

Conceptos básicos de Arquitectura de Computadores mediante juegos y desarrollo de juegos

Edurne Larraza-Mendiluze, Olatz Arbelaitz, Olatz Arregi, Jose Ignacio Martín, and Jose Francisco Lukas

Abstract—Objectives: The aim of this research is to study how games and game development can help students learn introductory concepts of a very abstract topic such as Computer Architecture.

Methodology: In a quasi-experimental scenario, quantitative and qualitative data has been collected from students before and after the intervention throughout three consecutive school years. Following the action-research methodology, year by year several changes have been implemented in the games, pursuing a better understanding of the concepts. Also during the last year a game development has been added in order to consolidate knowledge.

Results: No statistical differences were found between the knowledge acquired by students in the control and experimental groups during the first two years. Therefore, and given that motivation was higher when using games, the same game-based methodology was used in the third year in both groups. Afterwards, the students had to develop an interactive presentation in the form of an Escape Room to teach younger students the concepts they had learned. Only a few concepts gained knowledge after this intervention, but those concepts with less prior understanding did.

Conclusions: Although advanced computer architecture concepts can be difficult to handle through games, there are basic concepts that can be worked on in this way. In addition, the excitement and motivation provided by games make a good introduction to the subject. Also the development of simple games helps to understand some of the concepts that were not well understood before.

Index Terms—Computer Architecture, basic concepts, Game Based Learning, Game Development Based Learning.

I. INTRODUCCIÓN

EN julio de 2023 se presentó en JENUI el artículo *Juegos para presentar conceptos básicos de Arquitectura de Computadores* [1] que fue elegido como uno de los mejores del congreso y propuesto para ser enviado a IEEE-RITA. Este artículo, por tanto, es una extensión del anterior que muestra los resultados obtenidos durante un curso más en el que se ha aplicado la experiencia. Además de un nuevo ajuste en los juegos propuestos, se plantea otra intervención mediante el desarrollo de juegos en grupo para afianzar los conceptos aprendidos.

Edurne Larraza-Mendiluze, Olatz Arbelaitz, Olatz Arregi y Jose Ignacio Martín, Dpto. de Arquitectura y Tecnología de Computadores, Universidad del País Vasco (UPV/EHU) (e-mail: edurne.larraza@ehu.eus).

Jose Francisco Lukas, Dpto. de Métodos de Investigación y Diagnóstico en Educación, Universidad del País Vasco (UPV/EHU)

Manuscript received —, —; revised —, —.

DOI (Digital Object Identifier) Pendiente

La experiencia se plantea porque el alumnado de primer curso del grado de Inteligencia Artificial carece de los conocimientos básicos de arquitectura de computadores que según el Informe del grupo de trabajo SCIE/CODDI sobre la enseñanza preuniversitaria de la Informática [2] debería tener, como son entre otros: en primaria conocer los componentes de un ordenador y el papel que juegan en su funcionamiento; en secundaria diseñar circuitos digitales sencillos con puertas lógicas, 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 en bachillerato de ciencias comprender la arquitectura Von Neumann.

La arquitectura de computadores es un tema que al alumnado habitualmente le resulta costoso de asimilar [3]. Por mucho que tengan la 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, más aún cuando no hay ningún conocimiento previo.

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 Inteligencia Artificial (IA) se pretende que la visión del alumnado sobre la arquitectura sea mucho más funcional y con conceptos más relacionados con el rendimiento de las máquinas que utilizarán en el futuro. 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 caché, el paralelismo a nivel de instrucción (ILP) o los problemas que suponen los sistemas de almacenamiento.

Con el objetivo de reducir la brecha de conocimiento inicial, en esta experiencia se plantea dedicar un tiempo limitado de las horas lectivas a la introducción de los conceptos básicos, mediante juegos. En la literatura se pueden encontrar infinidad de referencias sobre el aprendizaje basado en el juego, no solo en el juego digital [4], sino también en analógico [5]. Todos ellos hacen referencia a la motivación, que es uno de los elementos que se van a analizar en este trabajo.

Además, aunque con menor profundidad puesto que es la experiencia de un solo año, también se propone trabajar el aprendizaje basado en el desarrollo de juegos. Ya en el año 2006 El-Nasr y Smith [6] empiezan a hablar de las posibilidades que ofrecen los entornos de desarrollo de

videojuegos, puesto que permiten no tener que diseñar los juegos desde cero sino modificarlos y nos sitúan frente al construccionismo de Seymour Papert [7], quien defiende que construyendo es como realmente se aprende.

A partir de este punto, en el apartado II se pretende relacionar este trabajo con otros llevados a cabo con metodología de aprendizaje basado en el juego y aprendizaje basado en el desarrollo de juegos, en torno a la arquitectura de computadores. El apartado III, presenta la metodología utilizada en la investigación mientras que los juegos utilizados en las diferentes intervenciones se relacionan en los apartados IV y V. El apartado VI analiza los resultados obtenidos en los experimentos realizados, y para finalizar, en el apartado VII se muestran las conclusiones obtenidas y las líneas en las que se quiere seguir trabajando.

II. TRABAJOS RELACIONADOS

Un artículo publicado en 2016 [8] hace una revisión sistemática de juegos utilizados en educación superior para enseñar informática. De los 107 artículos que finalmente analiza, solo tres hablan sobre arquitectura de computadores (AC) y el artículo menciona explícitamente que gran parte del material encontrado en el área de AC son simulaciones más que juegos y por eso no los han incluido en el estudio.

De los tres artículos mencionados, los dos primeros no describen los juegos, se centran en la metodología, bien docente [9], o de diseño de juegos serios [10]. La última referencia [11] es a una página web en la que se plantea una simulación viva, donde el papel de los componentes lo realiza el alumnado. La propuesta es para que alumnado tan joven como de 4-6 años entienda el funcionamiento del computador y sus periféricos.

Otra simulación viva, pero con conceptos más avanzados se plantea en [12]. Algo parecido a los simuladores de la máquina Little Man Computer [13][14][15].

Por otro lado, el juego Moon, que se presenta en el apartado III y que ha sido analizado desde el punto de vista del desarrollo del pensamiento computacional en [16], puede utilizarse para introducir la lógica booleana.

Se han encontrado tres propuestas más [17][18][19] que utilizan juegos digitales. Las dos primeras no explican cuáles son los conceptos de arquitectura de computadores tratados en el juego. La última opción trabaja las fases de ejecución y los pasos a llevar a cabo en cada una de ellas, introduciendo también la fase de interrupción.

En cuanto al desarrollo de juegos como metodología de aprendizaje en la rama de arquitectura, se han encontrado varias propuestas, la mayoría para trabajar la programación en ensamblador [20][21][22][23] y uno para trabajar la Entrada/Salida en lenguaje C a bajo nivel [24].

Esta lista de referencias demuestra que es un campo de interés pero que todavía hay mucho que hacer en él. El artículo que presentamos aquí quiere abrir camino en este campo de investigación.

III. CONTEXTUALIZACIÓN Y METODOLOGÍA DE INVESTIGACIÓN

El estudio se ha llevado a cabo en tres cursos consecutivos, desde el curso 2021-2022 hasta el curso 2023-2024, en la asignatura Introducción a la Arquitectura de Computadores del primer curso del grado de IA, con perspectiva investigación-acción. Es decir, analizando los resultados obtenidos en cada curso para actuar en consecuencia el curso siguiente.

En los dos primeros cursos se han utilizado para ello las clases, tanto teóricas como prácticas, de la primera semana del curso, 4,5 horas de clase. 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 cuasi-experimental (los grupos no han sido elegidos al azar), con un grupo control y otro grupo experimental. En cada grupo la asignatura se imparte en distinto idioma, pero la persona que imparte el tema introductorio es la misma.

Para la obtención de datos cuantitativos, se preparó un test para cumplimentar antes de la intervención (pretest) y otro para cumplimentar después (postest). 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, y fases de ejecución de las instrucciones.

El pretest realizado ha demostrado que en todos los casos los dos grupos (experimental y control) fueran equivalentes, es decir, que ninguno de los dos grupos empezaba con conocimientos superiores al otro grupo. Posteriormente, se ha utilizado la nueva metodología, utilizando los juegos de mesa que se explicarán en el apartado IV, en el grupo experimental mientras que en el grupo control se ha seguido utilizando la metodología habitual, basada en explicaciones y ejercicios.

En el postest se ha comprobado si tanto en el grupo control como en el experimental ha habido mejoras desde el pretest. También ha podido comprobarse si se han dado diferencias significativas entre el grupo control y el grupo experimental.

El curso 2021-2022 se hizo el seguimiento completo de 23 estudiantes en el grupo experimental y 19 en el grupo control y el curso 2022-2023 de 19 estudiantes en el grupo experimental y 25 en el grupo control. El último curso, aunque el alumnado sigue estando dividido en dos grupos, los datos se han analizado conjuntamente, puesto que en los dos grupos se ha utilizado la misma metodología. Se han analizado las respuestas de 50 estudiantes en total.

Se han desarrollado procedimientos cuantitativos y cualitativos de recogida de información, siguiendo un diseño mixto, concretamente el diseño convergente de Creswell [25] (Figura 1).

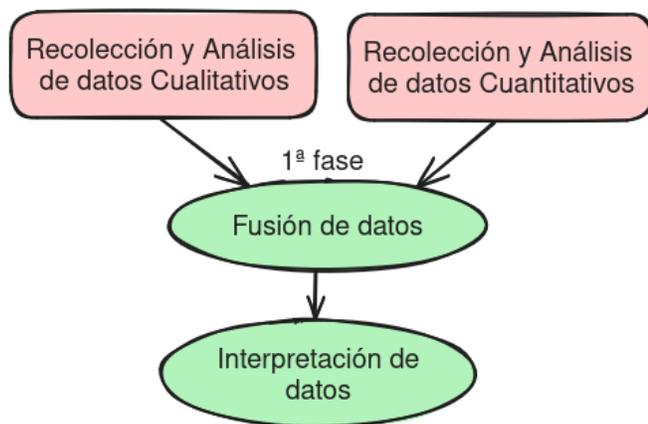


Figura 1: Diseño convergente de Creswell

Además, en el postest, también se pedía valorar los juegos y la metodología. En los dos primeros cursos, esta prueba se llevó a cabo tanto con el alumnado del grupo experimental como con el alumnado del grupo control, de forma que se pudieron llevar a cabo varias comparaciones. Por un lado, se compararon el grupo experimental y el grupo control entre ellos, antes y después de la intervención, y por otro lado se compararon las mejoras de cada grupo por separado. El último curso solo se compararon los resultados del pretest con los del postest, como si el alumnado hubiera estado en el mismo grupo, puesto que la metodología utilizada en los dos grupos fue la misma.

Las definiciones dadas por el estudiantado tanto en los pretests como en los postests 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 se valoraron las respuestas en una escala de 0 a 4. Estos valores numéricos significan 0 - sin respuesta; 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”).

El primer curso se realizaron grupos de discusión con el alumnado con el objetivo de recabar información cualitativa. 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. También se hicieron varias preguntas abiertas en el postest. El resultado de la evaluación de los textos obtenidos en los grupos de discusión y en las preguntas abiertas del postest no variaron

demasiado, por lo que en los cursos posteriores se utilizaron únicamente las preguntas abiertas del postest.

Para analizar la información cualitativa se realizó un análisis comprensivo [26] de los textos recogidos en los grupos de discusión.

El último curso se llevó a cabo una acción más, en la que una vez acabada la semana introductoria, el alumnado desarrolló, en grupos de 4 personas, un escape room digital. Con el objetivo de lograr una mayor motivación del alumnado, se les dijo que los escape room se utilizarían para enseñar los conceptos aprendidos a alumnado de secundaria. Debido al poco tiempo del que dispone la asignatura para esta actividad, no fue posible la coordinación con centros y no se llegaron a probar con alumnado de secundaria, aunque está en mente de las autoras y autores, revisar los escape room para corregir los posibles errores y ponerlos a disposición del profesorado de secundaria.

Los escape room se desarrollaron en Genially [27], sin necesidad de código de programación pero donde aparecen conceptos de programación como secuenciación (de las imágenes) y saltos (para la interactividad). El diseño de esta actividad se basó en el modelo de aprendizaje basado en el desarrollo de juegos. Después de desarrollar los escape room, actividad en la que han tomado parte todos y todas las estudiantes que han participado en el estudio, se ha repetido el postest para observar diferencias con respecto al conocimiento anterior, es decir, el conocimiento que tenían al realizar el primer postest.

Para extraer conclusiones, se combinaron y se interpretaron los resultados de los análisis cuantitativos y cualitativos, cuyo resultado se muestra en la sección V.

IV. 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 que se le ha dedicado en clase, a lo que habría que añadir en torno a 15 o 20 minutos de la puesta en común.

A. 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 pueden 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 (Figura 2). 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.



Figura 2: Piezas para programar los laberintos.

B. La carta perdida (30 min)

Según la RAE [28] un algoritmo es un conjunto ordenado y finito de operaciones que permite hallar la solución de un problema. 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.

Este juego se juega con una baraja cualquiera que tenga varios palos, el mismo número de cartas por palo y las cartas de cada palo tengan un orden.

El alumnado se divide en grupos de 3 o 4 personas. Cada grupo tiene que jugar con una baraja de cartas de la que van a extraer una carta sin saber cuál es. Una vez extraída la carta de la baraja tienen que definir las instrucciones a seguir para averiguar cuál es la carta perdida. El alumnado conoce la composición de la baraja, la única carta que no pueden ver es la carta que se ha extraído y tienen acceso a todas las cartas de forma aleatoria. Precisamente es el alumnado quien tiene que establecer el algoritmo para buscar la carta perdida de la forma más rápida posible. 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 algoritmia, también pueden llegar a darse cuenta de que algunos algoritmos son más rápidos que otros.

C. Carrera de dados (30 min)

Este juego está propuesto en code.org [29]. Se sigue trabajando el concepto de algoritmo, y se descubren algunas estructuras de control: bucles for y while y condicionales.

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 mientras haya 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.

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

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 magia [30]. 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.

E. Cartas binarias (30 min)

Con las cartas binarias se plantea jugar a los seises, aunque en este caso la carta que empieza el juego es el 0100 con signo OR. La dificultad consiste precisamente en que el valor de las cartas está en binario (Figura 3). Aunque aparecen 4 bits, cada palo tiene valores entre 0 y 7. Esto hace que el alumnado vaya interiorizando los valores de la representación binaria. Cualquier juego en el que los números de las cartas se usen consecutivamente sería válido.



Figura 3: Cartas binarias

F. Juego SUN (75 min)

Este juego es el más complejo de todos los juegos que se han utilizado. Se basa en *Little Robot Computer* [31] que a su vez se basa en *Little Man Computer* [13][14]. Se trata de una variación de estos simuladores. Por un lado, porque se pretende que sea más tangible que un juego digital, basándonos en investigaciones como [32] en las que se trabaja la idea de que 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 Unidad Aritmético Lógica (UAL), en cambio las máquinas que se utilizarán después en la asignatura no son de este tipo.

Los objetivos del juego son cuatro:

1.- Conocer los elementos básicos de un computador mediante una simplificación (Figura 4).

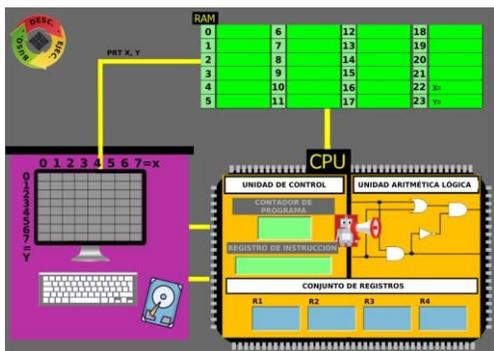


Figura 4: Tablero del juego SUN

2.- Conocer las fases de ejecución por las que tiene que pasar cada una de las instrucciones de un programa: búsqueda, decodificación y ejecución. Aquí ya puede hablarse también de lectura de operandos, ejecución y escritura de resultados.

3.- Entender cómo se ejecutan, paso a paso, las instrucciones.

4.- Familiarizarse con el lenguaje ensamblador.

En el primer reto que tienen que ejecutar (Figura 5), por un lado, las variables x e y toman los valores 2 y 3. Por otro lado, un registro (r3), el que se utilizará para hacer el control del bucle, toma el valor 5 y se va decrementando de uno en uno hasta que tenga el valor 0, que será cuando se salga del bucle. En esas cinco iteraciones del bucle se pinta en pantalla la posición (x, y) y se incrementa el valor de x. Por tanto, se pintan, en la fila y=3 las posiciones x=2 hasta x=6 (Figura 6).

```

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 6: Resultado

Figura 5: Primer programa del juego SUN

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 transforman a ensamblador las estructuras de control que se han aprendido en el juego “Carrera de dados”.

G. Juego MOON (45 min)

MOON es un juego de mesa creado por Pablo Garaizar [33]. El juego consta de 4 registros de 4 bits sobre los cuales hay que ir haciendo operaciones aritméticas y lógicas para conseguir los valores que aparecen en las cartas objetivo. Cada operación tiene un coste energético diferente y si el objetivo no se consigue con la energía de cada jugador/a hay que añadir una carta objetivo más a la pila. Si la pila de

cartas objetivo llega al máximo, la operación lunar no se ha podido llevar a cabo adecuadamente. Por el contrario, si se consiguen todos los objetivos, la misión se habrá llevado a cabo con éxito (Figura 7).



Figura 7: Tablero del juego MOON

H. Juego 1-2-3 (15 min)

Mediante este juego se pretende que el alumnado entienda que todas las instrucciones que se ejecutan en una máquina tienen que pasar por unas fases concretas y que dichas fases están temporizadas. Así, cada estudiante recibe un mazo de cartas binarias (Figura 4) en las que los palos son las puertas lógicas AND y OR (negras), XOR y NOT (rojas). Antes de empezar, la profesora indica los grupos de cartas que se deben formar, por ejemplo, se deben diferenciar las cartas rojas y las AND, el resto irán a un tercer grupo. Después, la profesora irá diciendo 1, 2 y 3, de forma sincronizada y el alumnado a la de 1, debe coger una carta del montón, sin mirarla (búsqueda), a la de 2 debe mirar la carta (descodificación) y a la de 3 debe colocarla en un montón según se ha indicado anteriormente (ejecución). Así, el alumnado aprende a distinguir las fases básicas de ejecución de las instrucciones: Búsqueda, Descodificación y Ejecución. Más adelante en la asignatura se trabajarán todas las fases de ejecución con mayor detalle. En este juego solo se pretende hacer una introducción al proceso de ejecución por fases.

V. JUEGOS POR CURSO

En esta sección inicialmente se indican los juegos que se han utilizado en cada curso académico. A continuación, se explican las razones por las que, de un curso a otro, se han eliminado algunos juegos y se han introducido otros.

TABLA I

SELECCIÓN DE JUEGOS EN CADA CURSO								
	A)	B)	C)	D)	E)	F)	G)	H)
2021-2022	X				X	X	X	
2022-2023		X	X	X		X		
2023-2024		X	X	X	X			X

El juego de programar laberintos que se utilizó en el curso 2021-2022 se sustituyó en el curso 2022-2023 por la carta perdida y la carrera de dados, no tanto por los resultados de aprendizaje obtenidos en el curso anterior, 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 de ejecución más rápida, por lo que 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 se 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, que es lo que más se trabaja en este juego, no tienen demasiada relevancia en la asignatura. Por todo ello en el curso 2022-2023 se optó por introducir el juego de magia y eliminar los anteriores. En cambio, en la experiencia se percibió una falta de asimilación del sistema binario, por lo que en el curso 2023-2024 se ha vuelto a añadir el juego las cartas binarias después de la explicación dada mediante el juego de magia.

El juego SUN, que agrupa varios de los conceptos que hay que tratar, se ha utilizado durante los dos primeros cursos, y, aunque ha cumplido bastante bien con su objetivo, se ha percibido que el tiempo que requiere es demasiado, teniendo en cuenta que es un tema al que después se le va a dedicar bastante tiempo, puesto que el lenguaje ensamblador se sigue trabajando durante el curso. Por eso, para el tercer curso se ha propuesto el juego 1-2-3 que es mucho más rápido y así da más tiempo para comentar la importancia de las fases de ejecución en la ejecución de un programa.

VI. ANÁLISIS DE LOS RESULTADOS

A. Análisis cuantitativo

A modo de resumen, las TABLAS II y III muestran las diferencias de los resultados entre el pretest y el postest. 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 pretest de cada grupo. Por 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 ítems. Independientemente de la metodología docente utilizada, ambos grupos han mejorado su conocimiento inicial. Al comparar los postests entre ambos grupos, el grupo experimental ha comprendido mejor las estructuras de control y el sistema binario, mientras que el grupo control ha comprendido mejor el concepto de memoria principal.

TABLA II

RESULTADOS DE LOS CURSOS 21-22 Y 22-23

item concepto	21-22			22-23		
	mejoras en:			mejoras en:		
	Ctrl.	Exp.	Exp. vs Ctrl.	Ctrl.	Exp.	Exp. vs Ctrl.
(1) Programación	X	✓	=	X	X	=

(2) Estructuras de control	X	✓	+	✓	✓	=
(3) Algoritmos	✓	✓	=	✓	✓	-
(4) Compilación	X	X	=	✓	✓	-
(5) Sistema binario	✓	✓	+	X	✓	=
(6) Bit	✓	✓	=	✓	✓	=
(7) Byte	✓	✓	=	✓	✓	=
(8) Memoria principal	✓	X	-	✓	✓	=
(9) CPU	✓	X	=	✓	✓	=
(10) Unidad de control	✓	X	=	✓	✓	+
(11) Fases de ejecución de las instr.	X	✓	=	✓	✓	=

(✓) Mejoras en postest, (X) No mejoras en postest. Gr. Exp igual (=), mejor (+), peor (-) que gr. Ctrl.

TABLA III

RESULTADOS DEL PRIMER Y SEGUNDO POSTEST EN EL CURSO 23-24

item concepto	Primer postest	Segundo postest
(1) Programación	✓	X
(2) Estructuras de control	✓	X
(3) Algoritmos	✓	X
(4) Compilación	✓	X
(5) Sistema binario	✓	X
(6) Bit	✓	✓
(7) Byte	✓	✓
(8) Memoria principal	✓	X
(9) CPU	✓	X
(10) Unidad de control	✓	✓
(11) Fases de ejecución de las instr.	✓	X

(✓) Mejoras en postest, (X) No mejoras en postest.

Segundo año (2022-2023)

Tanto el grupo experimental como el grupo control han obtenido mejoras significativas en casi todos los ítems. Cuatro ítems han obtenido diferencias significativas entre el grupo experimental y el grupo control, dos a favor y dos en contra del grupo experimental. En ambos grupos la definición de programación no ha obtenido una mejora significativa. Esto puede deberse a que sea el concepto más trabajado en los estudios preuniversitarios.

Tercer año (2023-2024)

Se han analizado los dos grupos conjuntamente puesto que ambos han utilizado la misma metodología. En todos los ítems se han encontrado diferencias significativas entre el pretest y el postest. Este resultado confirma la valía de la metodología utilizada y refuerza los resultados de los cursos anteriores.

En el segundo postest solo se han percibido mejoras significativas en tres ítems, bit, byte y unidad de control. Son los ítems que en el primer postest habían mantenido alto el valor de respuestas nulas y bajo el valor de respuestas correctas.

Una mirada diferente de los datos

También se han realizado otros análisis teniendo en cuenta la puntuación 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 eso, se han analizado las respuestas nulas del pretest, aquellas sin respuesta, lo que hace suponer que no reconocen

el concepto y aquellas que han obtenido una respuesta totalmente incorrecta, lo que demuestra que, aunque piensan que algo saben sobre el tema, no es cierto. Después se han analizado cuántas de ellas han dejado de ser nulas en el postest.

Figuras 8 a 10 muestran las respuestas nulas de los cursos 21-22 a 23-24. Las TABLAS II y III llevan otro tratamiento estadístico, y solo muestran si la diferencia entre pretest y postest es estadísticamente significativa. En estas gráficas en cambio se puede ver qué porcentaje del alumnado no ha definido o ha definido mal cada uno de los conceptos y se ve claramente, que, aunque en algunos casos la diferencia no sea significativa, en todos los casos ha mejorado e incluso en algunos de los ítems la barra llega a desaparecer.

La Figura 10 muestra la mejora del pretest al primer postest e incluso del primer postest al segundo postest. Aunque la mayoría de conceptos han mejorado mucho en la primera fase y no tienen mucho margen en la segunda, se observa claramente que los conceptos con margen de mejora, compilación, bit, byte e unidad de control, mejoran bastante en el segundo postest, después de haber creado un juego con el objetivo de enseñar los conceptos aprendidos.

Por otro lado, aunque ya se ha mencionado que en solo una semana no se espera que el alumnado dé una definición totalmente correcta de los conceptos, también se ha analizado el número de respuestas correctas (Figuras 11 a 13).

En este caso se ve que hay conceptos que pocos o pocas estudiantes llegan a entender a la perfección, aunque en todos los conceptos el número de respuestas correctas mejora.

En el último curso, después del segundo postest, en algunos de los conceptos el porcentaje de respuestas correctas incrementa bastante, por ejemplo, en el caso de los términos bit y byte, y en cambio en otros decrece, por ejemplo, en el caso de los términos memoria principal y CPU. Aunque es difícil dar con el porqué de esas diferencias, analizando los escape room realizados se puede ver que algunos términos se han utilizado más que otros, porque dan más juego a la hora de plantear retos, lo que puede hacer que mientras trabajan el escape room se dediquen más a unos conceptos que a otros, lo que podría explicar la diferencia en la capacidad para definir dichos conceptos.

La TABLA IV muestra los mínimos, máximos y medias de respuestas nulas y correctas para los cursos 21-22 y 22-23 y la TABLA V para el curso 23-24.

Satisfacción del alumnado

En los postests, 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 1 a 5, donde 1 muestra total desacuerdo y 5 total acuerdo. Las TABLAS VI a VIII muestran, para los tres cursos, las respuestas del alumnado del grupo de experimental, las del grupo control y las del segundo postest del último curso respectivamente.

B. 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 postests. A continuación, se intercalan con una pequeña explicación, algunos testimonios de baja inferencia extraídos de dicho análisis.

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 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 asignatura habría facilitado aún más el asentarlos 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."

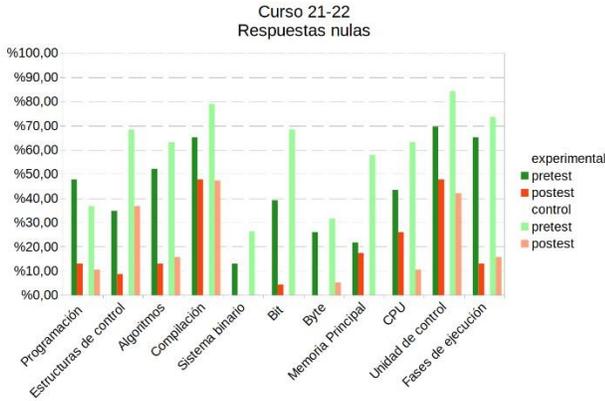


Figura 8: Respuestas nulas el curso 21-22

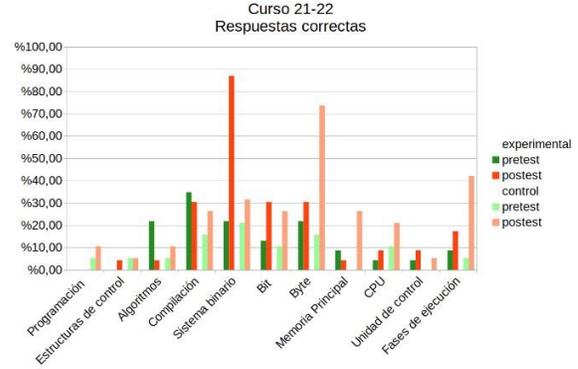


Figura 11: Respuestas correctas el curso 21-22

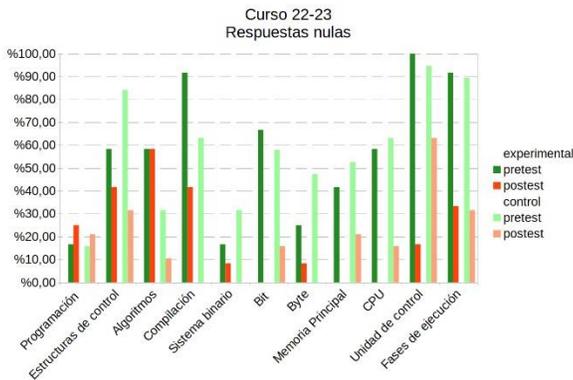


Figura 9: Respuestas nulas el curso 22-23

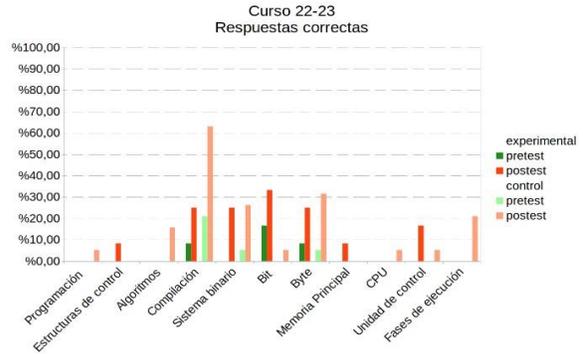


Figura 12: Respuestas correctas el curso 22-23

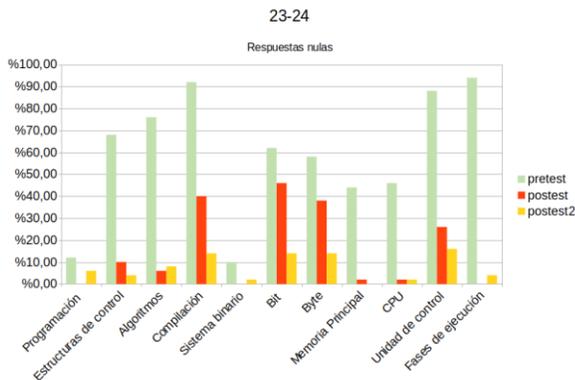


Figura 10: Respuestas nulas el curso 23-24

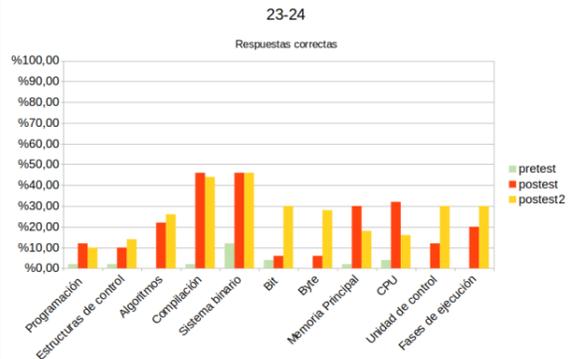


Figura 13: Respuestas correctas el curso 23-24

La percepción del último curso no ha variado demasiado con respecto a las intervenciones anteriores. El alumnado se siente más motivado en general, como muestran los siguientes testimonios:

- “Es más interactivo que otros métodos y es más fácil no distraerse.”
- “Resulta más liviana que la realización de una clase teórica común y hace que el alumno participe más activamente en el proceso de aprendizaje y profundización de conceptos.”
- “Al utilizar juegos entiendes los contenidos de manera natural y es más fácil mantener la atención.”

Pero también critican que se le dedique tan poco tiempo. Tienen la sensación de que no llegan a asimilar los conceptos y de que es fácil distraerse con el material lúdico, o temen no haber hecho la analogía correcta con respecto a la realidad.

- “Me gustaría tener más tiempo para apuntar las cosas en el cuaderno.”
- “Puede llevar a distracción tener encima de la mesa cartas/dados y llevarte a hacer otra cosa cuando deberías estar atendiendo.”
- “En ocasiones las analogías de los juegos están demasiado abstractas y, aunque en su lógica fundamental sí coinciden, me ha costado imaginarme cómo se

trasladan esos conceptos, en la práctica, al funcionamiento de un computador.”

TABLA IV

		Experimental		Control	
		Pretest	Postest	Pretest	postest
Nulas 21-22	Máx.	69,57%	47,83%	84,21%	47,37%
	Mín.	13,04%	0,00%	26,32%	0,00%
	Media	43,48%	17,39%	59,33%	16,75%
Correctas 21-22	Máx.	34,78%	86,96%	21,05%	73,68%
	Mín.	0,00%	0,00%	0,00%	5,26%
	Media	13,65%	20,55%	8,61%	25,36%
Nulas 22-23	Máx.	96,74%	63,16%	100,00%	58,33%
	Mín.	15,79%	0,00%	16,67%	0,00%
	Media	57,42%	19,14%	56,82%	21,21%
Correctas 22-23	Máx.	21,05%	63,16%	16,67%	33,33%
	Mín.	0,00%	0,00%	0,00%	0,00%
	Media	2,87%	16,27%	3,03%	12,88%

TABLA V

		Pretest	Postest1	Postest2
Nulas 23-24	Máx.	94,00%	46,00%	46,00%
	Mín.	0,00%	6,00%	10,00%
	Media	2,55%	22,00%	26,00%
Correctas 23-24	Máx.	12,00%	46,00%	46,00%
	Mín.	0,00%	6,00%	10,00%
	Media	2,55%	22,00%	26,55%

TABLA VI

SATISFACCIÓN GRUPO EXPERIMENTAL

Preguntas	21-22	22-23	23-24
Me ha gustado trabajar los contenidos básicos mediante juegos	4,38	4,50	4,80
Trabajando los contenidos básicos con juegos he aprendido más que con explicaciones directas	3,17	3,58	3,92
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	4,38
La carrera de dados es adecuada para introducir las estructuras de control	-	3,67	4,29
El juego de las cartas binarias es adecuado para ejercitar los números binarios	4,13	-	4,63
El juego de magia es adecuado para introducir el código binario	-	4,08	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	-
El juego 1-2-3 es adecuado para introducir las fases de ejecución	-	-	3,98
Promedio	3,88	4	4,26

TABLA VII

SATISFACCIÓN GRUPO CONTROL

Preguntas	21-22	22-23
Me ha gustado cómo se han trabajado los conceptos básicos	4,38	4,50
Me habría gustado trabajar estos conceptos mediante juegos	3,88	4

TABLA VIII

SATISFACCIÓN ESCAPE ROOM

Preguntas	23-24
Me ha gustado el hecho de desarrollar un juego para enseñar los conceptos introductorios	3,47
Saber que el trabajo se utilizará en secundaria me ha ayudado a esforzarme más	3,51

VII. CONCLUSIONES

El objetivo principal de este trabajo era el de analizar la posibilidad de presentar de forma rápida los conceptos básicos de arquitectura de computadores mediante juegos. En el artículo se presentan los juegos que se han utilizado en varias intervenciones. El análisis de los datos obtenidos durante dos cursos ha demostrado que, aunque el alumnado puede mostrarse inseguro ante la novedad, los resultados de aprendizaje son comparables con el método tradicional, con la mejora que supone la motivación del alumnado. Así pues, en el último curso la experiencia se ha llevado a cabo por igual con todo el alumnado, añadiendo el desarrollo de un juego tipo Escape Room para enseñar lo aprendido a alumnado más joven. Esta última intervención, aunque requiere de más ciclos para obtener datos concluyentes, muestra unos datos iniciales que apoyan su uso, puesto que mejoran los datos de los conceptos que después de los juegos quedaban más débiles. Por otro lado, es de mencionar, que la búsqueda bibliográfica realizada para este trabajo ha demostrado que el campo de la enseñanza de la arquitectura de computadores mediante juegos y mediante el desarrollo de juegos tiene todavía mucho que tratar y que el equipo de esta asignatura piensa seguir trabajando estos dos temas, proponiendo nuevos juegos y actividades de desarrollo en temas más avanzados.

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.

REFERENCES

- [1] E. Larraza Mendiluze, O. Arbelaitz Gallego, O. Arregi Uriarte, J. I. Martín Aramburu, and J. F. Lukas Mujika, "Juegos para presentar conceptos básicos de arquitectura de computadores," in *Actas de las Jornadas sobre la Enseñanza Universitaria de la Informática (JENUI)*, no. 8. Asociación de Enseñantes Universitarios de la Informática. AENUI, 2023, pp. 49–56.

- [2] G. 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/InformeSCIE-CODDI-2018-06.pdf>.
- [3] N. Thomas, F. Carroll, R. Kop, and 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>.
- [4] S. Tobias, J. D. Fletcher, and A. P. Wind, "Game-based learning," *Handbook of research on educational communications and technology*, pp. 485–503, 2014.
- [5] A. I. Abdul Jabbar and P. Felicia, "Gameplay engagement and learning in game-based learning: A systematic review," *Review of educational research*, vol. 85, no. 4, pp. 740–779, 2015.
- [6] M. S. El-Nasr and B. K. Smith, "Learning through game modding," *Computers in Entertainment (CIE)*, vol. 4, no. 1, pp. 7–es, 2006.
- [7] S. Papert, "Children, computers and powerful ideas," *New York: Basic Books*, vol. 10, no. 1990, p. 1095592, 1990.
- [8] P. Battistella, & C. G. von Wangenheim (2016). "Games for teaching computing in higher education—a systematic review." *IEEE Technology and Engineering Education*, 9(1), 8-30.
- [9] K. Kuk, D. Rancic, P. Spalevic, Z. Trajceviski, & M. Micalovic, (2012). "Use game based interactive multimedia modules to learning basic concepts on courses for computing science." *Przeglad Elektrotechniczny*, 88, 150-153.
- [10] J. Melero, D. Hernández-Leo, & J. Blat. (2012, July). "Considerations for the design of mini-games integrating hints for puzzle solving ICTrelated concepts." In *2012 IEEE 12th International Conference on Advanced Learning Technologies* (pp. 154-158). IEEE.
- [11] Workshop: Simulate Computer, Available: <https://sites.google.com/site/childrenandtechnology/workshop-simulatecomputer?authuser=0>.
- [12] G. Kacmarcik. (2007). "How Computers Work", Available: https://www.cse4k12.org/how_computers_work/
- [13] P. Higginson. "Little Man Computer", Available: https://www.peterhigginson.co.uk/lmc/?F5=16-Apr-24_08:58:02
- [14] 101Computing. "Little Man Computer (LMC) CPU Simulator, Available: <https://www.101computing.net/LMC/>
- [15] J. Gordon. "Binary LMC - Little Man Computer Simulator, Available: <http://www.mathsuniverse.com/lmc>
- [16] B. Maraza-Quispe, O.M. Alejandro-Oviedo, W. Choquehuanca-Quispe, F. Huallpa-Gonzalez, L.M. Quispe-Flores, & S.A. Choquehuayta Palomino, "Hacia el desarrollo del Pensamiento Computacional en los estudiantes de educación superior." *International Journal of Emerging Technologies for E-Learning*, vol. 2, No 1, 2023
- [17] A. Thili, F. Essalmi, & M. Jemni. (2015, July). A mobile educational game for teaching computer architecture. In *2015 IEEE 15th International Conference on Advanced Learning Technologies* (pp. 161163). IEEE.
- [18] A. Thili, F. Essalmi, & M. Jemni. (2015, December). An educational game for teaching computer architecture: Evaluation using learning analytics. In *2015 5th International Conference on Information & Communication Technology and Accessibility (ICTA)* (pp. 1-6). IEEE.
- [19] K.H.A. Tan, Y. Cai, B.T. Lokesh, Y. Zhu, C. Su, & Q. Cao, (2023, September). VR Serious Game for Learning the Computer Organisation and Architecture Course. In *2023 3rd International Conference on Educational Technology (ICET)* (pp. 1-6). IEEE.
- [20] N.C. Cruz, J.L. Redondo, J.D. Álvarez, & P.M. Ortigosa. (2018). "Programación de un juego en ensamblador CHIP-8 como actividad complementaria en la asignatura Arquitectura de Computadores". *XXIX Jornadas de Paralelismo (JP 2018)*.
- [21] J. Kawash, & R. Collier. (2013, October). "Using video game development to engage undergraduate students of assembly language programming". In *Proceedings of the 14th annual ACM SIGITE conference on Information technology education* (pp. 71-76).
- [22] F.J. Gallego-Durán, P. Compañ, R. Satorre Cuerda, & C.J. VillagràArnedo. (2018). "El código máquina mola." *Actas de las JENUI*, Vol.3. Available: https://aenui.org/actas/pdf/JENUI_2018_028.pdf
- [23] R. Bourbia, N. Gouasmi, M. Hadjeris, & H. Seridi, (2014). "Development of serious game to improve computer assembly skills." *Procedia-Social and Behavioral Sciences*, 141, 96-100.
- [24] E. Larraza-Mendiluze, N. Garay-Vitoria, J.I. Martín, J. Muguera, T. Ruiz-Vázquez, I. Soraluze, J.F. Lukas & K. Santiago. (2013). "Gameconsole-based projects for learning the computer input/output subsystem." *IEEE Transactions on Education*, 56(4), 453-458.
- [25] J. W. Creswell, "Revisiting mixed methods and advancing scientific practices," in *The Oxford Handbook of Multimethod and Mixed Methods Research Inquiry*, H.-B. S and J. R. (eds), Eds. Oxford: Oxford University Press, 2015, pp. 61–71.
- [26] S. J. Taylor & R. Bogdan, (1987). "Introducción a los métodos cualitativos de investigación" (Vol. 1, p. 348). Barcelona: Paidós.
- [27] Genially. Available: <https://genial.ly>
- [28] RAE. Available: <https://dle.rae.es/algorithm>
- [29] Code.org – Dice Race. Available : <https://code.org/curriculum/course3/10/Teacher>
- [30] Cse4k12 – Binary Magic Trick. Available: https://www.cse4k12.org/binary/magic_trick.html
- [31] Little Robot Computer: Available: <http://casne.ncl.ac.uk/events/conference/session/71/>
- [32] A. Manches and C. O'malley, "Tangibles for learning: a representational analysis of physical manipulation," *Personal and Ubiquitous Computing*, vol. 16, pp. 405–419, 2012.
- [33] Moon. Available: <https://compus.deusto.es/es/moon-1110011>
- Eduarne Larraza Mendiluze** received the M.S. and Ph.D. degrees in computer science from the University of the Basque Country (UPV/EHU), Donostia - San Sebastián, Spain, in 1999 and 2014, respectively. She is currently an Assistant Professor at UPV/EHU. She is a member of ADIAN research group and her research mainly focused on computer science education at all educational levels. ORCID : 0000-0003-2619-4176.
- Olatz Arbelaitz Gallego** received the M.S. and Ph.D. degrees in computer science from the University of the Basque Country (UPV/EHU), Donostia - San Sebastián, Spain, in 1993 and 2002, respectively. She is currently an Associate Professor at UPV/EHU. She is a member of Aldapa research group (<http://www.aldapa.eus/en/>) and her research interests include data mining, web mining, and unsupervised learning techniques. She is also interested in teaching innovation and implementation of new methodologies in the context of EHEA. ORCID: 0000-0001-7371-2971.
- Olatz Arregi Uriarte** received the M.S. and Ph.D. degrees in computer science from the University of the Basque Country (UPV/EHU), DonostiaSan Sebastian, Spain, in 1986 and 1991, respectively. She is currently an Associate Professor at UPV/EHU. She was director of that Department from 2019 to 2022. She is a member of Ixa research group (<http://www.ixaeus.com>) and HiTZ center (<http://www.hitzeus.com>) and her research mainly focuses on Natural Language Processing. She is also interested in teaching innovation and implementation of new methodologies in the context of EHEA. ORCID: 0000-0003-1187-9332.
- Jose I. Martín Aramburu** received the M.S. and Ph.D. degrees in computer science from the University of the Basque Country (UPV/EHU), DonostiaSan Sebastian, Spain, in 1990 and 1994, respectively. He is an Associate Professor with the Computer Architecture and Technology Department, UPV/EHU. His research interests include data mining and pattern recognition. He is also interested in teaching innovation and implementation of new methodologies in the context of EHEA. ORCID: 0000-0001-6455-131X.
- Jose F. Lukas Mugika** received the Ph.D. degree in pedagogy, a B.A. degree in philosophy and educational sciences (pedagogy section) and a diploma in teaching. He is a Senior Lecturer in the Department of Research and Diagnostic Methods in Education, UPV/EHU. He is an expert in educational measurement and evaluation. He was Deputy Director of the Institute of Educational Sciences of the UPV/EHU (1997-2000) and Director of the Department of Research and Diagnostic Methods in Education of the UPV/EHU (2003-2009). He has been a member of the Scientific Committee of the Basque Institute for Non-University Educational Evaluation and Research (ISEI-IVEI) (2002-2014). ORCID: 0000-0002-9440-035X.