

Competencias en Programación Orientada a Objetos

Lucas Spigariol¹³⁴, Nicolas Passerini¹² y Nahuel Palumbo¹²³

¹ Universidad Nacional de San Martín, San Martín, Provincia de Buenos Aires, Argentina

² Universidad Nacional de Quilmes, Bernal, Buenos Aires, Argentina

³ Universidad Tecnológica Nacional - FRBA, CABA, Buenos Aires, Argentina

⁴ Universidad Tecnológica Nacional - FRD, Campana, Buenos Aires, Argentina
nahuel.palumbo@gmail.com

Abstract. 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 la elaboración de un proyecto integrador a lo largo de la cursada, que se trató de programar un juego, utilizando la herramienta Wollok Game, que es una biblioteca para facilitar el desarrollo de interfaces gráficas interactivas, que forma parte del software educativo Wollok.

La experiencia se llevó a cabo en tres universidades públicas diferentes, en asignaturas que comparten objetivos, materiales, herramientas y enfoques pedagógicos.

Keywords: Educación, Competencias, Programación, POO, Juegos.

1 Competencias en programación

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]

Como punto de partida, 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, flexibili-

dad, 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 competencias 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 técnica, 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 cierto problema.

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]

2 Descripción de la experiencia

Como recorte metodológico se asume la temática de la programación orientada a objetos en el marco de diferentes carreras universitarias, también 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. Los cuerpos docentes son diferentes, pero con fluida comunicación entre ellos:

- “Algoritmos I” (1° y 2° cuatrimestre) en la Universidad Nacional de San Martín.
- “Objetos I” (1° y 2° cuatrimestre) 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. (En algunos cursos)

La oportunidad de redefinición curricular que aporta la perspectiva de trabajar en competencias permitió confirmar antiguas intuiciones y, a la vez, incorporar nuevos elementos.

Por lado, se confirma que lo importante no es poder enumerar de memoria los conceptos sino poder aplicarlos para resolver problemas nuevos. En otras palabras, saber desarrollar software en términos del paradigma de objetos. Durante años, los contenidos clásicos del paradigma de objetos, tales como mensajes, polimorfismo, clases, herencia, fueron entendidos como herramientas para poner en práctica el resolver problemas y los diversos instrumentos de ejercitación y evaluación respondían 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, 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, una de las modificaciones curriculares más recientes que se incorporó parte de repensar el sentido clásico de la tarea de programar, donde más que entenderla como una acción de escritura que concluye con el programa escrito, se la asume como una más dentro de una serie de actividades que son propias de la dinámica de desarrollo profesional de software, tales como realizar tests, organizar el desarrollo con sistemas de versionado, el uso de un entorno integrado de desarrollo y otras herramientas, proceso que se dio junto a la incorporación 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 sobre ellos fueron:

- Cómo contextualizar los programas realizados dentro de un ambiente útil para personas ajenas a la materia, como la utilización de algún tipo de interfaz, para lograr que no solo el estudiante, sino el resto de la comunidad, pueda interactuar con sus producciones de manera más fluida.

- Cómo ampliar metodológicamente el hecho de siempre tener los requerimientos impuestos para incorporar la definición de los propios y favorecer de esta manera la iniciativa y una actitud emprendedora.
- Cómo propiciar mayores niveles de creatividad y sostener la motivación durante el transcurso de la cursada de la materia.
- Cómo formular proyectos integradores de todo el proceso de aprendizaje, que se complementen con la progresión de la secuencia didáctica.

3 Los juegos como propuesta de trabajo

Poniendo énfasis en las cuestiones motivacionales, la apuesta fue introducir videojuegos en el aula de programación. Reconociendo características que podrán ser analizadas bajo 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" [5].

Entonces, a partir de las nuevas perspectivas generadas por los análisis realizados, se diseñó una actividad diferente a la que se venía empleando: una actividad de desarrollo de un videojuego a elección del alumno, que debía tener como única regla que se pongan en práctica los conceptos aprendidos 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 de la propuesta, 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 tuvo matices en cada curso: mayores o menores plazos de realización, con grupos más numerosos o pequeños, con o sin presentación pública, con carácter de ejercitación o también de evaluación, pero resulta difícil atribuir a alguna de estas variables los matices en los resultados, que presentan menos diferencias entre un curso y otro, que dentro del mismo curso.

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 gráfica que permite construir juegos y aplicaciones interactivas de menor o mayor complejidad y desde allí ir descubriendo y aplicando 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 que permite al estudiante interactuar con su creación como un usuario, vinculando los eventos con el código. Incluye facilidades propias de los videojuegos tales como los 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 al lenguaje como una herramienta más.

4 Resultados

Los juegos fueron de lo más variados y superaron las expectativas. Frente a la posibilidad de que algunos hicieran juegos demasiado modestos, sólo en casos aislados fue necesario intervenir pidiendo mayor complejidad en la tarea a realizar. Hubo también algún caso de grupo que no terminó la tarea, de grupo con algún integrante que no participó del trabajo o de estudiantes que decidieron dejar la materia, pero en una magnitud similar a experiencias de años anteriores.

Entre la extensa lista de juegos elaborados, se encuentran los siguientes:

- Juego de ingenio Bunny y Carrots
- Pokemon
- Salvando al mundo, con caricaturas infantiles
- Esqueletos en el laberinto
- Juego de ingenio con gravedad
- NemoTest, Juego de memoria con personajes de Nemo
- Mario Bross juntando monedas
- Space invaders
- Laberinto
- Juego de lucha
- Bob esponja y las kangreburger
- Tateti
- Juego de la vivorita
- Combate del príncipe y el dragón
- Nave espacial explosiva

A modo de ejemplo, ampliamos algunos de ellos.

Laberinto

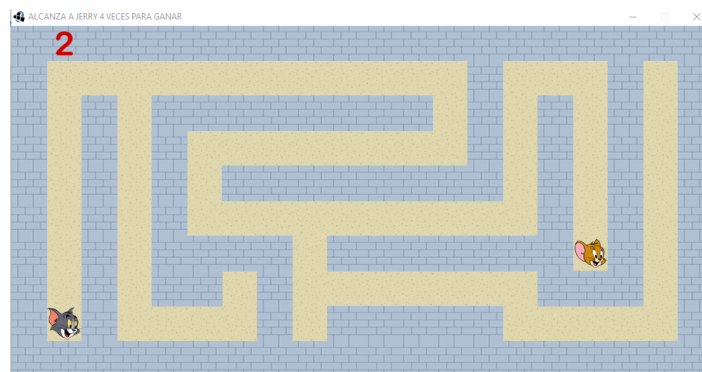


Figura 1. Tablero inicial del juego del laberinto.

Como se observa en la figura 1, 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.

```

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

```

Figura 2. Método para dibujar una pared del laberinto.

En un pequeño fragmento del código (figura 2) 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 primer 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 delegación y criterios de organización del código.

```

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

```

Figura 3. Colisiones de objetos y polimorfismo en Wollok.

En otra porción de código (figura 3) ve una buena aplicación de los conceptos, donde el objeto tom, recibe como parámetro algo que colisiona con él, que el en 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 que corresponda la realización de la tarea sin necesidad de indagar acerca de su tipo o naturaleza. Nótese que no hay ninguna anotación de tipos acerca de la variable algo, 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 4.

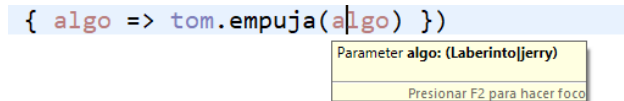


Figura 4. Información del sistema de tipos.

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 anterior 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

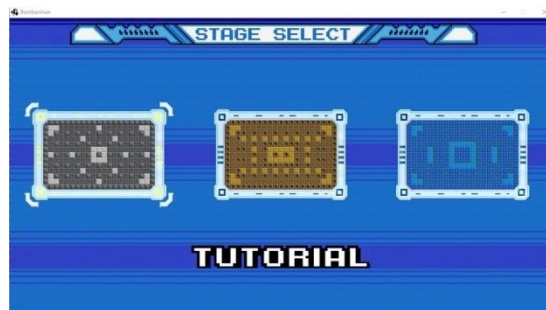


Figura 5. Menú principal del juego Bomberman, para elegir el nivel a jugar.

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 por ejemplo la elaboración de las imágenes y sonidos. Sólo la variedad y cantidad de imágenes utilizadas es una muestra del tiempo dedicado, lo que refleja la motivación del equipo de estudiantes en realizar el juego a su manera, con su propio estilo.

```

// 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()
    }
}

```

Figura 6. Modelado de las interacciones del jugador en el Bomberman.

En el código se pueden ver cómo el esfuerzo puesto en materia gráfico no 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.

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 videojuego de otras décadas.(Figura 7).



Figura 7. Presentación del juego Bomberman.

Bunny & Carrots

En este juego, se vio un grupo de trabajo estimulado por la actividad de tal forma que sorprendieron a los docentes. Como muestra de ello, el código de la figura 8 es el que crea 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 9 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 carrots y los números a las diferentes perspectivas de la cerca.

```
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)
}
```

Figura 8. Código para armar un nivel del juego "Bunny & carrots".

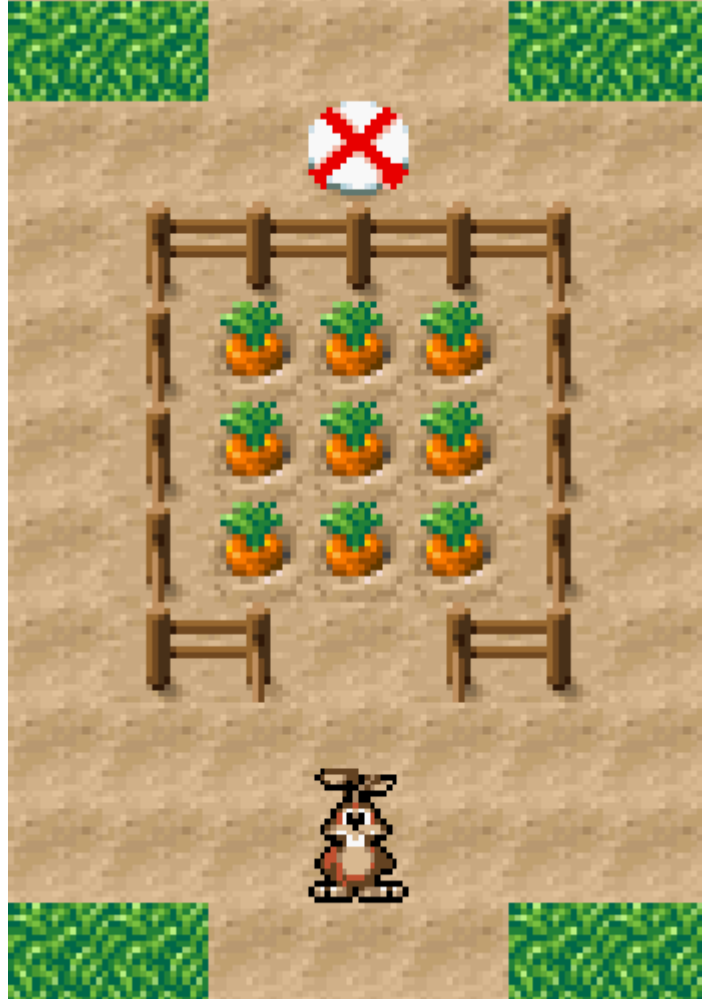


Figura 9. Uno de los niveles del juego "Bunny & carrots".

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.

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 una amplia mayoría de los estudiantes no sólo aprendió sino disfrutó del trabajo realizado.

"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 que 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 hacía 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!"

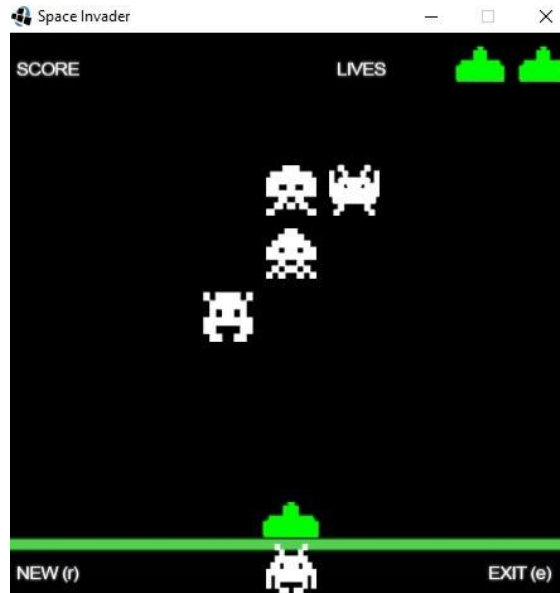


Figura 10. Space Invader.

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 en 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 features. 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 en ese sentido 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.

6 **Mirada docente**

Complementando el análisis de los docentes involucrados en la actividad, que en tanto autores del presente trabajo atraviesa toda su formulación, es digno de destacar la opinión de otros docentes. Al comentar la experiencia y mostrar los juegos terminados a profesores de otros cursos y universidades con materias afines que también utilizan Wollok pero que no implementaron propuestas de trabajo con Wollok Game, las reacciones fueron unánimemente favorables. Como muestra de ellas, 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!". Fue tema de análisis en reuniones de cátedra o área, se dedicó tiempo incluso a probar y jugar los diferentes jueguitos para ver más de cerca el nivel de producción y profesionalismo obtenido y en consecuencia fueron varios los que plantearon la posibilidad de reevaluar su planificación para incorporarlo, de alguna manera, en próximas cursadas.

Otra observación importante es que en los avances o correcciones parciales por parte de los docentes, con frecuencia se detectaron formas de implementar las funcionalidades que efectivamente funcionaban, pero sin aplicar adecuadamente los conceptos, 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 con los que tuvieron menos instancias de seguimiento o menor intercambio en las actividades de laboratorio.

7 **Conclusiones**

En primer lugar 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 satisfecho e incluso sorprendieron a los respectivos equipos docentes.

El conjunto de producciones logradas evidenció el funcionamiento del discurso de "hacer desear". El juego fue evidentemente un espacio de diversión, creativo, donde los estudiantes pudieron transgredir ciertas normas. Las posibilidades de crear superaron a las restricciones. La nueva propuesta didáctica probó ser muy enriquecedora y, si bien quedan muchas cuestiones para pensar, fue sorprendente el incremento del deseo por hacer, por aprender, por programar, por participar, un cambio en la actitud de casi todos los estudiantes de los cursos.

Se observó un mayor volumen de trabajo. Si bien la cantidad de líneas de código no es una métrica determinante de la calidad del trabajo ya que generalmente una

solución bien organizada es más breve que una que por ejemplo repite lógica, es un indicador que orienta sobre la magnitud del trabajo realizado y del tiempo invertido, que fue claramente mayor que frente a consignas anteriores, más cerradas en su formulación.

Sin variar los niveles de deserción, se notó una mayor motivación, participación y entusiasmo entre los que completaron y aprobaron la materia. 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 surgir con toda su contundencia. Esto también se vio reflejado las cursadas actuales, donde parte de los estudiantes ya vienen entusiasmados por la actividad que todavía no se enunció pero fueron contadas por compañeros que cursaron previamente la asignatura.

La modalidad de trabajo realizada, como ya se mencionó, compromete de otra manera al estudiante, pero a la vez también supone una mayor dedicación 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 lidiar con trabajos heterogéneos entre sí. 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, por lo que se trata también de una redistribución de tiempos y fundamentalmente un cambio de enfoque. Probablemente éste haya sido el aprendizaje más importantes para los equipos docentes involucrados.

En relación a la apropiación de los conceptos y su puesta en práctica, los trabajos muestra un buen desempeño, ciertamente con matices. Generalmente, se ve que se eligieron las herramientas adecuadas para el problema planteado, lo cual es valioso, pero en muchos casos, según el diseño del juego, las herramientas utilizadas para implementarlo constituyen un subconjunto acotado del alcance de la materia. Por lo tanto, se le reconoce un valor importante como parte del proceso de aprendizaje en competencias, pero por sí solo no es suficiente como instrumento de ejercitación o de evaluación.

Retomando las inquietudes 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, que confirma el protagonismo del estudiante en el proceso de aprendizaje y que más que garantizar resultados ofrece posibilidades para un aprendizaje significativo. A su vez, se valora que la plataforma de Wollok Game que sostuvo tecnológicamente todo el proceso es adecuada para la tarea emprendida.

Queda pendiente seguir revisando los criterios y modalidades de evaluación y calificación acorde a la incorporación de las competencias en la propuesta pedagógica.

Referencias

1. Coll, C (2007) "Las competencias en la educación escolar: algo más que una moda y mucho menos que un remedio" en Revista Aula de Innovación Educativa. Núm. 161. Editorial Grao.
2. Tobón, S. (2007). "El enfoque complejo de las competencias y el diseño curricular por ciclos propedéuticos." Acción Pedagógica 16: 14-28.
3. Aguerrondo, Inés (2009) Conocimiento complejo y competencias educativas UNESCO Oficina Internacional de Educación. Ginebra
4. Freire, P. (1970) Pedagogía del Oprimido. Ed. Tierra Nueva. Montevideo
5. Moyano, S (2010) "Los contenidos educativos: bienes culturales y filiación social" - Clase 11 del Diploma en Psicoanálisis y prácticas socioeducativas - FLACSO.