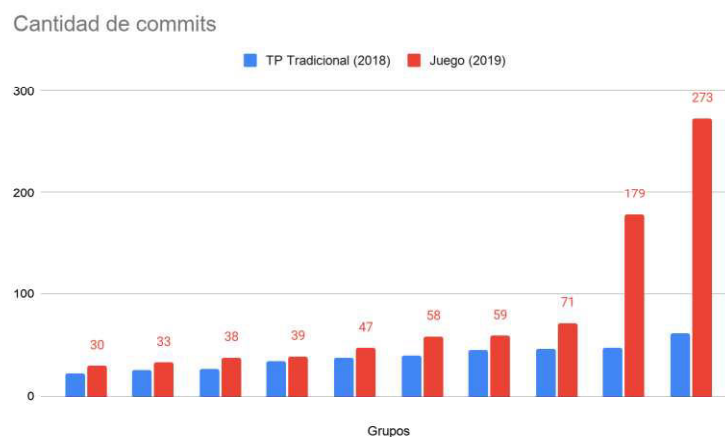
**Fig. 1.** Comparación de la cantidad de *commits* realizado por los grupos de estudiantes de todos los cursos involucrados entre la propuesta actual de realizar un juego (*game*) en 2019 y la propuesta utilizada el año anterior (*traditional*) en 2018. En el gráfico se puede observar el promedio, límites y desvío estándar normalizado.

En la Figura 1 se muestra que las cantidades de *commits* de los trabajos prácticos con la propuesta de actual con consigna abierta es ampliamente superior a la de un esquema anterior con consigna cerrada. Además, refleja cómo frente a un trabajo práctico tradicional la intensidad del trabajo es pareja entre los grupos, mientras que con la nueva propuesta de desarrollar un juego a consigna abierta la brecha entre ellas es significativa. Se puede interpretar que la propuesta favoreció, en general, una motivación que hizo que los estudiantes invirtieran más tiempo y esfuerzo en el armado de sus producciones, destacando que algunos de los grupos casos generó una intensidad de trabajo muy grande, superando ampliamente las expectativas de los docentes.

El cuadro de la Figura 2 muestra un recorte del análisis de los *commits* sobre uno de los cursos, en el que se comparan la misma cantidad de grupos de trabajo con am-

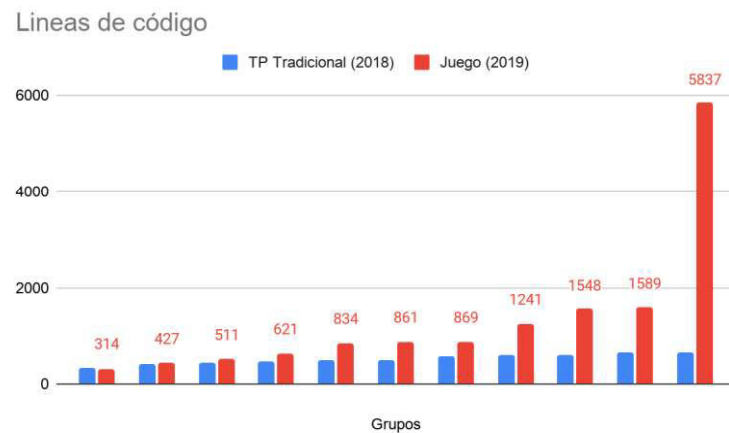


bas modalidades. Resulta evidente cómo dos de los grupos tuvieron una intensidad en el ritmo de trabajo y una búsqueda de mejora permanente muy marcada, que con un esquema de trabajo tradicional no se manifiesta.



**Fig. 2.** Comparación de la cantidad de *commits* realizado por dos grupos de estudiantes de una misma universidad entre la propuesta actual de realizar un juego y la propuesta tradicional utilizada el año anterior.

El otro indicador utilizado para analizar el tamaño de las producciones es la cantidad de líneas de código. La primera aclaración es acerca de lo relativo que puede ser este indicador, precisamente en una materia donde se enseña conceptos como la reutilización de código, que lleva a reducir su volumen en pos de lograr una mayor calidad; pero complementando esta variable con otros indicadores cualitativos que muestran que la calidad promedio del código fue buena en cuanto a la aplicación de los principios del paradigma, el tamaño del producto terminado es un indicio confiable respecto del compromiso con el trabajo, lo que refleja a su vez la motivación y el deseo de los estudiantes por hacer.



**Fig. 3.** Comparación de la cantidad de líneas de código realizado por dos grupos de estudiantes de una misma universidad entre la propuesta actual de realizar un juego y la propuesta tradicional utilizada el año anterior.

En la Figura 3 se puede observar una tendencia similar al caso anterior, ya que es esperable relacionar la cantidad de líneas de código con las que terminaron los trabajos y la cantidad de interacciones que sus autores tuvieron con ellos. En resumen

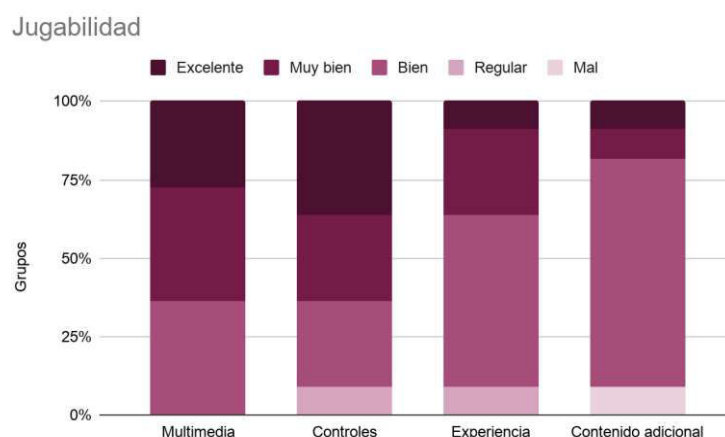
- Un 40% de los trabajos terminaron con una cantidad de líneas de código similar a los realizados con la propuesta anterior.
- Un 30% de los grupos produjeron 50% más líneas de código que la consigna tradicional.
- Otro 30% duplicaron la cantidad promedio de sus antecesores.
- Un grupo sobrepasó por demás todas las expectativas tenidas, haciendo una producción gigante en comparación al resto de sus compañeros.

Esto es consecuencia, en parte, de haber pasado de una consigna cerrada que impone los requerimientos del sistema, en donde existe un máximo al tamaño de la solución que se puede lograr, a una actividad de consigna libre, donde los propios autores del juego deciden, y van modificando, el alcance que desean sin sentirse limitados.

### 3.2 Indicadores de los juegos

Para hacer un estudio más en profundidad sobre la implementación de los trabajos se tomó una muestra de dos o tres juegos por curso y se los comparó a partir de ciertas categorías:

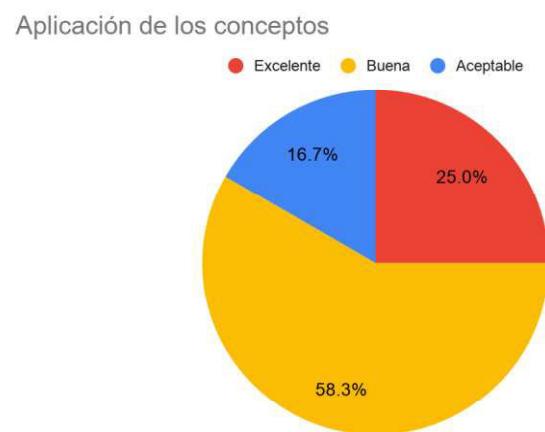
- La experiencia que otorga, es decir qué tan lograda resulta su jugabilidad y las sensaciones que despierta en el usuario.
- La facilidad de uso, lo que incluye el uso de los controles y la fluidez de la interacción.
- Los recursos multimedia, tanto en las imágenes, los sonidos, las animaciones y el aspecto visual en general.
- El contenido adicional. Se trata de instrucciones, guías de uso, *trailers* y otros elementos que no forman parte del software propiamente dicho, pero que, además de ser de utilidad para quien quiere aprender a jugar o ver de qué se trata, revela las ganas de los autores de mostrar sus producciones.
- La calidad del código, lo que representa cómo se aplican los conceptos del paradigma de programación.
- La complejidad del juego, ya sea en cuanto a la cantidad de personajes, variantes, niveles, menús, etc.



**Fig. 4.** Resultados de las categorías relacionadas con la jugabilidad de los juegos analizados representando los distintos niveles porcentualmente a la cantidad de juegos.

En la Figura 4 se presentan los primeros indicadores agrupados, porque analizados en conjunto dan cuenta que en su amplia mayoría los grupos lograron que sus juegos tengan una presentación que permite ser utilizados por cualquier persona y de manera agradable. También resulta valioso que una tercera parte incluso hayan superado las expectativas y alcanzado una muy buena forma de interactuar con los usuarios en los diferentes aspectos considerados.

También cabe señalar que además de las tareas de programación en sí, con el código necesario para su funcionamiento interno más el manejo de la interfaz gráfica de *Wollok Game*, cada grupo tuvo que imaginar y ponerse de acuerdo en el aspecto y reglas de juego en el caso de un juego original o adaptarlas en caso de implementar un juego preexistente, diseñar componentes visuales, hacer explicaciones de uso y preparar la presentación en los cursos donde se realizó una puesta en común pública. A su vez, se notó que en muchos casos se realizaron pruebas no sólo para que el programa funcione correctamente, sino para procurar que sea jugable, regulando por ejemplo la ubicación de los personajes, los puntajes, la dificultad de los objetivos, la velocidad de los movimientos y otras consideraciones que no se ven a simple vista pero que al avanzar en el desarrollo se torna imprescindible abordar.



**Fig. 5.** Medición sobre la aplicación de los conceptos vistos en clase en los trabajos analizados en una escala que va desde aquellos trabajos que aplicaron los conceptos mínimos y tenían algunas cosas a mejorar (Aceptable) hasta los que demostraron un buen manejo de todos los conceptos (Excelente).

Es importante señalar en la categoría de calidad del código, que aún con variaciones esperables, se muestra un buen promedio que deja tranquilos a los docentes sobre la apropiación y puesta en práctica de los conceptos. En la Figura 5 se evidencia un porcentaje de trabajos con ciertas dificultades, pero de una magnitud relativamente baja, compatible por otra parte con una porción de los estudiantes que aprueba la materia habiendo accedido a las instancias de recuperación de los exámenes parciales.

Generalmente, se vio que las herramientas elegidas y criterios aplicados eran adecuados para el problema planteado, lo cual es valioso y buscado dentro de la cursada; pero en algunos casos de diseños de juegos más simples, las herramientas utilizadas para implementarlo constituyen sólo una parte del alcance de la materia. A diferencia de la otra modalidad de trabajos prácticos, donde se construye una consigna que cubre

prácticamente todo el alcance temático de la asignatura, queda un interrogante abierto acerca de cómo poner en práctica los conceptos ausentes, aún reconociendo que también es parte del aprendizaje saber cuándo la situación no amerita el uso de una determinada herramienta.

Se debe tener en cuenta que en las planificaciones de los cursos estudiados, la fecha de entrega de estos trabajos fue cercana a los parciales, entre una o dos clases de diferencia, pero en todos los casos fue previo a las instancias de recuperación, por lo que la corrección que cada docente oportunamente hizo de estos trabajos fue un elemento más de preparación para finalmente aprobar la materia. Por otra parte, tratándose de un trabajo en equipo, no es sorprendente que entre los integrantes haya quienes comprendieron mejor los conceptos y pudieron reflejarlo en el código del trabajo y quienes no lo hicieron de igual manera y su participación en el trabajo fue menor o se centró en otros aspectos. Esto ayuda a explicar que integrantes de equipos que hicieron buenos trabajos luego tuvieron dificultad en la aprobación de los exámenes correspondientes.



**Fig. 6.** Gráfico sobre los resultados de la complejidad del código visto en los trabajos analizados basándose en la cantidad de componentes involucrados, sus relaciones y la lógica propia del dominio que se desea lograr.

Respecto a la complejidad del software creado por los grupos de trabajo, se puede observar que la mitad de los trabajos alcanzaron el nivel de complejidad mínimo esperable de una cursada universitaria, mientras que la otra mitad sobrepasó en mayor o menor medida esa expectativa. Los resultados de la primera mitad son posibles, en parte, porque la herramienta de *Wollok Game* fue diseñada con fines didácticos, para ser usada en un curso inicial de programación orientada a objetos y permitiendo la

creación videojuegos sin necesidad de introducir conceptos que no estaban en los planes de estudio o complejizando de más las prácticas de los cursos.

Aún así, los docentes percibieron un aumento en la complejidad promedio de los trabajos presentados adjudicándole ese resultado a la incorporación de la interfaz gráfica y la posibilidad de interactuar con los programas de una manera más amena. Esta hipótesis nace de observar a los grupos enfocarse en ciertas lógicas del programa, para lograr algún efecto en el juego o para arreglar errores mínimos, que con la metodología anterior no eran pertinentes. Por desgracia, no contamos con los datos de los trabajos anteriores necesarios para comprobarlo, pero quedará pendiente para trabajos futuros.

### 3.3 Aprobación de la materia

Reconociendo las limitaciones y objeciones que puede hacerse sobre utilizar una comparación de los porcentajes de aprobación entre diferentes cursos, incluso de las mismas asignaturas, para establecer conclusiones, se hizo el ejercicio de todas maneras y se constató la impresión de los docentes que no percibieron cambios en las calificaciones que les llamaran la atención. El porcentaje de estudiantes que aprobaron los exámenes de la correspondiente materia en los que se evalúa la programación orientada a objetos en relación a los inscriptos, durante el período estudiado, es aproximadamente del 70%, con algunas diferencias entre una facultad y otra, y pequeñas variantes si se toma un promedio simple de cada casa de estudios o se lo pondera por la cantidad de estudiantes, o si se asume como base los inscriptos formales en vez de quienes realmente asistieron al curso. El dato a rescatar es que dicho porcentaje no varió, en promedio, significativamente respecto al año anterior, más allá de las variaciones habituales que se observan entre cursos. Teniendo en cuenta además la multicausalidad de los resultados numéricos de una evaluación, no se puede afirmar que el cambio de modalidad haya tenido un impacto en los resultados de la evaluación, ni favoreciéndolo o empeorándolo.

Cabe señalar que en todos los cursos las evaluaciones consisten en el desarrollo de código a partir de un requerimiento concreto en el que los estudiantes tienen que demostrar la aplicación correcta de los conceptos del paradigma y no incluye temas relacionados a aspectos gráficos, interacción con el usuario u otras funcionalidades propias de la herramienta *Wollok Game* con la que se desarrollaron los juegos. Por otra parte, se trata de instancias de evaluación escritas y puntuales que apuntan a los conocimientos específicos y que no permiten dar cuenta de aprendizajes y capacidades tales como el trabajo en equipo, la motivación o la creatividad, que son algunos de los que se buscaba fomentar.

### 3.4 Casos significativos

Una tarea central fue analizar en profundidad algunos juegos puntuales, teniendo en cuenta el criterio de los profesores de los respectivos cursos y considerando no sólo el producto terminado sino reconstruyendo el proceso de desarrollo y el seguimiento docente. Se trata de trabajos que fueron significativos por diferentes motivos, en algunos casos incluso fueron comentados en clase con todo el curso.

A modo de ejemplo, ampliamos algunos de ellos.

#### Laberinto

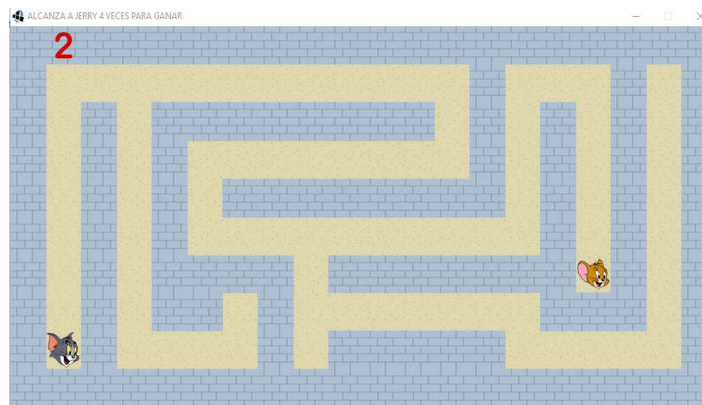


Fig. 7. Pantalla principal del juego Laberinto.

Como se observa en la Figura 7, la interfaz gráfica es sencilla, con un tablero con forma de laberinto, dos personajes y un conteo regresivo, lo que es suficiente para cumplir con su objetivo. La dificultad de juego consistió en programar los movimientos de los personajes considerando cuáles son los lugares permitidos y cuáles no, el manejo del tiempo y lo que sucede cuando *Tom* logra alcanzar a *Jerry*.

*Método para dibujar una pared del laberinto*

```
object nivel {
// ...
  method dibujarParedVertical(inicio, fin, pos){
    (inicio..fin).forEach{n => self.dibujar(
      new Laberinto( position = game.at(pos,n))
    )}
  }
// ...
}
```

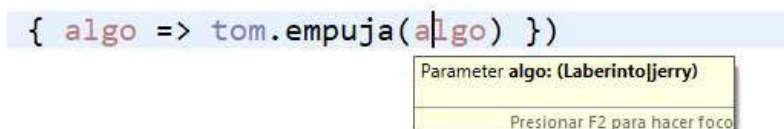
En este pequeño fragmento del código se puede ver la aplicación de varios conceptos específicos del paradigma, como manejo de colecciones, instanciación, envío de mensajes, manejo de referencias, uso de bloques. A su vez se plasman habilidades en cuanto al diseño, en particular la definición de nuevas abstracciones, tales como la clase `Laberinto` y el objeto `nivel`, que se articulan con `game` y `position` que ya vienen dados en la biblioteca el lenguaje.

En particular, en una primera versión no existía el método `dibujarParedVertical`, sino que para dibujar las numerosas paredes del laberinto se repetía la lógica a lo largo del código. Ante la indicación de los docentes al respecto, los estudiantes mejoraron la implementación creando dicho método dando cuenta de un aprendizaje en cuanto a la capacidad de abstracción, delegación y criterios de organización del código.

#### *Colisiones de objetos y polimorfismo en Wollok*

```
game.whenCollideDo(tom, {algo => tom.empuja(algo) })
// ...
object tom {
  method empuja(elemento) {
    elemento.moverse( )
  }
}
```

Otra porción de código donde se observa una buena aplicación de los conceptos es cuando se configura la colisión del objeto `tom`, que recibe como parámetro `algo` que colisiona con él, que en el contexto del juego puede ser el objeto `jerry` o el `laberinto`. El método `empuja` aplica el concepto de polimorfismo para poder delegar en el objeto recibido por parámetro el mensaje `moverse` y la realización de la tarea correspondiente sin necesidad de indagar acerca de su tipo o naturaleza. Nótese que no hay ninguna anotación de tipos acerca de la variable `algo` ni `elemento`, sino que el sistema de tipos de *Wollok* lo infiere. Esta información es visualizada por el entorno de desarrollo, para facilitar la tarea de los estudiantes, como lo muestra la Figura 8.



**Fig. 8.** Información inferida por el sistema de tipos de *Wollok*.

En conversaciones con los estudiantes autores del proyecto acerca de la posibilidad de extender el juego, una de las alternativas que precisamente gracias al polimorfismo resultaría sencillo implementar, consiste en agregar nuevos elementos que `tom` pueda



empujar. Con solo agregar el nuevo objeto y su correspondiente comportamiento (como mínimo el método `moverse()`) sería suficiente, sin modificar en absoluto la presente porción de código. En caso de hacerlo, el mensaje de información de tipos de la Figura 8 mostraría a los nuevos objetos polimórficos junto a `jerry` y el Laberinto.

Uno de los últimos agregados que surgió a lo largo del proceso de desarrollo y las sucesivas correcciones fue la incorporación del contador de tiempo, que si bien no implicó el uso de nuevos conceptos, permitió darle mayor realismo al juego.

### Bomberman



**Fig. 9.** Presentación del juego Bomberman.

Este es un juego muy completo, con numerosos personajes, niveles (incluyendo un tutorial del juego) efectos especiales y variantes, todos logrados con el manejo de conceptos básicos del paradigma. Sin dudas requirió de mucho trabajo en equipo, no sólo para el diseño de la solución, la escritura del código y las pruebas, sino también a otras tareas complementarias como la elaboración de las imágenes y sonidos. Sólo la variedad de más de 100 imágenes utilizadas es una muestra del nivel de detalle que el equipo de estudiantes consideró para realizar el juego a su manera, con su propio estilo. La creatividad se pone de manifiesto en numerosos momentos, tanto de la dinámica del juego en sí, como en el aspecto visual, con reminiscencias de videojuegos de otras décadas, como se observa en la Figura 9.

#### *Modelado de interacciones*

```
// JUGADOR
class Jugador {
// ...
power.chocoJugador(self)
```

```

// ...
}
//ESCUDO
class PowerUpEscudo inherits PowerUP{
    override method chocoJugador( jugador) {
        jugador.ponerEscudo()
    }
}
//MAS BOMBAS
class PowerUpMasBombas inherits PowerUP{
    override method chocoJugador(jugador) {
        jugador.cambiarBombasEnPantalla(1)
    }
}
//AGREGAR VIDA
class PowerUpAgregarVida inherits PowerUP{
    override method chocoJugador(jugador) {
        jugador.sumarVida()
    }
}
}

```

En el código se pueden ver cómo, a pesar del esfuerzo puesto en material gráfico, no se descuidó la aplicación de los conceptos importantes, tales como clases, herencia, redefinición y polimorfismo. Puntualmente, el fragmento elegido refleja criterios de modelado complejos, donde intuitivamente se implementan patrones de diseño que probablemente el estudiante conceptualizará en materias más avanzadas.

### **Bunny & Carrots**

En este juego, se vio un grupo de trabajo estimulado por la actividad de tal forma que sorprendió a los docentes. Como muestra de ello, el siguiente código tiene por finalidad crear el primer nivel del juego. En él se observa cómo los estudiantes inventaron un lenguaje específico para construir sus propios niveles mediante símbolos que se corresponden con los elementos visuales. Luego, los símbolos son interpretados y se crean los diferentes objetos visuales del juego en las posiciones específicas de acuerdo a la ubicación relativa de cada símbolo. La Figura 10 ilustra cómo se visualiza el tablero de juego que surge de la ejecución del mencionado código. Por ejemplo, nótese que la letra C corresponde a las zanahorias, la S al conejo y los números a las diferentes perspectivas de la cerca.

#### *Definición de un nivel del juego*

```

method levelOne() {
    const mapDefinition = []
    mapDefinition.add("GG  GG")
    mapDefinition.add("  E  ")
    mapDefinition.add(" 12223 ")
}

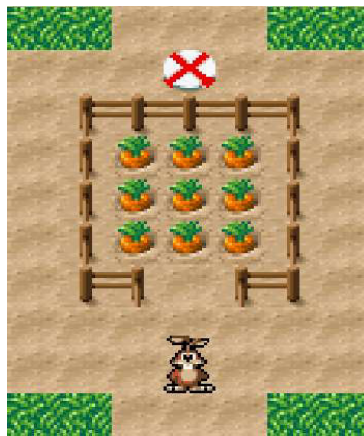
```

```

mapDefinition.add(" 4CCC6 ")
mapDefinition.add(" 4CCC6 ")
mapDefinition.add(" 4CCC6 ")
mapDefinition.add(" 73 19 ")
mapDefinition.add("      ")
mapDefinition.add("   S   ")
mapDefinition.add("GG   GG")

const map = mapBuilder
  .buildMapFromMatrix(mapDefinition)
return new Level(1, map)
}

```



**Fig. 10.** Uno de los niveles del juego Bunny & carrots.

Se puede observar un gran esfuerzo por evitar la lógica compleja del armado de un nivel, abstrayéndose del algoritmo y logrando una forma declarativa de construir los niveles, más allá de aplicar algún patrón específico, lo cual es una excelente práctica de programación. De esta manera, es posible definir numerosos y diferentes niveles con sólo especificar adecuadamente la secuencia de símbolos dentro de la lista.

### 3.5 Percepción de los estudiantes

Un elemento fundamental que se recupera son las opiniones de los mismos estudiantes sobre el trabajo realizado, mediante encuestas del final de la cursada y conversaciones personales. Aún asumiendo que se trata de los estudiantes que completaron la cursada de la materia, que falta la voz de quienes no lograron hacer el trabajo y que la relación asimétrica docente-estudiante puede generar una tendencia a que aquellos que no les satisfizo la propuesta prefieran no opinar, se percibe que muchos estudiantes no

sólo aprendieron sino disfrutaron del trabajo realizado. Y lo más valioso, que describen los motivos por los cuales lo hicieron.

*"Cuando te dan un trabajo, tenés que manejarte con las reglas que te dan, en cambio, con una consigna libre empezás a ver qué podés hacer con lo que sabés"*

*"Está bueno porque en realidad me motivó más hacer el game que todos los otros ejercicios. Podía agregarle las cosas extras más que quería. Por ejemplo, en el granjvilla le agregué un arbusto que no deja pasar, y le hice el efecto que cuando el personaje llegara a la punta apareciera por el otro lado."*

*"Lo que está bueno es que el juego nunca está terminado, siempre se le puede agregar algo. Te da idea de la escalabilidad que permite el paradigma de objetos. Al mismo juego le vas agregando más cosas, te vas cebando y querés más."*

*"En algún punto, queda en las ganas de cada uno. También podés cumplir con lo mínimo y listo."*

*"Con el game lo ves más tangible, algo que ves cómo está funcionando, que es interactivo, no sólo código y marquitas verdes del test".*

*"Es la idea de desafío: cómo hacer que eso que te imaginaste llegue a andar. Pero también ves de acotar las expectativas desmesuradas."*

*"Me di cuenta con las herramientas que tenía podía hacer más cosas de las que pensaba".*

*"Arrancamos pensando una idea que parecía sencilla pero que al empezar a implementarla vimos que era gigante. Aunque después, al ir haciéndolo y superando obstáculos, enfocándonos en resolver pequeñas partes, nos dimos cuenta que era posible."*

*"Terminás haciendo lo que querés, como querés (o como podés). Nos pasó de querer hacer cosas, de no poder y luego de intentar lo sacamos andando como nos salió. Más adelante, lo volvimos a ver y nos preguntábamos ¡cómo pudimos hacerlo así! y ponernos a corregirlo."*

*"No te quedás con lo de la cursada nomás, podés probar otras cosas, ensayar si podés ir más allá."*

*"Se puede continuar jugando cuando hay un error, ya que en el game no se interrumpe la ejecución. Está bueno para manejar mejor los errores."*

*"Era el primer trabajo de programación que hacíamos en grupo, tuvimos que aprender a organizarnos según los tiempos disponibles de cada uno. Aprendimos a dividir en distintas tareas el trabajo, de las que cada integrante se ocupaba en forma independiente, y a cómo unirlo todo después".*

*"Siempre me gustó el 'Space invader'... ¡y lo puede hacer!"*



**Fig. 11.** Juego del Space Invader.

Es interesante ver que hay varias opiniones que apuntan a la motivación, el desafío y la creatividad, a la vez que hay otras que dan cuenta de las competencias más específicas de lo técnico de la disciplina, como también varias que refieren a un aprendizaje en la dinámica del desarrollo de software.

Interpretando las expresiones y la forma de expresarlas se percibe la satisfacción por lo realizado. No pocos confiesan que una de las motivaciones por ingresar a la carrera tiene que ver con el mundo de los videojuegos. Otro elemento destacado, aunque expresado de diferentes maneras por los estudiantes, apunta a que el trabajo realizado les permitió una comprensión más integral de la programación y su fuerte vínculo con otras áreas.

Por otra parte, se dieron situaciones interesantes de colaboración en los laboratorios, de querer ver lo que hacía otro grupo y cómo había logrado tal o cual efecto. Incluso, ya cuando se estaban presentando los trabajos terminados, la posibilidad de probar los juegos en clase fue vivida con mucha intensidad y distensión.

Se destaca la experiencia de uno de los estudiantes que además de hacer con sus compañeros uno de los juegos, elaboró y publicó un tutorial de cómo usar *Wollok Game*, con ejemplos personales, indicaciones que complementan la documentación oficial y sugerencias de cómo aprovechar ciertas funcionalidades. Otro caso es el de un par de estudiantes que habiendo aprobado la materia manifestaron su interés por sumarse al equipo de desarrollo de *Wollok* y luego participaron de alguna actividad que resultaron en valiosos aportes al proyecto. Sin dudas no es lo mayoritario, pero se

trata de una oportunidad más que se le ofrece al estudiante para encauzar sus inquietudes y ayudar a despertar su vocación por la investigación.

### 3.6 Mirada docente

La primera reflexión de los docentes de los cursos involucrados, desde su propia percepción al terminar la cursada y previo a tener sistematizado los resultados, es de satisfacción por la experiencia realizada. Quienes aún tenían cierto temor acerca de una eventual dispersión, distracción o pérdida de foco sobre los aspectos conceptuales de las asignaturas terminaron de disipar sus dudas. Una sensación reiterada fue de grato asombro por algunas de las producciones de los estudiantes, ya sea por su volumen, su calidad de presentación o su profesionalismo.

Como muestra de estas reacciones, uno de los docentes lo expresa de esta manera "*¡Estoy re-emocionado, posta, es increíble que esta herramienta permita que los pibes descubran todas sus capacidades!*". Otro reconoce que "*de pronto me hacían preguntas sobre cómo hacer para lograr tal efecto o hacer que suceda tal cosa, y tuve que sentarme con ellos a aprender, probar y jugar yo también*".

Un aspecto que se destaca es el clima agradable que se percibió en el aula, en especial en las clases que se trabajó sobre el juego, pero que también se extendió a otras a medida que avanzaba la cursada. Se notó buena interacción entre los grupos, que preguntaban y se interesaban por los avances de los otros, y que luego, con el juego terminado, querían poder probarlo. En particular, en varios de los cursos se hicieron clases enteramente destinadas a exponer los juegos, viéndolos en acción, mostrando también el código e incluso permitiendo que sus compañeros lo prueben y jueguen un rato. Además, en algunos de ellos hubo una o más puestas en común con los avances parciales, lo que anticipó las posibilidades de intercambio y la circulación de ideas.

Otra observación importante es que, durante los avances, las correcciones por parte de los docentes con frecuencia detectaron formas de implementar las funcionalidades que efectivamente funcionaban, pero descuidaban los conceptos que se querían transmitir, por ejemplo repitiendo lógica, con poca flexibilidad en la solución, con un modelado innecesariamente rebuscado y otras dificultades. En estos casos, se abordaron las orientaciones pertinentes y se pidieron los cambios en la implementación, aprovechando instancias de comunicación más personales entre los docentes y estudiantes. Por su parte, también es visible una cierta correlación entre los grupos con trabajos más flojos y los que tuvieron menos instancias de seguimiento o menor intercambio en las actividades de laboratorio.

En las reuniones docentes, en particular en las materias con mayor cantidad de cursos, se produjeron diálogos interesantes. Fue tema de discusión, se contó la experiencia y se dedicó tiempo incluso a mostrar los juegos terminados y jugar con ellos para ver más de cerca el nivel de producción. Al intercambiar con profesores de otros cursos como así también a docentes de otras universidades con materias afines que también utilizan *Wollok* para enseñar a programar en objetos pero que no implementaron propuestas de trabajo con *Wollok Game*, las reacciones fueron variadas. Hay quienes

se mostraron favorables y plantearon la posibilidad de repensar su planificación para incorporarlo, de alguna manera, en próximas cursadas, mientras que otros se manifiestan más escépticos o al menos con dudas acerca de si vale la pena el esfuerzo de encarar un cambio de esa naturaleza.

En relación a esto último, es oportuno mencionar que la modalidad de trabajo realizada compromete de otra manera al estudiante, pero a la vez también supone una mayor dedicación por parte del docente. La dinámica de avances y correcciones parciales resultó fundamental para orientar el trabajo de los grupos, requiriendo un mayor nivel de seguimiento por parte de los docentes que con otras modalidades de trabajo, además de tener que estar atentos a trabajos heterogéneos e interpretar no sólo lo que cada grupo llegó a hacer sino también lo que quiso lograr y tal vez no pudo. En contrapartida, es justo señalar también que requiere menos tiempo de elaboración previa que una consigna de trabajo más cerrada que busque articular un abanico amplio de conceptos en un mismo problema.

#### 4 Conclusiones

Se constata que el aprendizaje no se limitó a comprender conceptos sino que se los puso en práctica para resolver un problema que el mismo grupo de estudiantes pudo construir. El proceso permitió favorecer el desarrollo de competencias, algunas de las cuales anteriormente también se lograban con otras estrategias y otras nuevas que dejaron satisfechos a los respectivos equipos docentes. En otro plano, se valora la versatilidad y sencillez que ofrece la plataforma de *Wollok Game* para sostener tecnológicamente todo el proceso llevado adelante.

El elemento básico que aporta la tranquilidad de haber alcanzado el umbral mínimo que se esperaba es que la modalidad de trabajo práctico empleada permitió que los estudiantes aprendan a desarrollar software en los términos del paradigma de la programación orientada a objetos, en similares términos que con las estrategias anteriores. En otras palabras, se muestra que no por la intención de desarrollar otro tipo de competencias entre los estudiantes se descuidaron los contenidos habituales de las materias. En este sentido, los datos estadísticos de aprobación general de las asignaturas son consistentes con el estudio sobre la calidad del código de los juegos realizados en cuanto a una buena aplicación de los conceptos, similar a años anteriores. A su vez, la percepción de los docentes lo confirma.

De todas maneras, sin dejar de lado que hubo matices en cuanto a la apropiación de los conceptos, que hubo algunos juegos de modesto alcance y que aún siendo minoritaria hay una parte de los estudiantes que habiendo hecho el trabajo presenta luego dificultades al momento de los exámenes, resulta pertinente considerar esta propuesta como un instrumento valioso en la medida que haya un seguimiento de cerca por parte de los docentes para monitorear el avance de los trabajos y que esté contextualizado dentro de una ejercitación práctica que se complemente con el abordaje de ciertos temas puntuales y permite focalizar en los estudiantes a quienes les cuesta más apro-

piarse de los conceptos. En particular, el relato del proceso de elaboración del juego del "Laberinto" muestra un ejemplo de cómo se fue logrando que un grupo con alguna dificultad en la aplicación de los conceptos pueda ir mejorando su trabajo.

El conjunto de producciones logradas puso en evidencia la creatividad de los estudiantes. El juego fue un espacio de diversión, de poder crear, donde los estudiantes pudieron transgredir ciertas restricciones. La jugabilidad lograda y la variedad de recursos gráficos relevada en los trabajos, como así también las expresiones de los estudiantes y las opiniones de los docentes dan cuenta de ello. Ciertamente no en todos los casos, ya que también hubo producciones básicas o que simplemente cumplieron con los objetivos mínimos, pero estuvo la posibilidad, el acompañamiento y muchos estudiantes lo aprovecharon con intensidad y mejor aún, lo disfrutaron. Muchos de los juegos tienen animaciones y efectos gráficos interesantes y originales a la vista del usuario final. Sin embargo, también hay ideas recurrentes en la forma de diseñar el código, como se detalla en el relato del juego "Bunny & Carrots", lo que sugiere que la dimensión creativa no sólo está en la artística de las pantallas del juego sino que hay también arte en la escritura del código.

Desde los diferentes indicadores analizados se observó un mayor volumen e intensidad de trabajo. Se lo vio no sólo en el código propiamente dicho, sino en la producción gráfica, de sonidos y demás elementos complementarios, en el tiempo dedicado a imaginar y probar el funcionamiento del juego y ponerlo a punto, que son imposibles de medir pero que se estiman mayores que con otras metodologías. No habiendo una consigna preestablecida que condicione el trabajo o que estipule parámetros previos de cuándo está terminado, la actitud de fuerte compromiso con la tarea se explica desde la motivación, las ganas y el entusiasmo de cada grupo de estudiantes de hacer algo que los satisfaga a ellos mismos antes que congratular a los docentes. Ciertamente los estudiantes sabían que por simple que fuera el juego que elegían hacer debía estar terminado y funcionando en los plazos acordados, lo cual constituye un piso -y de hecho como ya se mencionó hubo una parte de los estudiantes que no lograron alcanzarlo-, pero no había un techo a alcanzar, por lo que cada grupo se puso sus metas y objetivos en función de sus intereses y posibilidades.

Desde otra perspectiva, cabe señalar que la propuesta pedagógica más que ser la responsable de motivar o entusiasmar al estudiante, fue la oportunidad para canalizar la motivación y ganas de hacer que los estudiantes ya tenían previamente y que muchas veces la estructura institucional no deja aflorar con toda su contundencia.

Retomando los objetivos iniciales, se constata que la propuesta pedagógica implementada, que surgió de la revisión de la planificación desde la perspectiva de competencias, en términos generales fue satisfactoria, propició un incremento del deseo por hacer, por aprender, por programar y por participar de muchos de los estudiantes de los cursos y, en definitiva, pone en evidencia el protagonismo del estudiante en el proceso de aprendizaje y reubica al docente en un rol de mediador o facilitador. Más allá de una diferente forma de elaborar o proponer una consigna de trabajo, lo que se valora es el cambio de enfoque por el cual el docente deja de ser quien dictamina qué hacer o no y se vuelve un instrumento que ayuda a que el estudiante desarrolle su propio



proyecto. Probablemente éste haya sido el aprendizaje más importante para los equipos docentes involucrados.

La pregunta que aparece a modo de reflexión final es si en el afán de buscar garantizar ciertos estándares entre los estudiantes, al plantear consignas de trabajo muy específicas o por querer establecer un mínimo a superar, se termina fijando también un tope máximo a alcanzar que limita las potencialidades. Lo que se puede vislumbrar en esta experiencia es que con una propuesta de trabajo que apela a la creatividad, el interés y la motivación de los estudiantes, aún sin poder garantizar resultados mínimos uniformes, en la medida que se ofrecen condiciones apropiadas, se abre la posibilidad para un aprendizaje significativo sin límites, y que como pasó en esta ocasión, puede generar resultados sorprendentes.

Una inquietud que surgió, que trasciende los objetivos del presente trabajo, y que deja más interrogantes abiertos que certezas es acerca de los criterios de evaluación. En esta experiencia, el modelo de exámenes se enfocó en las competencias más vinculadas a poner en práctica los conceptos teóricos, como habitualmente se hizo en las asignaturas, pero queda pendiente seguir revisando los criterios y modalidades de evaluación y calificación acorde a la incorporación de nuevas competencias en la propuesta pedagógica.

## Referencias

1. Coll, C.: Las competencias en la educación escolar: algo más que una moda y mucho menos que un remedio. *Revista Aula de Innovación Educativa*. Núm. 161. Editorial Grao (2007)
2. Tobón, S.: El enfoque complejo de las competencias y el diseño curricular por ciclos propedéuticos. *Acción Pedagógica* 16 (2007) 14-2
3. Agüerrondo, I.: Conocimiento complejo y competencias educativas. UNESCO Oficina Internacional de Educación. Ginebra (2009)
4. Freire, P.: *Pedagogía del Oprimido*. Ed. Tierra Nueva. Montevideo (1970)
5. Montero M.: *La investigación Acción Participativa. Orígenes, definición y fundamentación epistemológica y teoría. Hacer para transformar. El método en la psicología comunitaria*; Editorial Paidós, Bs.As. (2011)
6. Moyano, S.: Los contenidos educativos: bienes culturales y filiación social. Clase 11 del Diploma en Psicoanálisis y prácticas socioeducativas. FLACSO. (2010)