

Juegos para presentar conceptos básicos de Arquitectura de Computadores

Edurne Larraza Mendiluze, Olatz Arbelaitz Gallego,
Olatz Arregi Uriarte, Jose Ignacio Martín Aramburu
Depto. de Arquitectura y Tecnología de Computadores
Universidad del País Vasco (UPV/EHU)
20018 Donostia - San Sebastian
edurne.larraza@ehu.eus

Jose Francisco Lukas Mujika
Depto de Métodos de Investigación
y Diagnóstico en Educación
Universidad del País Vasco (UPV/EHU)
20018 Donostia - San Sebastian

Resumen

La mayoría de estudiantes de primer año en el grado de Inteligencia Artificial (IA) no ha tenido contacto previo con la arquitectura de computadores, aunque se están haciendo esfuerzos para cambiar esta situación. Esto supone un salto muy drástico a temas abstractos sin ningún soporte sobre el que sostenerse. Por esta razón, en la asignatura Introducción a la Arquitectura de Computadores, se plantea hacer una pequeña introducción mediante juegos de los conceptos más básicos para crear ese sustento sobre el que poder construir, de forma atractiva, el resto de conocimientos que se requieren en la asignatura.

El aprendizaje basado en juegos es una metodología interactiva e inmersiva que puede hacer que el proceso de aprendizaje sea más atractivo y agradable para el alumnado, permitiéndole comprometerse activamente con los conceptos que está aprendiendo, en lugar de limitarse a recibir información de forma pasiva. Por esto se pensó que esta metodología, en contraposición a la tradicional, ayudaría al alumnado a asimilar mejor los conceptos básicos.

Se han llevado a cabo dos intervenciones de las que se ha concluido que, siendo el grado de aprendizaje del alumnado similar, el aprendizaje basado en juegos motiva más al alumnado.

Abstract

Most first year students in the Artificial Intelligence degree have had no previous contact with computer architecture, although efforts are being made to change this situation. This is a very drastic leap to abstract topics without any support on which to build, also taking into account that less hours will be devoted to the architecture branch than in Computer Engineering degree. For this reason, this paper proposes a small introduction through games of the most basic concepts to create the

needed support on which to build, in an attractive way, the rest of the knowledge required in the Introduction to Computer Architecture subject.

Game-based learning an interactive and immersive methodology that can make the learning process more attractive and enjoyable for students, allowing them to actively engage with the concepts they are learning, rather than just passively receiving information. For these reasons, it was thought that this methodology would help students to assimilate the basic concepts better than traditional methodologies.

Two interventions have been carried out, from which it has been concluded that, although the level of student learning is similar, game-based learning better motivates students.

Palabras clave

Arquitectura de Computadores, conceptos básicos, aprendizaje basado en el juego.

1. Motivación

Según el Informe del grupo de trabajo SCIE/CODDI sobre la enseñanza pre-universitaria de la Informática [2], el alumnado de primaria debería conocer los componentes de un ordenador y el papel que juegan en su funcionamiento. A su vez el alumnado de secundaria debería ser capaz de diseñar circuitos digitales sencillos con puertas lógicas, debería comprender los componentes hardware y software que forman un ordenador, y cómo se comunican entre sí y con otros ordenadores y cómo se almacenan y ejecutan las instrucciones en un ordenador. Y finalmente, el alumnado de bachillerato de ciencias debería comprender la arquitectura Von Neumann.

El caso es que hoy en día, nuestro alumnado de primer curso de IA no tiene esta base.

La arquitectura de computadores es un tema que al alumnado habitualmente le resulta costoso de asimilar. Por mucho que tengan costumbre de utilizar computadores, existen varios niveles de abstracción que ocultan a las personas usuarias el funcionamiento interno de la máquina. Por eso llegar hasta el fondo de los niveles de abstracción se hace complicado para el alumnado [4].

En el grado de Ingeniería Informática hay varias asignaturas que van impartiendo los conceptos poco a poco y profundizando en cada uno de ellos, desde el punto de vista del diseño de la arquitectura. En cambio, en el grado de IA se pretende que la visión que tiene el alumnado sobre la arquitectura sea mucho más funcional y con conceptos más relacionados con el rendimiento de las máquinas, que utilizarán en IA. Por ello, se pretende que el alumnado afronte conceptos como los de la ejecución de un programa en lenguaje ensamblador, la jerarquía de memoria, la memoria cache, el paralelismo a nivel de instrucción, los procesadores superescalares o los problemas que suponen los sistemas de almacenamiento.

Para ello, se dedicará un tiempo limitado de las horas lectivas a la introducción de los conceptos básicos, mediante juegos.

2. Contextualización y metodología de investigación

El estudio se ha llevado a cabo en dos cursos consecutivos, el curso 2021-2022 y el curso 2022-2023, en la asignatura Introducción a la Arquitectura de Computadores del primer curso del grado de IA, con perspectiva investigación-acción. Analizando los resultados obtenidos el primer curso para actuar en el segundo año en consecuencia.

Se han utilizado para ello las clases, tanto teóricas como prácticas, de la primera semana de curso. Junto con las instrucciones de juego se explican los objetivos de aprendizaje. Al finalizar cada juego también se lleva a cabo una puesta en común sobre la dinámica del juego y los conceptos trabajados para enlazarlos con los objetivos de la asignatura.

La asignatura se oferta en dos grupos, lo que ha dado opción a plantear un diseño cuasiexperimental, con un grupo de control y otro grupo experimental. En cada grupo la asignatura se imparte en distinto idioma pero la persona encargada de la impartición del tema introductorio es la misma en los dos grupos.

El curso 2021-2022 se hizo el seguimiento completo de 23 estudiantes en el grupo experimental y 19 en el grupo de control y el curso 2022-2023 de 19 estudiantes en el grupo experimental y 25 en el de control.

Se han desarrollado procedimientos cuantitativos y

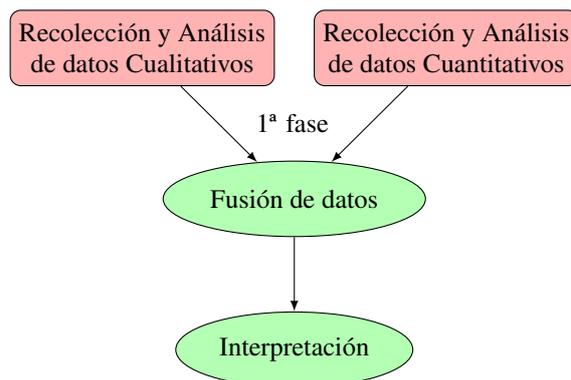


Figura 1: Diseño convergente de Creswell

cualitativos de recogida de información como se indica en la Figura 1, siguiendo un diseño mixto, concretamente el diseño convergente de Creswell [1].

Para la obtención de datos cuantitativos, se preparó un test para cumplimentar antes de la intervención (pre-test) y otro para cumplimentar después (post-test). En ambos se solicitaba al alumnado definir los conceptos que se trabajan en la introducción de la asignatura: programación, estructuras de control, algoritmos, compilación, sistema binario, bit, byte, memoria principal, CPU, unidad de control, periféricos de entrada/salida (E/S), fases de ejecución de las instrucciones. Además en el post-test se pedía valorar los juegos y la metodología. Esta prueba se llevó a cabo tanto con el alumnado del grupo experimental como con el alumnado del grupo de control, de forma que se pudieron llevar a cabo varias comparaciones, como se muestra en la Figura 2. Por un lado se compararon el grupo experimental y el grupo de control entre ellos, antes y después de la intervención, y por otro lado se compararon las mejoras de cada grupo por separado.

Las pruebas fueron evaluadas por separado entre dos docentes que dominan la asignatura pero que no la impartieron en esos cursos. Para ello, en primer lugar, se consensuaron las definiciones correctas de los conceptos en base a la bibliografía temática y valoradas en una escala de 0 a 4. Estos valores numéricos significan 0 - sin respuesta o respuesta nula; 1 - respuesta incorrecta; 2 - tiene alguna idea pero no es correcta; 3 - tiene algún error pero está bastante cerca de la respuesta correcta; y 4 - la respuesta es correcta. A continuación se calculó la media entre los dos resultados. Finalmente, los datos se analizaron con el programa estadístico SPSS (versión 27). Para el análisis de las comparaciones entre grupos se utilizó la distribución *t* de Student (en el caso de comparaciones entre resultados de un mismo grupo "la *t* de Student para grupos relacionados" y en el caso de comparaciones entre grupos diferentes "la *t* de Student para grupos no relacionados").

Para recoger información cualitativa se realizaron

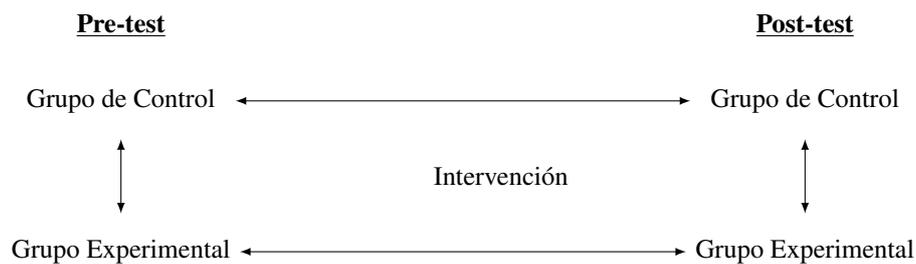


Figura 2: Recogida de datos

grupos de discusión con el alumnado. Estas discusiones fueron grabadas y transcritas con el consiguiente permiso. En ellas se plantearon cuestiones sobre la nueva metodología. El moderador del grupo de discusión, aunque pertenece al grupo de investigación es externo a la asignatura, es decir, no tuvo ninguna relación con el alumnado durante la intervención.

Para analizar toda la información cualitativa se realizó un Análisis del Contenido.

Para extraer conclusiones, se combinaron y se interpretaron los resultados de los análisis cuantitativos y cualitativos.



Figura 3: Piezas para programar laberintos.

3. Juegos propuestos

En este apartado se explicarán los juegos utilizados en las intervenciones. En la siguiente sección se aclarará qué juegos se utilizaron en cada uno de los cursos y las razones por las que se cambiaron algunos de ellos de una intervención a otra.

Las sesiones de aula son de 90 minutos. Al lado de cada juego se ha indicado el tiempo dedicado en clase a cada uno de los juegos, a lo que habría que añadir en 15 y 20 minutos de la puesta en común.

3.1. Programar laberintos (75 min)

Se trata de un juego de retos en el que el alumnado tiene que programar recorridos para llegar al objeto deseado. Este tipo de laberintos puede encontrarse en el sitio web code.org. La diferencia entre los de la web y los aquí planteados es que en este caso, los bloques de programación son físicos y no digitales (ver Figura 3). La observación del uso de dichos bloques nos ha llevado a la conclusión de que el alumnado, al no tener opción de ejecutar el programa, piensa más en la lógica del código que en el simple hecho de si ha funcionado o no.

3.2. La carta perdida (30 min)

Según la Wikipedia¹ un algoritmo es un conjunto de instrucciones o reglas definidas y no ambiguas, ordenadas y finitas que permite, típicamente, solucionar un problema, realizar un cómputo, procesar datos y llevar a cabo otras tareas o actividades. Es un concepto esencial dentro de la IA y aunque con toda seguridad será introducido en las asignaturas de programación, es también importante en esta asignatura.

En este juego el alumnado se divide en grupos de 3 o 4 personas. Cada grupo tiene que jugar con un mazo de cartas del que van a extraer una carta sin saber cuál es. Después tienen que definir las instrucciones a seguir para encontrar la carta perdida. El algoritmo creado por un grupo será utilizado por otro grupo para buscar la carta perdida, analizando las diferencias con su propio algoritmo.

Es un juego muy útil para que el alumnado se dé cuenta de la importancia de escribir instrucciones no ambiguas. Por otro lado, aunque sin entrar demasiado en la ciencia de la algoritmia, también pueden llegar a darse cuenta de que algunos algoritmos son más rápidos que otros, aunque es cierto que al introducir el factor humano en las pruebas no queda demasiado evidente.

¹<https://es.wikipedia.org/wiki/Algoritmo>

3.3. Carrera de dados (30 min)

Este juego está propuesto en code.org². Se sigue trabajando el concepto de algoritmo, y se descubren algunas estructuras de control: bucles for y while y condicionales.

A este juego se puede jugar tanto en grupos pequeños como con toda la clase. Consta de 3 rondas. En la primera ronda cada jugador/a tira el dado y gana quien haya conseguido la puntuación más alta. Se trabaja el concepto de bucle while porque la acción de tirar el dado se repite tantas veces como jugadores/as.

En la segunda ronda cada jugador/a tira el dado tres veces. También gana quien haya conseguido la mayor puntuación, sumando las puntuaciones de las tres tiradas. Se trabaja el concepto de bucle for, repitiendo las tiradas 3 veces.

La última ronda es igual que la segunda pero si alguien obtiene un 1 vuelve a tirar el dado. Se trabaja el concepto de condición.

Además de introducir las tres estructuras de control básicas, se hace hincapié en la necesidad de que las instrucciones no sean ambiguas. Por ejemplo se puede dar el caso de empate, o que en la segunda ronda quienes juegan no sepan si tienen que tirar las tres veces seguidas o por turnos. Son detalles que no se han especificado en las normas y que entre personas se puede decidir en un momento qué hacer pero que un computador no sabría resolver.

3.4. Magia: acierta el día de mi cumpleaños (15 min)

Llega el momento de explicar que por mucho que se siga un buen algoritmo y se escriba un buen programa en lenguaje de alto nivel, ese código no va a ser comprensible para el computador que solo entiende combinaciones de 0s y 1s. Es decir código binario.

Para introducir el sistema binario de representación de números se utiliza un juego de magia³. Una vez realizado el truco de magia se procede a explicar su funcionamiento y a aclarar que funciona porque en el sistema binario, que es un sistema posicional, cada una de las posiciones tiene solo dos valores posibles. Así se relacionan el sistema binario y el decimal, ambos sistemas posicionales, y se explican los patrones que siguen los dos sistemas.

3.5. Cartas binarias (30 min)

Con las cartas binarias se plantea jugar a los seises. La dificultad consiste precisamente en que el número



Figura 4: Cartas binarias.

de las cartas está en binario, como puede verse en la Figura 4. Esto hace que el alumnado vaya interiorizando el valor de los números binarios.

3.6. Juego SUN (75 min)

Una vez que se ha explicado que todo lo que se quiera representar en un computador debe estar en sistema binario, se explica también que para que los programas sean comprensibles deben pasar por un proceso de traducción al que se llama compilación. Dicho proceso tiene dos fases y después de la primera se obtiene el programa traducido al lenguaje ensamblador.

Este juego es el más complejo de todos los juegos que se han utilizado. Se basa en Little Robot Computer⁴ que a su vez se basa en Little Man Computer⁵. Se trata de una variación de estos simuladores. Por un lado, porque se pretende que sea más tangible que un juego digital, basándonos en la afirmación de [3] "Los tangibles permiten vincular formas y acciones físicas con representaciones más simbólicas.", y por otro lado, porque estas dos opciones plantean máquinas en las que el acumulador se reutiliza como registro de entrada de la UAL, en cambio las máquinas que se utilizarán después en la asignatura no son de este tipo.

Los objetivos del juego son tres:

El primero, conocer los elementos básicos de un computador (ver la Figura 5):

- CPU: que consta de la Unidad de Proceso y la Unidad Aritmética Lógica (UAL), que es donde se realizan las operaciones con los datos. También incluye un conjunto de registros donde se almacenan los datos que se están utilizando y la Unidad de Control, que indica a la Unidad de Proceso lo que debe hacer.
- Memoria RAM: donde se guardan las instrucciones del programa y los datos en binario.

²<https://code.org/curriculum/course3/10/Teacher>

³https://www.cse4k12.org/binary/magic_trick.html

⁴<http://casne.ncl.ac.uk/events/conference/session/71/>

⁵https://es.wikipedia.org/wiki/Little_man_computer

- Dispositivos de E/S: mediante los cuales las personas se comunican con el ordenador introduciendo datos o recibiendo el resultado de las operaciones.
- Buses: los cables que comunican los elementos.

El segundo, conocer las fases de ejecución por las que tiene que pasar cada una de las instrucciones de un programa:

- **Búsqueda:** se lee de la memoria el contenido de la posición indicada por el registro PC (Program Counter o Contador de Programa), que corresponde a la instrucción a ejecutar. Además se actualiza el PC para que contenga la dirección de memoria de la siguiente instrucción a ejecutar.
- **Descodificación:** la unidad de control analiza la instrucción y decide qué pasos hay que llevar a cabo en la unidad de proceso.
- **Ejecución:** en la UAL se llevan a cabo los pasos indicados por la unidad de control.

Las instrucciones que recibe el alumnado para empezar a jugar se pueden ver en la Figura 6.

El tercero, entender cómo se ejecutan, paso a paso, las instrucciones y empezar a familiarizarse con el lenguaje ensamblador.

El primer reto que tienen que ejecutar se puede ver en la Figura 7 y el resultado en la Figura 8.

De esta manera con varios retos el alumnado puede entender las fases de ejecución de una instrucción e incluso familiarizarse con las instrucciones del lenguaje ensamblador que posteriormente se usará en la asignatura y entender cómo se programan las estructuras de control que se han aprendido en la carrera de dados.

3.7. Juego MOON (45 min)

MOON es un juego de mesa creado por Pablo Garaizar⁶. El juego se basa en la modificación de los registros A, B, C y D de un computador usando para ello los valores previos de los registros y modificándolos por medio de cartas de operación (lógicas en su mayoría). Es muy interesante para aprender a realizar operaciones lógicas sobre datos binarios Figura 9.

4. Juegos por curso

En el curso 2021-2022 se comenzó programando laberintos, en cambio, en el curso 2022-2023 el juego de los laberintos se sustituyó por la carta perdida y la carrera de dados, no tanto por los resultados de aprendizaje, sino por las características de los mismos. Como primeros juegos a presentar, el reto de la carta perdida y la carrera de dados son opciones más rápidas y que

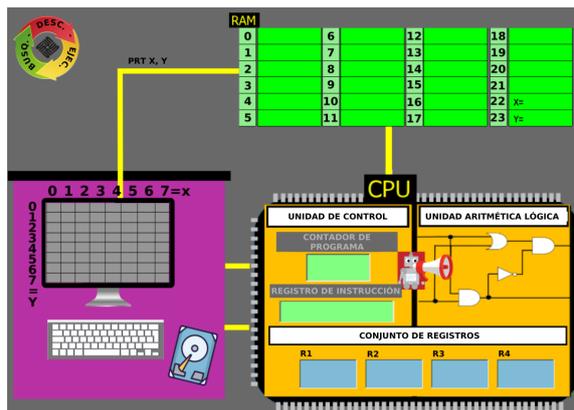


Figura 5: Tablero del juego SUN.

- Copiar las instrucciones en la memoria RAM a partir de la dirección 0.
- Copiar las variables X e Y en la memoria RAM en las últimas posiciones de memoria.
- Poner el valor 0 en el contador de programa (PC).
- Por turnos cada jugador/a llevará a cabo las tres fases de ejecución.

FASES

BÚSQUEDA

- Copiar en el **registro de instrucción (IR)** lo que haya en la dirección de memoria indicada por el **contador de programa (PC)**
- Incrementar el **contador de programa (PC)** (+1).

DESCODIFICACIÓN

- Analizar lo que tiene que hacer la instrucción que se ha guardado en el **registro de instrucción (IR)** e indicar al resto de componentes, **memoria, UAL, registros** etc lo que deben hacer.

EJECUCIÓN

- Ejecutar las decisiones tomadas en la fase de descodificación en **memoria, UAL, registros** etc.



Figura 6: Instrucciones del juego SUN.

```

1. X=2;
   Y=3;

0: LDR r1, X;           //r1=X
1: LDR r2, Y;           //r2=Y
2: MOV R3, #5;          //r3=5
3: BEQ R3, #9;          //si r3=0 PC=9
4: PRT X,Y;             //pintar pos. X,Y
5: ADD r1, r1, #1;      //r1=r1+1
6: STR r1, X;           //X=r1
7: SUB r3, r3, #1;      //r3=r3-1
8: B #3;                //PC=3
9: RET                  //terminar
  
```

Figura 7: Primer programa del juego SUN.

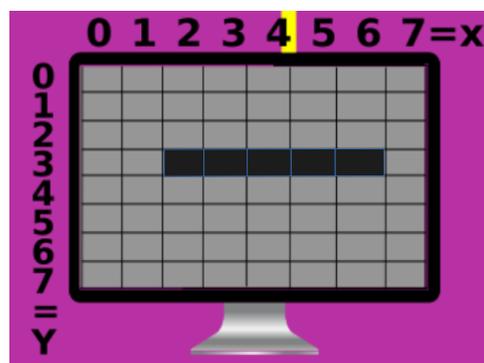


Figura 8: Resultado del primer reto del juego SUN.

⁶<https://compus.deusto.es/es/moon-1110011/>



Figura 9: Resultado del primer reto del juego SUN.

dan más opción a jugar y comentar los objetivos con el grupo completo.

A la hora de trabajar el sistema binario, el curso 2021-2022 utilizaron las cartas binarias para jugar a los seises después de una explicación del sistema binario y a continuación se utilizó el juego MOON. En el análisis sobre el funcionamiento de los juegos se percibió que el juego MOON requiere mucho tiempo de explicación y también mucho tiempo de ejecución, cuando realmente las operaciones lógicas no tienen demasiada relevancia en la asignatura. Por todo ello el curso 2022-2023 se optó por introducir el juego de magia y eliminar los anteriores. Para cursos posteriores se recomienda sumar las cartas binarias a la explicación mediante el juego de magia.

Finalmente, en los dos cursos se ha utilizado el juego SUN que agrupa varios de los conceptos que hay que tratar y ha cumplido bastante bien con su objetivo.

5. Análisis de los resultados

5.1. Análisis cuantitativo

A modo de resumen, el Cuadro 1 muestra las diferencias de los resultados entre el pre-test y el post-test. A continuación se comentan los resultados de cada curso. Hay que tener en cuenta que en ninguno de los cursos ha habido diferencias significativas entre los pre-test de cada grupo. Por lo tanto, podemos pensar que los dos grupos empiezan el experimento en una situación similar.

Primer año (2021-2022)

Los dos grupos han obtenido mejoras significativas en varios items. Independientemente a la metodología docente, ambos grupos han mejorado su conocimiento inicial.

Al comparar los post-tests entre ambos grupos, el grupo experimental ha comprendido mejor las estructuras de control y el sistema binario, mientras que el grupo de control ha comprendido mejor el concepto de memoria principal.

Segundo año (2022-2023)

Tanto el grupo experimental como el de control han obtenido mejoras significativas en casi todos los items. No se aprecian diferencias significativas entre las dos metodologías docentes. Cuatro items han obtenido diferencias significativas entre en grupo experimental y el de control, dos a favor y dos en contra del grupo experimental. En ambos grupos la definición de programación no ha obtenido mejora significativa. Esto puede deberse a que sea el concepto más trabajado en los estudios pre-universitarios.

Una mirada diferente de los datos

También se han realizado otros análisis teniendo en cuenta la valoración de las respuestas. Al tratarse de una intervención de una semana no se espera una definición perfecta de los conceptos, pero sí una mejora parcial. Por ello hemos analizado las respuestas nulas del pre-test, aquellas sin respuesta, lo que hace suponer que no reconocen el concepto y cuántas de ellas han dejado de ser nulas en el post-test.

En el Cuadro 2 se puede observar que en el curso 2021-2022 los items 6 y 11 fueron los que más respuestas nulas recibieron en el pre-test. Es decir, una parte importante del alumnado accede a cursar estudios de IA sin ser capaces de siquiera intentar definir lo que es un bit o lo que son las fases de ejecución de una instrucción. En cambio, al cabo de la intervención todos intentaron definir el bit y sólo dos estudiantes por grupo dejaron de definir las fases de ejecución de las instrucciones.

En el curso 2022-2023 en el pre-test el alumnado se siente incapacitado con más conceptos. En este caso la compilación, bit, unidad de control y fases de ejecución de las instrucciones son los conceptos que más respuestas nulas reciben. Se observa en el cuadro 2 que la mejora en el post-test es sustancial.

Satisfacción del alumnado

En el post-test, además de pedir al alumnado que definiese los conceptos, también se ha pedido valorar su grado de satisfacción con respecto a los juegos y la metodología, mediante una escala Likert de 0 a 5 donde 0 muestra total desacuerdo y 5 total acuerdo. Los Cuadros 3 y 4 muestran para ambos cursos, las respuestas del alumnado del grupo de experimental y las del grupo de control respectivamente.

5.2. Análisis cualitativo

El análisis cualitativo se ha realizado sobre las transcripciones de los grupos de discusión y las respuestas a las preguntas abiertas de los post-test. A continuación se intercalan algunos *testimonios de baja inferencia* extraídos de dicho análisis.

#item	concepto	21-22: mejoras en			22-23: mejoras en		
		gr. ctrl	gr. exp.	exp. vs ctrl.	gr. ctrl	gr. exp.	exp. vs ctrl.
(1)	Programación	-	✓	=	-	-	=
(2)	Estructuras de control	-	✓	+	✓	✓	=
(3)	Algoritmos	✓	✓	=	✓	✓	-
(4)	Compilación	-	-	=	✓	✓	-
(5)	Sistema binario	✓	✓	+	-	✓	=
(6)	Bit	✓	✓	=	✓	✓	=
(7)	Byte	✓	✓	=	✓	✓	=
(8)	Memoria principal	✓	-	-	✓	✓	=
(9)	CPU	✓	-	=	✓	✓	=
(10)	Unidad de control	✓	-	=	✓	✓	+
(11)	Periféricos de E/S	✓	-	=	✓	✓	+
(12)	Fases de ejecución de las instrucciones	-	✓	=	✓	✓	=

Cuadro 1: Resultados de pre- y post-tests. Las dos primeras columnas muestran las mejoras significativas de cada grupo y la tercera columna muestra si hay diferencia significativa a favor del grupo experimental (+) o del grupo de control.

	pre-ctrl	post-ctr.	pre-exp.	post-exp.
2021-2022 (6) Bit	8	0	9	0
2021-2022 (12) Fases de ejecución	14	2	12	2
2022-2023 (4) Compilación	11	0	13	0
2022-2023 (6) Bit	8	0	10	0
2022-2023 (10) Unidad de control	21	6	17	0
2022-2023 (12) Fases de ejecución	10	2	15	2

Cuadro 2: Conceptos que más respuestas nulas recibieron.

Preguntas	2021-22	2022-23
Me ha gustado trabajar los contenidos básicos mediante juegos	4,38	4,50
Trabajando los contenidos básicos con juegos he aprendido más que con explicaciones directas	3,17	3,58
El juego de programar laberintos me ha parecido adecuado para obtener una primera idea sobre la programación	4,38	-
El juego de buscar la carta perdida es adecuado para introducir los algoritmos	-	4,17
La carrera de dados es adecuado para introducir las estructuras de control	-	3,67
El juego de las cartas binarias es adecuado para ejercitar los números binarios	4,13	-
El juego de magia es adecuado para introducir el código binario	-	4,08
El juego MOON es adecuado para hacerse una idea del funcionamiento de las operaciones binarias	4	-
El juego SUN es adecuado para introducir la Arquitectura Von Neumann, las fases de ejecución y el lenguaje ensamblador	3,88	4
Promedio	3,88	4

Cuadro 3: Satisfacción del grupo experimental

Preguntas	2021-22	2022-23
Me ha gustado cómo se han trabajado los conceptos básicos	3,79	2,26
Me habría gustado trabajar estos conceptos mediante juegos	3,79	4,11

Cuadro 4: Satisfacción del grupo de control

El alumnado del grupo experimental se ha mostrado más “motivado”. Las clases son más dinámicas, más divertidas y no les resultan pesadas.

- “Las clases no son tan pesadas y algunos juegos han sido adecuados para introducir algunos temas.”

- “Las clases prácticas han captado nuestra atención haciendo más fácil que nos demos cuenta de cómo funciona.”
- “Al hacerlo mediante juegos hemos podido tomar parte más activa y ayudarnos entre nosotros. De esta manera si alguien no entiende algo el resto le ayuda.”
- “Mediante esta metodología los estudiantes obtenemos más capacidad de representar cómo funcionan distintos sistemas. Y aunque sean cosas de gran complejidad entender de alguna manera los procesos básicos.”

Sin embargo, también hay quien percibe inseguridad ante el cambio metodológico. Tienen la sensación de no haber aprendido lo suficiente. Esto nos lleva a pensar que hay que dar más explicaciones a la hora de proponer los juegos, intentando reforzar el vínculo entre el juego y los contenidos.

- “Es necesaria una mayor profundidad en los juegos para conseguir una visión más específica de esa materia. Cada juego [necesita] una introducción a la materia que se trata.”
- “No se aprende tanto como en una clase convencional. Con los juegos no se pueden aprender los conceptos del todo y no quedan claros.”
- “Juegos complejos para alguien con conocimiento nulo en la materia y poco tiempo para entenderlos bien.”

En cuanto al grupo de control, en los dos cursos han mostrado interés por la nueva metodología:

- “Me hubiera gustado probar metodologías más innovadoras.”
- “El planteamiento de juegos para dinamizar el contenido de la signatura habría facilitado aún más el asentar los contenidos, aunque la presentación de la asignatura está siendo igualmente excelente.”
- “Yo creo que se han explicado de forma correcta los contenidos introductorios, aunque hubiese agradecido aprenderlos de forma más interactiva, utilizando algún juego.”

6. Conclusiones

En general el alumnado ha mostrado su satisfacción con el método utilizado. En ambos grupos, tanto el experimental como el de control, los resultados de aprendizaje del alumnado han sido similares. Podemos concluir entonces que la metodología de aprendizaje mediante juegos es tan válida como la clásica en este contexto.

En cuanto a los cambios que se han hecho en los jue-

gos, se han comparado los valores de significación de los grupos experimentales que comparan el pre-test y el post-test de los dos cursos, donde se observa que los valores dan probabilidades más grandes en el segundo curso, lo que hace concluir que los cambios en los juegos utilizados han sido positivos.

Aún así, hay que tener en cuenta que en algunos casos el alumnado ha manifestado sentir incomodidad y han tenido miedo de haber aprendido menos que en la metodología clásica, sobre todo en el primer curso. Aunque los resultados muestran que se equivocan, sería conveniente eliminar este miedo con algunos cambios en la metodología y enfatizando las explicaciones para que tengan claros los objetivos de aprendizaje y cómo los van logrando. También hay que dejar claro que son conceptos básicos, que se van a seguir trabajando y que lo importante es no llegar a ese momento sin conocerlos.

Agradecimientos

Este trabajo ha sido financiado por el Departamento Educación del Gobierno Vasco (ADIAN, IT1437-22) y el Servicio de Asesoramiento Educativo de la Universidad Del País Vasco / Euskal Herriko Unibertsitatea (IKDi321-18). También queremos agradecer al alumnado del grado de Inteligencia Artificial que ha tomado parte en la iniciativa.

Referencias

- [1] John W Creswell. Revisiting mixed methods and advancing scientific practices. En Hesse-Biber S y Johnson RB (eds), editores, *The Oxford Handbook of Multimethod and Mixed Methods Research Inquiry*, pp. 61–71. Oxford University Press, Oxford, 2015.
- [2] Grupo de trabajo SCIE-CODDII. Informe del grupo de trabajo scie/coddi sobre la enseñanza preuniversitaria de la informática, junio 2018. disponible en <https://coddi.org/wp-content/uploads/2021/07/Informe-SCIE-CODDI-2018-06.pdf>.
- [3] Andrew Manches y Claire O'malley. Tangibles for learning: a representational analysis of physical manipulation. *Personal and Ubiquitous Computing*, 16:405–419, 2012.
- [4] Nathan Thomas, Fiona Carroll, Rita Kop, y S Stocking. ibook learning experience: the challenge of teaching computer architecture to first year university students. *P Worldcomp*, pp. 16–19, 2012. disponible en <http://worldcomp-proceedings.com/proc/p2012/EEE2439.pdf>.