

# Aplicando el modelo de aula semi-invertida para una enseñanza práctica más atractiva en programación de videojuegos

Ivan Rodriguez-Conde<sup>1</sup>, Sean Orme<sup>1</sup>, Celso Campos<sup>2</sup>, Florentino Fdez-Riverola<sup>3,4</sup>

<sup>1</sup>Department of Computer Science, University of Arkansas at Little Rock, Little Rock, Arkansas, US; <sup>2</sup>Departamento de Informática, ESEI - Escuela Superior de Ingeniería Informática, Universidade de Vigo, Ourense, España; <sup>3</sup>CINBIO, Universidade de Vigo, Departamento de Informática, ESEI – Escuela Superior de Ingeniería Informática, Ourense, España; <sup>4</sup>Grupo SING, Instituto de Investigación Sanitaria Galicia Sur (IIS Galicia Sur), SERGAS-UVIGO

{irconde, smorme}@ualr.edu, {ccampos, riverola}@uvigo.gal

## Resumen

Este trabajo presenta y analiza la experiencia docente llevada a cabo para la adopción del modelo de aula semi-invertida como parte del rediseño e implementación del curso *Introduction to Game Programming* ofrecido por la *University of Arkansas at Little Rock*, a fin de paliar la ausencia de sesiones específicas de laboratorio en una asignatura no sólo de carácter eminentemente práctico, sino que, además, constituye dentro del itinerario formativo la primera exposición de los estudiantes al uso de técnicas y tecnologías específicas de desarrollo de videojuegos para el diseño e implementación de soluciones *software* interactivas. En este contexto, el modelo de aula semi-invertida ha resultado adecuado a la hora de mantener la motivación en aquellos alumnos más capaces gracias al protagonismo de las actividades prácticas colaborativas durante la clase, permitiendo igualmente incrementar el interés y la dedicación de los alumnos con mayores dificultades gracias a la breve revisión y al refuerzo de los conceptos clave al comienzo de cada sesión.

## Abstract

This paper reports our experience towards adopting the semi-flipped classroom model for redesigning and implementing the *Introduction to Game Programming* course at the *University of Arkansas at Little Rock*. The model was implemented to overcome the lack of dedicated lab sessions in an eminently practical course, constituting the students' first to video game development techniques and technologies. The semi-flipped classroom model has proven to be effective in maintaining the motivation of the highest-skilled students due to the emphasis on hands-on cooperative learning

activities during class time. Additionally, it has increased the interest and dedication of those with more significant difficulties by briefly reviewing and reinforcing the key concepts at the beginning of each session.

## Palabras clave

Metodologías docentes, aula semi-invertida, curso práctico, programación de videojuegos.

## 1. Introducción

La programación de videojuegos no sólo constituye la base de una industria que rivaliza hoy en día con otras opciones de entretenimiento como el cine, sino que además ha evolucionado hacia una disciplina cada vez más demandada para la creación de experiencias interactivas en industrias tan diversas como la arquitectura, la moda, la automoción o la producción audiovisual. No obstante, a pesar del auge actual del sector del videojuego, su indudable atractivo y el enorme potencial de las técnicas y tecnologías sobre las que se sustenta como puerta de entrada a sectores afines, la oferta formativa específica dista de ser reflejo de esta realidad, y su presencia en el catálogo de las instituciones de educación superior estadounidenses es todavía reducida.

Conocedora de estas necesidades y la oportunidad de mercado que esto supone en el contexto de competitividad extrema que vive el espacio de formación universitario en Norteamérica, *University of Arkansas at Little Rock (UALR)* ofrece como parte de su catálogo formativo la titulación de *Bachelor of Science in Computer Science - GAME (Graphics, Animation, Multimedia, and Entertainment) option*, una titulación de

grado de 120 horas, que junto al *Bachelor of Fine Arts in Game and Interactive Design* de la *Arkansas Tech University*, constituye una de las dos únicas alternativas en formación universitaria específica de programación de videojuegos existente en el estado de Arkansas. Desafortunadamente, este escenario aparentemente favorable no sólo no se ha traducido en unos mejores números de matrícula, sino que, contrariamente a lo que cabría esperar, ha visto en los últimos años cómo el interés por la titulación y el nivel de satisfacción de los estudiantes caía hasta alcanzar mínimos históricos.

A falta de un estudio que determine de manera empírica este fenómeno y sus posibles causas, es posible, sin embargo, identificar de manera intuitiva ciertos elementos que, sin duda, han podido jugar un papel importante en este decrecimiento como (i) el bajo nivel económico y cultural general de la población del estado de Arkansas, (ii) la complejidad inherente del espacio temático abordado y las materias impartidas bajo su paraguas y (iii) el método de enseñanza empleado para ello.

i. En Arkansas un considerable número de estudiantes presentan graves carencias de aprendizaje, siendo esta brecha especialmente profunda entre estudiantes afroamericanos y estudiantes con ingresos bajos [4]. Esta desigualdad social y los retos educativos que ello conlleva se ha visto agudizado por las condiciones económicas del estado después de la pandemia. El 16.3% de los habitantes se encuentran en situación de pobreza, y más de una de cada tres familias con niños vive en hogares por encima del nivel federal de pobreza, pero con ingresos insuficientes para cubrir las necesidades presupuestarias básicas.

ii. La programación de videojuegos, por su parte, constituye un campo multidisciplinar de elevada complejidad técnica que requiere de un amplio abanico de habilidades transversales [1]. Requiere una elevada capacidad de abstracción y demanda, además, una profunda comprensión y la consiguiente puesta en práctica de conceptos o fundamentos de ámbitos de conocimiento fronterizos o parcialmente solapados con la informática como las matemáticas o la física que, por un lado, en la mayoría de los casos, o bien no han sido abordados previamente en la vida académica del estudiante o bien, si lo han sido, no con el nivel de detalle requerido.

iii. Por último, existe escaso interés entre el profesorado universitario por explorar metodologías innovadoras de enseñanza [6]. En el caso concreto del *Department of Computer Science* (en adelante, departamento de informática) de la UALR, al que se encuentra adscrita la titulación de programación de videojuegos, las asignaturas, a pesar de su carácter eminentemente práctico, son ofertadas como cursos de tan solo 3 *credit hours* (2 horas y media de clases semanales) y principalmente en el formato tradicional de sesión magistral.

Así, salvo alguna excepción puntual como la materia *CPSC 1375 Programming I*, que contempla dos horas semanales de laboratorio, las clases se articulan en esencia en torno a aspectos eminentemente teóricos y delegan la puesta en práctica de estos en las asignaciones planteadas como trabajo personal del alumno.

Todas estas complejidades, redundan, en definitiva, en un escenario hostil para el estudiante (que se ve envuelto en un proceso de aprendizaje frustrante) e igualmente complicado para el docente (que debe lidiar con el corsé administrativo impuesto y dar respuesta al mismo tiempo a la brecha educativa anteriormente apuntada), derivando en ambos casos en una disminución de interés y motivación.

Con el objetivo de hacer frente a esta complicada coyuntura y revertir una situación que bien pudiera responder a la realidad de infinidad de titulaciones e instituciones académicas de todo el mundo en el actual contexto postpandemia, el departamento de informática de la UALR puso en marcha en otoño del 2021 un proyecto piloto de innovación docente, explorando la aplicación en la materia *CPSC 2377 Introduction to Game Programming* de un modelo de aula semi-invertida. Este modelo, como veremos en la siguiente sección, se implementa como una ligera variación del modelo de aula invertida, un modelo basado en sesiones más centradas en el debate y la resolución de problemas que no sólo se ha erigido como un método efectivo para salvar la brecha digital entre estudiantes desfavorecidos [2], sino que ha demostrado ser ventajoso y adecuado para impartir cursos de programación, maximizando el soporte al aprendizaje de los estudiantes [7].

En este trabajo se detalla la nueva propuesta docente elaborada para la materia *CPSC 2377*, resultado de la adaptación del curso al modelo de aula semi-invertida, especificando en la segunda sección los aspectos más relevantes de la organización de la asignatura como el método de enseñanza empleado, los recursos elaborados, las herramientas utilizadas o la evaluación. En la sección 3, en un análisis preliminar de la experiencia, se presentan y discuten los resultados académicos de los estudiantes y los datos obtenidos a través de las encuestas de evaluación docente. Finalmente, la sección 4 concluye el trabajo incluyendo una serie de conclusiones generales y acciones a llevar a cabo en trabajos futuros.

## 2. Propuesta docente

### 2.1. Visión general del curso

La asignatura *CPSC 2377 Introduction to Game Programming* es un curso de 3 *credit hours*, impartida en el semestre de otoño de cada año académico y que comprende un periodo lectivo de 15 semanas. Concretamente se ubica en el tercer semestre de la especiali-

dad en videojuegos del *Bachelor's Degree in Computer Science* ofertado por el departamento de informática de UALR, como una ampliación natural de las asignaturas *CPSC 1375 Programming I* y *CPSC 2376 Programming II* en las que se introduce a los estudiantes del grado a la resolución de problemas mediante (i) la codificación de algoritmos bajo el paradigma de programación orientada a objetos, y (ii) la explotación de estructuras de datos adecuadas para la creación del modelo de datos pertinente en cada caso, utilizando para ello el lenguaje de programación C++.

Apoyado en estos fundamentos, el curso introductorio a la programación de videojuegos utiliza de nuevo el lenguaje C++ como elemento vehicular para abordar, en su caso, la aplicación de múltiples técnicas avanzadas de programación de particular relevancia a la hora de implementar no solamente videojuegos, sino, en general, soluciones *software* multimedia e interactivas: (i) la gestión y uso adecuado de recursos (i.e., contenido incorporado por lo común a través de ficheros externos como imágenes, audio o animaciones), (ii) la representación o dibujado en pantalla de gráficos, (iii) la creación de una arquitectura fácilmente escalable y mantenible, (iv) la implementación de físicas realistas (entre ellas, la detección y resolución de colisiones), (v) la puesta en práctica de algoritmos básicos para la simulación de agentes autónomos a fin de crear los denominados personajes no jugables (i.e., personajes no controlados directamente por el jugador, conocidos como NPC), y (vi) la comunicación en red de diferentes instancias del juego.

Más aún, y al margen de las técnicas listadas, el curso da a conocer y forma al alumno en el manejo de herramientas y en el uso de metodologías de desarrollo de *software* estándar en la industria, más allá de la mera creación de micro algoritmos frente a problemas “de juguete”, habitual en cursos introductorios. Así, la asignatura supone para gran parte del alumnado de la titulación la primera exposición al uso de librerías externas en un proyecto *software* (en particular, *Boost*, *SDL* y *Box2D*), a la correcta configuración y enlazado de éstas en un entorno de desarrollo integrado profesional (concretamente, Visual Studio), así como al aprovechamiento de los sistemas de control de versiones como método de copia de seguridad e instrumento de soporte al trabajo colaborativo.

Con esta finalidad, los objetivos de aprendizaje listados en la guía docente de la asignatura establecen que, una vez completado con éxito el curso, los estudiantes serán capaces de: (i) formular soluciones lógicas a diferentes problemas, (ii) utilizar de manera conjunta las estructuras lógicas y los procedimientos algorítmicos apropiados para implementar un motor de videojuegos, (iii) aplicar conceptos de programación orientada a objetos para la concepción de soluciones en C++, (iv) aplicar conceptos relativos a la corrección de los programas *software*, y crear estos últimos de

manera inteligente para dar respuesta a problemas en diversas áreas temáticas, (v) escribir código bien estructurado, (vi) escribir código limpio y documentado adecuadamente, y (vii) aplicar conceptos matemáticos relativos a las leyes físicas en entornos de juego digital.

## 2.2. Método de enseñanza

Tal y como se indica en la Introducción, en el año 2021 la asignatura pasa a ser impartida bajo la modalidad de aula semi-invertida, en dos sesiones semanales de 1 hora y 15 minutos de duración. Una gran parte del proceso de aprendizaje es delegado en los estudiantes, a los que se indica que revisen con antelación los contenidos relativos a la temática a tratar cada semana, permitiendo así destinar el grueso del tiempo de clase al trabajo en actividades cooperativas dirigidas a la aplicación práctica de dichos contenidos. No obstante, a diferencia del modelo de aula invertida, el docente emplea la primera clase de la semana para repasar las cuestiones más relevantes o aquellos aspectos de la materia abordada que tradicionalmente han supuesto un mayor reto a los estudiantes en ediciones pasadas del curso, planteando la sesión como un seminario o foro de discusión en el que los estudiantes pueden compartir sus dudas e incluso contribuir al aprendizaje común con su propia visión y comprensión de los temas.

Por su parte, la segunda clase de la semana se articula como una sesión complementaria de laboratorio en la que cada estudiante trabaja en la resolución de un caso práctico relacionado con las técnicas concretas examinadas dicha semana. Siguiendo la filosofía de fomentar la generación de conocimiento de manera colectiva y ofrecer una experiencia de aprendizaje integral, estas tareas son planteadas como actividades a llevar a cabo en equipo. El docente, acompañado por un asistente, se limita a desempeñar labores de asistencia y consulta, ayudando en la organización o gestión interna de los equipos, y respondiendo preguntas tanto de índole técnica como referidas a la propia organización del curso.

A mayores de lo indicado, también desde 2021 la asignatura se oferta como un curso híbrido (abierto a estudiantes presenciales y *online*), con el fin de adaptarse al nuevo escenario en educación superior motivado por la irrupción del COVID-19 a comienzos del año 2020. A este respecto, se incorporó al curso una sección *online* síncrona, surgiendo la necesidad de acomodar la enseñanza para dar soporte efectivo a los estudiantes matriculados en esta modalidad. Con tal objetivo, desde entonces, cada clase es retransmitida en vivo para todos aquellos alumnos que asisten en línea a la sesión, permitiendo a los participantes utilizar de manera ordenada su micrófono e incluso compartir su pantalla para intervenir en el transcurso de la clase, a fin de fomentar un debate lo más fluido posible. De igual modo, se pone a disposición del grupo una herra-

mienta de chat como alternativa de comunicación por escrito, siendo ésta monitorizada en todo momento por el asistente para moderar las cuestiones y mensajes posteados, proporcionando respuesta inmediata en la medida de lo posible y minimizando así las interrupciones ocasionadas por tal motivo.

### 2.3. Materiales

Para apoyar el proceso de aprendizaje de los estudiantes bajo el diseño de aula semi-invertida descrito, se pone a su disposición una importante colección de contenido multimedia de diversa naturaleza con el fin de facilitar la toma de contacto y asimilación de los conceptos teóricos en las tareas de revisión previas a la primera clase de cada semana. En esencia, se combina el uso de contenido textual (en la línea de los apuntes en papel tradicionales) con contenido videográfico y ejemplos de código. Más concretamente, al final de cada semana se publican los siguientes materiales de estudio y consulta.

*i.* Visión general en texto. En líneas generales, el uso de contenido textual como material educativo en el curso ha sido minimizado. En esencia, se limita a una mera descripción general de los objetivos de aprendizaje y la temática abordada en cada módulo formativo, sirviendo así de guía de aprendizaje o hilo conductor de los restantes contenidos ofrecidos. Dicho esto, dependiendo del nivel de complejidad y alcance de la temática abordada, se utiliza adicionalmente este formato para la entrega de resúmenes, presentando conceptos clave o técnicas de interés de manera esquemática, a modo de referencia para una consulta rápida.

*ii.* Micro-lecciones en video. Para una presentación más atractiva de los contenidos teóricos se utiliza como principal material de aprendizaje píldoras en vídeo de una duración promedio de cinco minutos para flexibilizar su consumo y evitar posible fatiga. Cada módulo comprende una media de 10 vídeos en los que el docente, típicamente a través de la codificación de un ejemplo y explicando a viva voz los pasos realizados, introduce los conceptos y prácticas *core* de la asignatura. A pesar de ser un proceso tedioso, la producción de videos ha sido una tarea de preparación ineludible, ante la ausencia en plataformas de uso masivo como *Youtube* de videos con duración contenida y temática acorde a los objetivos perseguidos.

*iii.* Presentaciones en formato PDF. Las diapositivas digitales utilizadas por el docente en la primera sesión de cada semana son compartidas igualmente con los estudiantes, no sólo con el objetivo de que conozcan de antemano los puntos concretos a tratar en la clase de revisión y puedan con ello preparar con mayor conocimiento sus comentarios y posibles preguntas, sino también con la idea de proporcionarles un medio sobre el que tomar sus notas.

*iv.* Ejemplos de código. Acompañando a las micro-lecciones en video, se ponen a disposición del alumna-

do proyectos de ejemplo. Estos proyectos juegan un doble papel en el proceso de aprendizaje. En primer lugar, pueden ser utilizados como punto de partida o solución parcial a los ejercicios prácticos planteados. En segundo lugar, cada proyecto es concebido como un repositorio de código con múltiples ramas, convirtiéndose en una herramienta de aprendizaje en sí misma. La creación de múltiples ramas permite por un lado aislar funcionalidades concretas y especificar así el código utilizado en cada caso y, por otro, definir ramas paralelas para cada funcionalidad, aportando en una la solución esperada e incluyendo en la segunda, en lugar del código, comentarios TODO (comentarios de línea que marcan tareas a realizar) a modo de guía con el fin de que los estudiantes intenten de manera voluntaria codificar la misma solución como ejercicio.

De manera complementaria a los materiales proporcionados como instrumento de estudio anterior a la clase, la primera sesión de cada semana es grabada en video y publicada al final de esa semana. En total, se facilitan a los estudiantes 12 grabaciones de aproximadamente 50 minutos, que recogen únicamente la explicación teórica llevada a cabo por el docente y omiten la parte de preguntas y comentarios relativos al trabajo práctico (que se comentará en el apartado 2.5) a fin de incentivar con ello la asistencia y la participación de los estudiantes en las sesiones de repaso.

### 2.4. Infraestructura

Reducir la parte instructiva del tiempo de clase proporciona a los estudiantes mejores oportunidades para interactuar más profundamente con el contenido y con sus iguales [3], otorgando al curso una flexibilidad fundamental para ser llevado a buen puerto en su modalidad presencial, pero sobre todo en su modalidad *online*. La transición a este modelo híbrido ha sido posible gracias a la disponibilidad en el escenario tecnológico actual de herramientas *software*, tanto educativas como de propósito específico, para dar respuesta a las nuevas necesidades de distribución de contenido, gestión de la comunicación y trabajo cooperativo.

En particular, el uso de *Blackboard* (el sistema de gestión de aprendizaje implantado en la UALR) ha permitido forjar un marco didáctico sólido para la materia *CPSC 2377*, maximizando la disponibilidad de los contenidos y haciendo más sencilla la organización y distribución controlada de los mismos. En relación precisamente a su distribución, gracias a las herramientas de edición integradas en la plataforma ha sido posible no sólo crear de manera racional los diferentes módulos de aprendizaje, sino también y más importante, programar y establecer de antemano la fecha de publicación de materiales didácticos y asignaciones, así como la fecha de vencimiento de estas últimas.

También el propio *Blackboard*, concretamente *Blackboard Collaborate Ultra*, ha sido utilizado como herramienta de videoconferencia para ofrecer a los es-

tudiantes en línea una alternativa virtual a las clases y tutorías en el campus. Su versatilidad ha permitido unir en cada sesión a estudiantes de las modalidades *online* y presencial en una experiencia de aprendizaje conjunta, poniendo a su disposición funcionalidades como la mano alzada virtual y las encuestas, aprovechadas por el docente, especialmente en las sesiones de revisión, para hacer preguntas de control, recoger el *feedback* de los estudiantes e incrementar así su nivel de participación. Se suman a estas herramientas, además, la funcionalidad “grupos de trabajo” que, como su propio nombre indica, ha sido utilizada para dar soporte al trabajo en grupo de los miembros del curso, permitiendo al docente distribuir a los estudiantes en salas independientes y moderar su actividad en ellas.

Si bien *Blackboard Collaborate Ultra*, tal y como se acaba de comentar, goza de la polivalencia necesaria para cubrir las necesidades de las sesiones síncronas del curso en lo que a organización y retransmisión se refiere, se trata de una herramienta fundamentalmente orientada al docente, con una interfaz de usuario concebida para un uso esencialmente en sistemas de escritorio y, además, limitada en sus capacidades de mensajería, no permitiendo, por ejemplo, el almacenamiento, y búsqueda de información. Constituye así, en consecuencia, una opción no demasiado conveniente para fomentar la interacción fuera del tiempo de clase. Por este motivo, finalmente ha sido *Slack*, la herramienta de mensajería profesional, la alternativa adoptada para facilitar dichas interacciones. Concretamente, a través de *Slack* se incorpora a la gestión de la comunicación en el curso dos elementos fundamentales: (i) un espacio de debate abierto, que permite al docente proporcionar *feedback* inmediato a los estudiantes, sino además fomentar la solidaridad entre ellos y la construcción progresiva de una base de conocimiento conjunta, y (ii) un espacio de trabajo privado para cada equipo en el que centralizar las cuestiones específicas de grupo e intercambiar materiales.

Por último, como soporte a las tareas de programación en grupo, se requiere a los estudiantes de cada equipo la creación de un repositorio de código compartido, siendo *GitHub* el servicio de alojamiento en remoto seleccionado para permitir la colaboración eficaz entre los miembros del grupo y el control de cambios en su código. Su utilidad no se circunscribe, no obstante, al ámbito de actividad de los estudiantes. También el docente ha aprovechado sus capacidades para agilizar la creación y asignación de tareas a los equipos, empleando para ello *GitHub Classroom*, un servicio *online* orientado a la educación basado en la funcionalidad de *GitHub*. Dicho servicio ha permitido automatizar gran parte del flujo de trabajo asociado a la creación y gestión de los repositorios grupales, permitiendo incluso al docente proporcionar *feedback* a sus estudiantes través de su sistema de *pull requests* mediante correcciones directas sobre el código creado.

## 2.5. Aprendizaje basado en proyectos

Tanto la planificación como el desarrollo de las clases o sesiones síncronas tiene por objetivo último facilitar la construcción progresiva de un videojuego o experiencia interactiva que los estudiantes deben presentar al final del semestre como resultado del trabajo realizado. De temática abierta, se considera como un proyecto válido toda aquella aplicación gráfica que se base en una secuencia estructurada de interacciones dirigidas a un objetivo.

Más concretamente, se requiere que el alumno diseñe e implemente un programa que incluya los elementos o aspectos listados a continuación: (i) una temática que sirva de hilo conductor de los diferentes niveles, (ii) al menos dos niveles o subobjetivos únicos pero relacionados entre sí, de modo que constituyan una evolución hacia el objetivo general del juego, (iii) un conjunto de mecánicas que definan la jugabilidad necesaria para la consecución de los subobjetivos planteados, (iv) contenido multimedia (p.ej., imágenes, *sprites*, efectos de sonido, etc.) al servicio de la jugabilidad y gobernado por las mecánicas implementadas, y (v) cierto componente de inteligencia en el juego, de modo que sea capaz de tomar algún tipo de decisión en función de las entradas del usuario.

Con el objetivo de fomentar un aprendizaje entre iguales, el proceso de desarrollo del proyecto se modela en la asignatura como un trabajo colaborativo, abordado en equipos de hasta tres personas que son creados por los propios estudiantes durante la primera semana del semestre. Asimismo, al igual que sucede en el ámbito profesional, dicho proceso es llevado a cabo de manera progresiva y se estructura en diferentes fases. Concretamente, el proyecto contempla una fase inicial de diseño y seis fases de desarrollo diferenciadas, que se extienden desde la segunda semana del curso hasta su finalización, existiendo una correspondencia uno a uno entre las fases de desarrollo y los bloques temáticos apuntados en el segundo párrafo del apartado 2.1 como eje vertebrador de la asignatura.

Es preciso mencionar que, si bien cada equipo goza de total libertad a la hora de distribuir la carga de trabajo, el docente establece en la descripción de cada asignación a abordar un reparto de tareas recomendado como guía a este respecto. Se pretende con ello evitar posibles conflictos internos en el equipo, así como potenciales desajustes en lo que a expectativas se refiere en tal sentido entre el propio docente y el alumnado.

## 2.6. Evaluación

Siguiendo un modelo de evaluación continua, con el objetivo de realizar un seguimiento adecuado y medir el progreso de los estudiantes en el transcurso del proyecto, se concibe el final de cada fase como un hito que requiere la entrega de un producto parcial como evidencia de finalización con éxito. Guiado por los requi-

sitos y los objetivos de aprendizaje proporcionados de antemano por el docente a través de un documento detallado de descripción de la asignación, la expectativa es que cada miembro de un equipo no sólo contribuya en la misma medida a la codificación de la solución perseguida, sino que, además, demuestre una comprensión completa de ésta, más allá del código aportado de forma individual.

Para ello, cada estudiante debe incorporar todo código escrito al repositorio grupal indicando los cambios realizados a través del uso adecuado de los mensajes de *commit* y, adicionalmente, debe grabar en vídeo un recorrido de 5 minutos máximo por el código producido, explicando las nuevas funcionalidades implementadas. Igualmente, la presentación de resultados debe ser realizada de manera individual, siendo necesaria la subida a *Blackboard* tanto del *screencast* como del enlace al repositorio.

En lo que respecta a la valoración del trabajo realizado, el código proporcionado se presume (como mínimo) compilable y ejecutable, e idealmente debe presentar las características señaladas en la especificación suministrada, así como reflejar buenas prácticas de programación (p.ej., código limpio y comentarios informativos). Estos aspectos conforman el grueso de los criterios de evaluación aplicados y gobiernan la política de calificaciones adoptada, basada en el sistema de penalizaciones que se detalla a continuación: 2.5% por faltas menores como el uso inconsistente de llaves y espacios en el código; 5% por malas prácticas, esto es, la utilización de métodos inadecuados para el desarrollo; 10% por cada tarea principal no completada; 20% por cada tarea principal no comenzada; 75% por no distribuir la carga de trabajo de manera equitativa; y 20% por no presentar el video explicativo.

Por último, la calificación final del alumno en el curso pretende ser reflejo de su rendimiento global a lo largo de todo el semestre y, como tal, no solo no se deriva en exclusiva de la nota obtenida en el trabajo final, sino que depende en gran medida de la puntuación lograda en las diferentes asignaciones (ocho en total). Así, las asignaciones, en particular, representan el 70% de la nota final del curso, mientras que el proyecto final supone el 30% restante.

### 3. Resultados

Como primer paso hacia el estudio y evaluación de los efectos logrados con la puesta en marcha de la presente experiencia, se han recogido y analizado los resultados académicos y los resultados de las encuestas de evaluación docente correspondientes a los años 2020 y 2021. A pesar de que la muestra se circunscribe únicamente a estas dos ediciones del curso *CPSC 2377*, debido a que la enseñanza en años previos fue realizada por otro docente, recoge al menos los datos referidos al estudio piloto llevado a cabo en el curso

académico 2021-2022, permitiendo su comparación con los recogidos durante el curso 2020-2021, la última edición impartida bajo el modelo más tradicional de sesión magistral.

#### 3.1. Resultados académicos

A nivel académico, al margen de la significativa reducción de matrícula que ya se apuntaba en la primera sección, los datos recogidos en el Cuadro 1 refrendan en cierto modo las expectativas de partida acerca del modelo de aula semi-invertida, y muestran una mejora importante en todos los parámetros considerados para determinar y calibrar la consecución de los objetivos de aprendizaje por parte del alumnado.

Más allá de la subida en algo más de dos puntos de la nota media en el curso, destaca la total ausencia de suspensos y, muy especialmente, de estudiantes que han causado baja. En este sentido, aunque es prematuro aventurarse a realizar afirmaciones demasiado categóricas, las cifras transmiten que el paradigma adoptado, gracias a una interacción más fluida y a un enfoque más práctico, no sólo ha permitido mantener el rendimiento y, por lo tanto, la implicación de los estudiantes de mejor nivel, sino que ha logrado mantener la motivación (o al menos, captar su atención hasta el final) de aquellos que en otras ocasiones acababan abandonando la asignatura.

Semestre	# matrículas	# bajas	# suspensos	Nota media
Fall2020	18	5	4	6.7
Fall2021	8	0	0	8.8

Cuadro 1: Resumen comparativo de los resultados académicos de los estudiantes en 2020 y 2021.

Los datos presentados en el Cuadro 2 refuerzan, además, en cierto modo, las observaciones realizadas, descartando un posible efecto del nivel de formación específica en programación previo del alumnado (representado por la nota media obtenida en los cursos *CPSC 1275* y *CPSC 2376*, prerequisite de *CPSC 2377*), al mostrar los datos recogidos para 2020 y 2021 una distribución no demasiado diferente.

#### 3.2. Resultados de las encuestas de evaluación docente

Las respuestas proporcionadas por los estudiantes a las encuestas de evaluación docente, aunque en gran medida sesgadas por su progreso en la asignatura entre otros factores [5], permiten complementar este ejercicio de análisis preliminar incorporando nuevos datos representativos de la percepción del alumnado sobre su propio aprendizaje y la calidad de la docencia recibida. Como referencia, se proporcionan a continuación las

Semestre	Estudiante	CPSC 2377	CPSC 1375	CPSC 2376
Fall 2020	X01	A	A	A
	X02	A	A	A
	X03	A	A	B
	X04	A	A	A
	X05	W	C*	C*
	X06	A	A	A
	X07	B	A	A
	X08	D	A	A**
	X09	W	A	B
	X10	F	B	C**
	X11	W	B	B
	X12	F	A	B
	X13	W	C	W
	X14	D	B	A**
	X15	C	A	C
X16	W	A	C	
X17	F	A	A	
X18	F	A	C	
Fall 2021	Y01	A	A	A
	Y02	C	C	C**
	Y03	A	A	A
	Y04	A	A*	A*
	Y05	A	A	A
	Y06	C	A**	C
	Y07	C	B	B
	Y08	A	A	A
Equivalencia de letras con escala de calificación española. A: sobresaliente [9,10]; B: notable [8, 9]; C: aprobado [7, 8]; D: aprobado [6, 7]; F: suspenso [0, 6]. Variaciones. *: traspaso de expediente; **: segunda convocatoria.				

Cuadro 2: Rendimiento académico de los estudiantes en los cursos 2020 y 2021, junto a su rendimiento previo en cursos de programación.

preguntas concretas formuladas.

- P01. ¿Qué nivel de conocimientos demostró tener el profesor?
- P02. ¿Ha explicado el material del curso con claridad?
- P03. ¿En qué grado se ha preocupado de que los alumnos aprendieran el material?
- P04. ¿Ha preparado las clases de manera adecuada?
- P05. ¿Ha respondido correctamente las preguntas de los estudiantes?
- P06. ¿La velocidad con la que presentó el material del curso fue demasiado rápida, demasiado lenta o más o menos adecuada?
- P07. ¿En qué medida te han ayudado las asignaciones a comprender la materia?
- P08. ¿Te resultó fácil o difícil ponerte en contacto con el profesor fuera del horario de clase?
- P09. Los contenidos son de calidad y no requieren demasiadas explicaciones.

P10. La evaluación se ajusta a los materiales tratados previamente de manera adecuada.

P11. El método de calificación me parece justo.

P12. He aprendido mucho sobre la materia del curso.

P13. En una escala del 1 al 5, valora la actuación general del docente.

Los resultados obtenidos en este caso no vienen sino a apoyar los buenos indicios apuntados por los resultados académicos. De acuerdo con las cifras que muestra el Cuadro 3, mejora el nivel de satisfacción de los estudiantes con los contenidos y asignaciones (P07, P09) y confirman su conformidad con el método de evaluación empleado (P10, P11), lo que ha revertido a su propio juicio en una mejora del aprendizaje (P12), revalorizando incluso la imagen del docente en las diferentes facetas de la enseñanza (P02-P05, P13).

Preguntas	Fall2020	Fall2021	Óptimo
P01	1.56	1.2	1
P02	2.56	1.6	1
P03	2.33	1.2	1
P04	2.45	1.4	1
P05	2.33	1.2	1
P06	2.33	2.6	3
P07	2.56	1.8	1
P08	1.89	1.2	1
P09	2.78	3.8	5
P10	2.89	4	5
P11	3.56	4.8	5
P12	3.89	4.6	5
P13	3.34	4.8	5

Cuadro 3: Puntuación media obtenida para cada pregunta de las encuestas de satisfacción docente consideradas, junto al valor considerado óptimo para cada una.

## 4. Conclusiones y trabajo futuro

En este trabajo se ha presentado la experiencia docente puesta en práctica durante el curso académico 2021-2022 en una asignatura de introducción a la programación de videojuegos para su adaptación a la modalidad de aula semi-invertida. Para esto se han propuesto una serie de prácticas enfocadas a reforzar el aspecto práctico de la asignatura, mejorar la interacción de los estudiantes, y mejorar su nivel de implicación. El proceso de adaptación no ha sido una tarea directa, sino que ha implicado una profunda reestructuración de la asignatura, empezando por la organización de las clases, la modificación de los materiales, la creación de nuevos recursos e incluso un cambio en las herramientas de soporte utilizadas.

Si bien los resultados obtenidos son prometedores e indican que la metodología empleada es el camino a

seguir, resulta necesario, entre otras cosas, incrementar la participación de los estudiantes en las sesiones teóricas y ahondar en la automatización de tareas del flujo de trabajo descrito para aliviar la carga de trabajo adicional que esta metodología supone para el docente. Asimismo, de cara a trabajos futuros y, especialmente, en lo que a la evaluación de la experiencia se refiere, los resultados obtenidos no sólo deberán ser refrendados por un contraste estadístico, sino que, además, será necesario extender el estudio tanto en el tiempo como en lo que al número de estudiantes involucrados respecta, a fin de (i) confirmar la tendencia, (ii) profundizar en el análisis para dirimir qué elementos y en qué grado han sido artífices del potencial progreso, y, por último, (iii) obtener un balance global de la experiencia, evaluando la dedicación total que ha supuesto su diseño y puesta en práctica frente a los beneficios finalmente obtenidos.

## Referencias

- [1] Björn Berg Marklund, Henrik Engström, Marcus Hellkvist y Per Backlund. What Empirically Based Research Tells Us About Game Development. *The Computer Games Journal*, 8(3):179-198, 2019. doi: 10.1007/s40869-019-00085-1.
- [2] Blaženka Divjak, Bart Rienties, Francisco Iniesto, Petra Vondra y Mirza Žižak. Flipped classrooms in higher education during the COVID-19 pandemic: findings and future research recommendations. *International Journal of Educational Technology in Higher Education*, 19(1):9, 2022. doi: 10.1186/s41239-021-00316-4.
- [3] Hsiu-Ting Hung. Flipping the classroom for English language learners to foster active learning. *Computer Assisted Language Learning*, 28(1):81-96, 2015. doi: 10.1080/09588221.2014.967701.
- [4] Jerry D Johnson, Craig B Howley y Aimee A Howley. Size, Excellence, and Equity: A Report on Arkansas Schools and Districts. 2002. p. 49
- [5] Rebecca J. Kreitzer y Jennie Sweet-Cushman. Evaluating Student Evaluations of Teaching: a Review of Measurement and Equity Bias in SETs and Recommendations for Ethical Reform. *Journal of Academic Ethics*, 20(1):73-84, 2022. doi: 10.1007/s10805-021-09400-w0.
- [6] David Pundak y Shmaryahu Rozner. Empowering Engineering College Staff to Adopt Active Learning Methods. *Journal of Science Education and Technology*, 17(2):152-163, 2008. doi: 10.1007/s10956-007-9057-3.
- [7] Zeynep Turan y Yuksel Goktas. A new approach in higher education: the students' views on flipped classroom method. *Journal of Higher Education and Science*, 5:156, 2015.