

Aula invertida en cursos consecutivos de Programación (Programación I y II)

Inés Friss de Kereki, Dr. Ing.¹, Alejandro Adorjan, M. Ing.¹

¹Universidad ORT Uruguay, Uruguay, kereki_i@ort.edu.uy, adorjan@ort.edu.uy

Abstract– En 2019 se implementó un nuevo plan de estudios en la Facultad de Ingeniería de la Universidad ORT Uruguay. Las materias Programación I (de 1er semestre) y Programación II (de 2do semestre) se modificaron en forma de dictado y contenido. Antes, se dictaban en formato de clases magistrales y las dos utilizaban Programación Orientada a Objetos y el lenguaje Java. En el nuevo plan, se pasaron completamente a aula invertida: el estudiante previamente a cada clase revisa ciertos materiales y en la clase se realizan actividades de aplicación y discusión. El nuevo contenido de Programación I está orientado a desarrollar el pensamiento computacional y las habilidades básicas de programación. El lenguaje es JavaScript. Programación II pasó a centrarse en Programación Orientada a Objetos y algoritmia más avanzada. El lenguaje es Java. El resultado del primer dictado del nuevo curso de Programación I finalizado en julio de 2019 mostró valores de aprobación similares a los de cursos previos y se detectó alta motivación e involucramiento tanto de los estudiantes como de los docentes con la propuesta. En este trabajo, se presenta la implementación del segundo semestre de 2019 de la misma cohorte, que corresponde al curso de Programación II. Se obtuvieron valores de aprobación similares a los de cursos previos de Programación II y también alta motivación y compromiso. Se mantuvo la conformidad con el formato invertido que fue utilizado en todo el año. Considerando que las evaluaciones son de nivel equivalente al del curso anterior de Programación II, es de destacar que con el nuevo plan y forma de dictado, al terminar el 2do semestre, los estudiantes manejan dos lenguajes (en vez de solamente uno) y más paradigmas de programación.

Keywords- Programación, Aula Invertida, Java.

I. INTRODUCCIÓN

La enseñanza inicial de la programación es un proceso complejo y presenta desafíos en relación a la actualización de contenidos y prácticas desde la perspectiva pedagógica [1,2].

A partir de 2019 y con la meta de actualizar los planes de estudios y cambiar el enfoque de enseñanza hacia el aprendizaje activo, fomentar el autoaprendizaje y aumentar la motivación, se implementó un nuevo plan en las carreras de Ingeniería en Sistemas, Eléctrica, Electrónica, Telecomunicaciones y Licenciatura en Sistemas en la Facultad de Ingeniería de la Universidad ORT Uruguay. Los cursos de Programación I (P1) y Programación II (P2), que corresponden tradicionalmente a los dos primeros cursos consecutivos en la secuencia del área Programación de

carreras de Computación y similares, fueron modificados. “Cursos como P1 y P2 pueden presentar un desafío pedagógico interesante cuando se trata de la relación teoría-práctica, junto con aspectos que involucran la implementación del curso, el lenguaje de programación utilizado y las características de los estudiantes involucrados en el proceso” [2].

En el plan anterior, estos cursos estaban focalizados en la Programación Orientada a Objetos (POO) utilizando Java como lenguaje de programación. El método de enseñanza era de clases magistrales, donde el rol principal del docente era la exposición y presentación de los contenidos del curso. El nuevo curso de P1 tiene por objetivos desarrollar el pensamiento computacional y las habilidades básicas para resolver problemas sencillos de programación en un lenguaje como JavaScript. El nuevo curso de P2 está orientado a presentar los conceptos de programación orientada a objetos (POO) y profundizar en diseño de algoritmos, implementándose en Java.

En el nuevo plan se decidió utilizar “aula invertida” en la totalidad de ambos cursos. El aula invertida es un enfoque híbrido que combina clases presenciales con materiales y actividades en línea. Como señalan Bergmann y Sams [3], la idea central sobre el aula invertida es que “lo que tradicionalmente se hace en clase se hace ahora en casa, y lo que tradicionalmente se hace como deberes se hace en clase”. El aula invertida mejora el interés de los estudiantes y las habilidades de aprendizaje autónomo [4].

En el primer semestre de 2019, en P1 ya se aplicó el formato de aula invertida y se reportó en [5] por parte de los autores la evidencia de alto compromiso e involucramiento de parte de los estudiantes. Wei et al sugieren explorar los efectos del aula invertida en varias materias y en un período mayor de tiempo [6]. El presente trabajo tiene como objetivo presentar la aplicación de las modificaciones del curso de P2 y sus resultados. Está estructurado de la siguiente forma: en la siguiente sección se describen antecedentes relativos a los cursos de P1 y P2 así como al aula invertida, luego se detalla el nuevo curso de P2, haciéndose énfasis en los contenidos, preparación, dictado en sí y evaluación; finalmente se presentan los resultados del primer dictado y se ofrecen conclusiones y líneas de trabajo futuro.

Digital Object Identifier: (only for full papers, inserted by LACCEI).
ISSN, ISBN: (to be inserted by LACCEI).

II. ANTECEDENTES

En esta sección se describen detalladamente los cursos previos de P1 y P2 de Universidad ORT Uruguay, así como el nuevo plan de P1, para contextualizar el nuevo curso de P2. También se presentan aspectos sobre aula invertida.

A. Cursos de Programación I y Programación II en Universidad ORT Uruguay

En el plan anterior de las carreras citadas, P1 y P2 estaban orientadas a la POO. El lenguaje era Java en ambos cursos y la forma de dictado era la tradicional: clases magistrales donde se presentaban los temas y para actividad posterior fuera de clase se planteaban ejercicios prácticos de aplicación.

El curso de P1 presentaba los conceptos básicos de programación (estructuras de control, variables, arrays) y de objetos (clases, relaciones) y en el de P2 se hacía énfasis en diseño orientado a objetos más avanzado, incluyendo también algoritmia sobre matrices, archivos, interfaz gráfica y persistencia.

La evaluación de ambos cursos incluía un parcial individual, actividades de clase y dos trabajos obligatorios de un mes de duración, realizados en equipos de dos estudiantes. Cada trabajo obligatorio tenía “defensa” individual. La defensa implica la implementación de una modificación en el código fuente sobre la funcionalidad del trabajo entregado. La no resolución del cambio implica la pérdida de puntos del trabajo.

Cada asignatura se aprobaba en dos niveles: curso y examen. Para aprobar el curso se requerían 70% o más de los puntos. Quienes aprobaban el curso, tenían derecho a dar el examen. En el caso de obtener 86% o más en el curso, se consideraban exonerados del examen. La carga horaria de las asignaturas era de 4 horas semanales en salón de clases y 2 horas en laboratorio, durante 15 semanas.

En el plan nuevo, vigente desde 2019, la materia P1 fue revisada y rediseñada, teniendo como contenidos temáticos los conceptos básicos de Programación con un enfoque multiparadigma implementando en JavaScript. Se presentan diferentes paradigmas de programación: imperativo (incluye programación orientada a procedimientos e introducción a objetos), declarativo (por ejemplo, se presentan algunas nociones de programación funcional) e impulsado por eventos. Los temas detallados son: pseudocódigo, variables, expresiones aritméticas y lógicas, estructuras de control, funciones, arrays, nociones de programación web (con HTML: lenguaje de marcado de hipertexto y CSS: hojas de estilo en cascada) e introducción a objetos y clases. La forma de evaluación se mantuvo relativamente similar: se agregó un parcial más.

El formato elegido para el dictado de P1 fue de aula invertida. En el primer dictado del curso de P1 en marzo 2019, con 299 estudiantes en 12 grupos y un total de 20 profesores, no se observaron diferencias en cuanto a los resultados generales de aprobación y de deserción en relación a años anteriores: 67% de aprobación, valor que coincide con el promedio de la misma materia de los 10 años anteriores. Los

estudiantes manifestaron en encuestas anónimas una percepción positiva del curso y metodología. Los docentes también manifestaron su conformidad hacia toda la propuesta [5].

B. Aula Invertida

Bishop y Verlenger [7] definen el aula invertida como una técnica educacional que consiste en dos partes: actividades grupales interactivas que se desarrollan en la clase y actividades individuales en línea fuera de clase. El uso del aula invertida involucra a los estudiantes en un aprendizaje activo y desarrolla el pensamiento crítico y nuevas habilidades de estudio [8]. La evidencia sugiere que los estudiantes aprenden en el modo de aula invertida igual, o mejor que simplemente tomando clases magistrales [9].

Las clases de programación según Kung y Kung [10] son candidatas perfectas para aplicar el enfoque de aprendizaje invertido ya que se requieren muchas actividades prácticas para ejercitar los principios de programación y dominar un lenguaje de programación específico.

En muchos casos los estudiantes prefieren este formato invertido. En su reporte, Kung y Kung [10] refieren a que aproximadamente el 70% de los estudiantes prefieren el enfoque invertido al tradicional. Guerrero et al refieren 78% de estudiantes satisfechos con la metodología invertida [11]. A su vez, Indi [12], en su estudio sobre el uso de clases invertidas para enseñar Java (estrategia utilizada en 15% de su curso) señala que a 98% (de 60 estudiantes) les gustó la estrategia de aula invertida y también que les ayudó a entender los conceptos la discusión posterior en clase.

Preparar el curso en formato invertido es trabajoso, pero permite uniformizar. Como señalan Guerrero et al, el tiempo y trabajo que lleva el aula invertida y el aprendizaje basado en problemas son arduos [11], aunque cuando ya están prontos, servirán para siguientes semestres. Tyler y Abdrakhmanova [13] destacan que el uso del aula invertida permite realizar la entrega del curso en una forma consistente en múltiple grupos y a través de los distintos semestres.

Prevalla y Uzunboyly [8] señalan en su trabajo como línea futura de investigación si el uso de esta pedagogía pierde efectividad en cursos consecutivos, a medida que los estudiantes se hacen más familiares con ella. Respecto al uso en siguientes cursos, hay posiciones diversas. Indi [12] refiere que la mayoría de los estudiantes querrían que se use en otras materias. Por otro lado, Lee y Lee, señalan que los estudiantes indicaron que el modelo de aula invertida lleva más tiempo y esfuerzo de su parte y más de la mitad de ellos preferirían no tener en cursos siguientes el mismo modelo [14].

Experiencias realizadas en cursos consecutivos con aula invertida presentan resultados disímiles: Hayashi et al [15] destacan la efectividad de este pedagogía en cursos de C y luego de Java, y por otro lado, Tyler y Abdrakhmanova [13], en el caso de esta misma metodología invertida en P1 y P2, indican que no detectaron diferencias significativas en los resultados de examen comparando antes y después del uso de aula invertida.

En resumen, el formato de aula invertida es aplicable a cursos de Programación y hay variedad de perspectivas en cuanto a su uso y efectividad en cursos consecutivos. El objetivo de este trabajo es intentar aportar en la discusión sobre estos puntos en el caso específico de materias consecutivas del área inicial de enseñanza de Programación.

III. NUEVO CURSO DE PROGRAMACIÓN II

En esta sección se describe el contenido temático de P2 en el nuevo plan. También se describe cómo se diseñó el curso, su primer dictado y la evaluación.

A. Contenido temático

Los objetivos de esta materia son continuar la formación en el área de programación, con fuerte énfasis en la POO y en algoritmia más avanzada y desarrollar aplicaciones orientadas a objetos. Los logros esperados son conocer y comprender los principales conceptos de la POO, y programar aplicaciones de forma robusta, correcta y eficiente en ese paradigma.

Los temas a tratar incluyen:

- Nociones de orientación a objetos. Modularidad, re- uso, encapsulación.
- Nociones de análisis y diseño orientado a objetos. Concepto de dominio del problema e interfaz de usuario.
- Conceptos básicos de orientación a objetos: Clases, objetos. Mensajes, métodos. Jerarquía de clases. Relaciones: Asociación, Agregación, Herencia. Concepto de “this” y “super”.
- Polimorfismo. Copia de objetos. Igualdad e identidad. Notación UML (Unified Modelling Language).
- Conceptos de Programación: Parámetros. Matrices. Manejo avanzado de colecciones. Archivos.
- Depuración de programas. Uso de estándares y estilo de programación. Excepciones: manejo y propagación.
- Interfaz gráfica.
- Manejo de persistencia.

Al igual que en cursos previos de P2, el entorno de desarrollo elegido fue NetBeans [16], el cual es sencillo de instalar, funciona en distintas plataformas y es fácil de usar [17]. La versión de Java utilizada es la 11 [18].

B. Preparación del curso y metodología

La preparación del curso fue realizada entre fines de 2018 y julio 2019. Incluyó el rediseño completo del curso, elaboración de todos los materiales, rediseño del sitio web y preparación de los docentes. El rediseño del curso estuvo a cargo de un equipo integrado por la Catedrática, dos docentes con alta experiencia y dos docentes más que colaboraron en las filmaciones de los videos así como en la revisión y chequeo de todos los materiales.

Se elaboraron 32 videos, de una duración entre 4 y 8 minutos cada uno e incluyen animaciones y código (ver Fig. 1 Ejemplo de video). Corresponden a 28 temas. Se diseñaron 16

cuestionarios. Se redactó un libro con todo el contenido teórico de 150 páginas y otro con el contenido práctico, con 12 prácticos y sus soluciones. Todo el material estuvo disponible desde el comienzo del curso en el sitio web de la materia en la plataforma Moodle [19]. Para simplificar el pasaje entre asignaturas, se mantuvo la misma estética y organización del sitio utilizados en el curso de P1.



Fig. 1 Ejemplo de Video

El plan clase a clase estuvo disponible al comienzo del curso, y se fue actualizando permanentemente durante el semestre. Incluye número de clase, lugar de dictado (salón o laboratorio), temática principal, ejercicios y actividades a realizar (por ejemplo, uso de Kahoot [20] o Quizizz [21] (Ver Fig. 2 Plan). También contiene la indicación de las tareas y videos para la siguiente clase.

Clase 2	Salón	INTRD. JAVA	Habilitar Tarea 1
		Leer datos e indicar si paso el 37 (uso de var boolean) Codificación	Java basico
		Quizizz Sem 01 de Introducción a Java	
		Operadores y sentencias (ppt en materiales) (Reposo de estructuras)	
		Opcional: Ppt con Cuestionario sobre definición de variables y corrida a mano (en materiales)	
		Reposo de Casting (El tema se ve en el video 3) (ppt en materiales)	
		Ejemplos de Casting (doc en materiales)	
		Ej. Calcular raíces ecuación 2do grado (uso Scanner, Float, Math. Nombrar Math sqrt, max, min, pow)	
		Ej de descubrir que es el código (resp Tablas de Multiplicar, con errores de indentación y líneas) (Ver mat docente)	
		Recomendaciones de estilo: uso de comentarios, indentación, llaves, nombres memotónicos, unico return, una sentencia por línea, una variable por línea + ejercicio (doc y ppt en materiales)	
		Ejercicio de Prueba de programas (ppt en materiales, tiene tambien un ej. de arrays)	

Fig. 2 Plan detallado

Previo al dictado, se hizo un “Taller de preparación” dirigido a docentes y asistentes docentes, donde se discutieron aspectos del nuevo curso y se repasaron los conceptos a presentar. Se realizó tutoría semanal a los docentes nuevos o con menos experiencia. También se dispuso de 9 horas semanales de apoyo a los estudiantes fuera del horario de clases, contando con estudiantes más avanzados de la carrera.

C. Dictado del curso de P2

La mecánica utilizada durante todo el curso de P2 fue indicar a los estudiantes para la siguiente clase qué video/s debían verse y, o, cuáles actividades debían ser resueltas y en la propia clase discutir la temática, con otros ejemplos y ejercicios. Esta metodología ya era conocida por los estudiantes por haber sido aplicada en el curso de P1. Al igual que en ese curso, se tuvo especial cuidado en evitar la sobrecarga de tareas, siguiendo la sugerencia de Weerasinghe [22].

Como ejemplo de actividades previas, durante las primeras semanas de familiarización con Java y el uso de

matrices, además de cuestionarios se propusieron ejercicios en la plataforma online Hackerrank [23]. En esta plataforma, el docente puede definir problemas de programación, indicar los datos de prueba y cada estudiante, realizar la codificación y testeado en la propia plataforma. Se eligió que los datos de prueba de cada ejercicio estén ocultos a los estudiantes, con el objetivo de fomentar el análisis del código y el estudio detallado de los casos requeridos. La ventaja adicional de utilizar esta herramienta es que frecuentemente es usada en entrevistas laborales de la industria del software [24] y acerca un poco más a los estudiantes a la realidad del mercado.

Al final de cada clase, se hizo un cierre destacando los principales puntos. Además, los docentes completaron luego de cada clase, una planilla con el avance de cada grupo, con el objetivo de documentar y simplificar la coordinación de los 9 grupos de clase con 13 docentes en total.

D. Evaluación

En relación a la evaluación, el esquema es similar al de los cursos previos de P2, consistiendo en actividades, parcial individual y dos trabajos en equipo de dos estudiantes de un mes de duración, denominados “Obligatorios”. En cursos anteriores, los trabajos de larga duración trataban un mismo problema, con dos entregas. Tradicionalmente era un juego abstracto con matrices. En la primera entrega, se realizaba la implementación utilizando la consola de comandos y en la segunda, se cambiaba a interfaz Windows, se agregaba el manejo de archivos, persistencia y cambios menores en la lógica. Los estudiantes ya manejaban Java desde el primer semestre y habían experimentado con herencia, colecciones y demás elementos en el curso previo.

En el curso 2019, el formato de estos trabajos se modificó. El primero mantuvo el estilo orientado a la gamificación y requirió la implementación de un juego abstracto de estrategia sobre 2 tableros ubicados uno encima del otro (ver Fig. 3), es una adaptación del “Shobu” [25]. En el escenario de juego participan dos jugadores. Cada jugada consiste en dos partes: preparación (que se realiza en un tablero y consiste en mover una ficha propia) y su réplica (que repite la misma dirección y sentido del movimiento en otra posición del otro tablero y puede eliminar fichas del tablero). Los turnos se alternan y termina el juego cuando no hay movimientos posibles o no hay más fichas. Se requieren 3-4 clases para su implementación. El juego implica el uso de algoritmos de matrices, uso de interfaz de consola con manejo de color y múltiples validaciones.

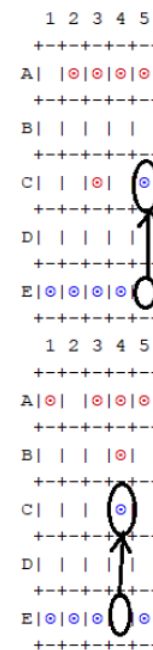


Fig. 3. Primer Obligatorio: Juego

Con el objetivo de que los estudiantes desarrollen más código en Java (pero sin aumentar la complejidad general), se decidió modificar el estilo del segundo obligatorio. Así, el segundo trabajo obligatorio consistió en realizar una herramienta para la gestión del Concurso de Programación. El Concurso es una actividad académica que se realiza todos los años en la Facultad de Ingeniería de la Universidad ORT Uruguay, similar al Concurso Internacional de Programación [26]. El sistema a realizar debía permitir gestionar estudiantes, equipos, profesores, ejercicios y envíos de resoluciones de ejercicios (individual y en lotes) (Ver Fig 4). Debía informarse del estado del Concurso en todo momento. La implementación incluía el diseño de varias clases y su codificación completa (no se reutilizaba nada del primer obligatorio), uso de herencia, manejo de interfaz gráfica con múltiples ventanas, procesamiento de archivos, persistencia y algoritmos interesantes para el control de envíos y tiempos.

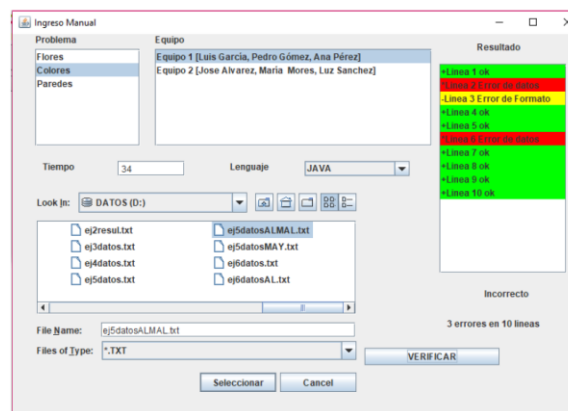


Fig 4. Segundo Obligatorio: Gestión del Concurso

Para la corrección de todas las tareas se utilizaron rúbricas. Su uso tiene por objetivo estandarizar la corrección. Además, cada obligatorio fue corregido por dos docentes y se unificó el resultado. Si bien en la Cátedra ya se utilizaban rúbricas, el nivel de detalle de las de este curso se ha incrementado. A modo de ejemplo, para la corrección del primer obligatorio, entre otros aspectos la rúbrica incluyó los ítems: diseño de UML de clases, atributos y relaciones; evidencia de testing, calidad del código y estilo de codificación, separación del dominio e interfaz y presentación de la documentación. Relativos a la ejecución en sí, se destaca evaluar la funcionalidad de los siguientes puntos: menú, alta de jugador, configuración de la partida, juego en sí (validaciones, jugadas válidas y no válidas, manejo de turnos, terminación), manejo del ranking y gestión de excepciones. Cada uno de estos ítems tiene un cierto peso y se evalúa en 3 posibles niveles: completo (100%), parcial (50%) o no disponible/incorrecto (0%). Por cada ítem además se puede agregar un comentario. En la Fig. 5 se muestra una imagen parcial de la rúbrica. La planilla calcula el total y destaca los ítems faltantes de evaluación.

	100%	50%	0%	
UML clases	Contiene todas las clases, solo dominio ok	1 Incluye clases de interfaz o clase prueba o faltan clases	0	Muy incompleto o no esta
UML atributos	Atributos ok (visibilidad, formato, tipo)	0 Atributos parcialmente bien	1	No pone atributos
UML relaciones	Relaciones todas ok	0 Relaciones parcialmente ok	1	No pone relaciones
Evidencia de testing	Estan las 3 partidas, cada una con 4 jugadas, indicando todo	0 Faltan partidas o no detalla que ingresar	0	Muy incompletos o no estan
Listado completo fuentes	Esta completo	1 Parcial	0	No esta
Estilo codificación y std, comentarios	Correctos	1 Parcial	0	Estilo incorrecto, no sigue stds, no comentarios
Nombres de autores en código	Esta en todos	1 Algunos con nombre	0	No pone nombres
Separar dominio interfaz	Correcta separación	0 Parcialmente separado	0	No separado
Repetir código	No repite código	0 Repite algo de código	0	Mucho código repetido
Presentación general PDF	Correcta, unico pdf	0 Varios pdf	0	No hay pdf
Caratula	Esta y completa	0 Incompleta	0	No hay caratula
Índice	Esta y completo	0 Incompleto	0	No hay índice

Fig. 5. Rúbrica para el 1er. obligatorio (imagen parcial)

El parcial individual consistió en 3 ejercicios: uno de 5 puntos de aplicación de conceptos teóricos, uno de 20 puntos con manejo de archivos y matrices y el tercero de 20 puntos orientado al diseño de clases y algoritmia. El estilo y nivel del parcial es similar al de cursos previos. Lo mismo sucede con el examen.

IV. RESULTADOS

Al comenzar el primer dictado del curso de P2 con los nuevos contenidos y formato, los estudiantes, en su gran mayoría, ya habían experimentado el aula invertida: según la primera encuesta anónima no obligatoria realizada en la semana 4, de los 209 estudiantes en P2, 149 estudiantes respondieron y de ellos, solamente 20 no cursaron la materia previa con el formato actual, o sea, 13.4% de quienes respondieron. En ese momento del curso, 118 de los 149 estudiantes (79.2%) indicaron que la formación recibida en Programación I era completa o mayormente suficiente (ver Fig. 6: Formación recibida).

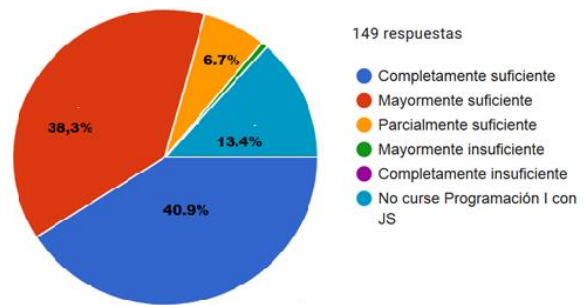


Fig. 6 Formación recibida en el curso de Programación I para seguir el curso de Programación II

De los 209 estudiantes del curso de P2, aprobaron 166 (79.4%). De ellos, 110 estudiantes (66.3%) obtuvieron 86% o más del curso, por lo cual quedaron exonerados de dar el examen. Se observan porcentajes relativamente similares de aprobación/eliminación y exoneración en comparación con los últimos 3 dictados del segundo semestre del año del plan anterior (ver Fig. 7 Resultados del curso 2016-2019). En la Tabla I (ver Tabla I Resultados Detallados), se presentan los datos completos de aprobación, exonerados dentro de los aprobados y eliminados, incluyendo el total de estudiantes.

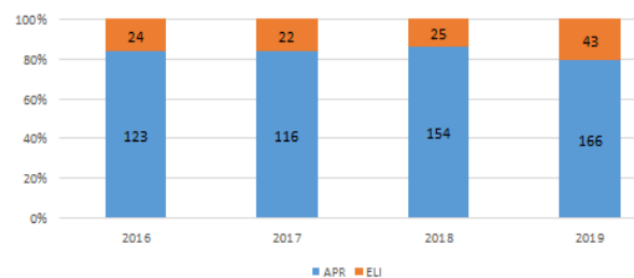


Fig. 7 Resultados curso Programación II - 2016-2019 (APR: aprobado, ELI: eliminado)

TABLA I
RESULTADOS DETALLADOS DE 2016 A 2019

Plan:	Anterior			Nuevo
Año	2016	2017	2018	2019
Total de estudiantes	147	138	179	209
Aprobados	123	116	154	166
Aprobados que exoneran	88	71	92	110
Eliminados	24	22	25	43
% aprobación	83.7%	84.1%	86.0%	79.4%
% aprobados que exoneran	71.5%	61.2%	59.7%	66.3%
% exoneración global	59.8%	51.5%	51.4%	52.6%
% eliminación	16.3%	15.9%	14%	20.6%

En la segunda encuesta anónima no obligatoria realizada al final del curso, sobre 115 respuestas, los estudiantes evaluaron los materiales como excelente/muy bueno/bueno (EMB) en 94.7% (109 estudiantes), los videos les resultaron EMB para 90.5% (104 estudiantes), el formato de aula

invertida le resultó EMB para el 87.8% (101 estudiantes), el uso de Java le pareció EMB para 95.6% de los estudiantes (110 estudiantes), y su aprendizaje de la programación fue evaluado como EMB por 99.1% (114 estudiantes) de los estudiantes que respondieron (ver Tabla II: Segunda Encuesta).

TABLA II
SEGUNDA ENCUESTA - FINAL DEL CURSO

115 respuestas	Excelente	Muy Bueno	Bueno	Regular	Malo	No opina
Materiales	23.5% (27)	52.1% (60)	19.1% (22)	3.5% (4)	0.9% (1)	0.9% (1)
Videos	26.1% (30)	34.8% (40)	29.6% (34)	7.8% (9)	0% (0)	1.7% (2)
Aula invertida	27% (31)	33% (38)	27.8% (32)	8.7% (10)	0.9% (1)	2.6% (3)
Uso de Java	52.1% (60)	33.9% (39)	9.6% (11)	3.5% (4)	0% (0)	0.9% (1)
Aprendizaje	26.1% (30)	48.7% (56)	24.3% (28)	0% (0)	0% (0)	0.9% (1)

La cantidad total de visualizaciones de los 28 temas presentados en los videos fue 5974, considerando 209 estudiantes da un promedio de 1.02 visualizaciones por tema/estudiante.

En relación al planteo de Prevalla y Uzunboyly [8] sobre la posible pérdida de efectividad de la metodología al aplicarla en cursos consecutivos, es de destacar que, si bien en el curso de P1 la conformidad con el uso de aula invertida fue de 95.6% (111 de los 116 estudiantes que respondieron la encuesta anónima final) [5] y en este curso fue un poco menor (87.8%), la percepción de su aprendizaje como EMB mejoró de 92.2% en P1 [5] a 99.1% en P2. Los docentes, a través de entrevistas, expresaron mejoras: “Los estudiantes aprovechan mucho más los recursos disponibles en el sitio porque ya vienen acostumbrados del curso de P1. Se nota un involucramiento mayor que en la modalidad anterior. En cuanto a la motivación, llegan esperando contar con los mismos recursos que usaban en P1, al notar que los siguen teniendo se sienten más seguros.” refiere uno de los docentes en las entrevistas realizadas al finalizar el semestre (Docente 1). “Considero que llegan igual de preparados o incluso más” señala el Docente 2. “Los estudiantes llegan con un mayor conocimiento. Se nota en clase un mejor desarrollo de los conceptos y mayor profundidad.” refiere el Docente 4.

Asimismo, los docentes se sienten motivados y conformes con el curso. “A mí me encantó. Me pareció mucho más motivante la preparación y dictado de este curso que el P2 anterior” indica el Docente 4. También, el Docente 2 señala: “El curso me parece que es más dinámico y en lo personal me agrada más. Al ser más dinámico genera otro tipo de interacción en las clases lo que en lo particular me gusta mucho”. “La clase es mucho más dinámica y divertida” expresa el Docente 1.

Al finalizar el curso, se volvió a revisar con los docentes la temática: “Los temas cubiertos son adecuados. En dos semestres logran otro lenguaje, fortaleciendo los principios y

conceptos de POO y sin perder temas en relación a los cursos anteriores” (Docente 4). “Se suprimieron esfuerzos en aquellas cosas que se consideró que agregan menos valor y se reforzaron otras necesarias. El curso ha quedado más dinámico en mi percepción. Los temas cubiertos creo que fueron mejorados con respecto al curso anterior” indica el Docente 3.

La planificación también fue un aspecto destacado en las entrevistas con los docentes: “La planificación previa, rúbricas y demás aspectos del diseño instruccional permiten unificar y tener una guía excelente para poder planificar la clase día a día” (Docente 4).

Durante todo el curso, los docentes fueron tomando nota de posibles mejoras al curso. Algunas de ellas fueron: abreviar la parte de introducción a Java en las primeras semanas, evaluar intercambiar el tema de herencia y colecciones si fuera necesario según la temática de los obligatorios (por ejemplo, el primer obligatorio no requirió herencia, por lo cual se puede presentar ese tema después) y agregar más ejercicios para el práctico (para tener para el caso de estudiantes que ya manejan Java o que resolvieron todos los ejercicios planteados).

V. CONCLUSIONES Y TRABAJO FUTURO

En el año 2019, se implementó en la Facultad de Ingeniería de la Universidad ORT Uruguay un nuevo plan de estudios, y en particular, en el área de Programación, se realizaron varios cambios, no sólo de los contenidos sino fundamentalmente en la metodología y modalidad de enseñanza-aprendizaje, pasando a aula invertida. En el primer semestre del año se dictó Programación I y en el segundo Programación II. La mayoría de los estudiantes del curso de Programación II habían cursado el semestre previo el nuevo de Programación I, por lo cual, ya estaban familiarizados con la metodología de las clases invertidas.

Analizando los resultados del nuevo curso, los resultados de aprobación (alrededor de 80%) están en los valores tradicionales de los cursos en nuestra universidad. Según las encuestas anónimas, los estudiantes perciben positivamente el curso de Programación II con aula invertida así como adecuada la formación recibida en la asignatura previa, con el mismo formato. Los docentes manifestaron estar de acuerdo con la propuesta y se mostraron involucrados en la adopción de la modalidad de aula invertida.

El nivel alcanzado de conocimiento de programación y de Java por los estudiantes es similar a los cursos previos, considerando las evaluaciones realizadas en relación a años anteriores. Es de destacar que al final de los dos cursos, programan utilizando varios paradigmas: por un lado el orientado a objetos y por otro, imperativo, declarativo e impulsado por eventos (del curso de Programación I) y manejan dos lenguajes (JavaScript, Java), por lo cual, los conocimientos y habilidades adquiridas luego de los dos cursos con formato de aula invertida son más amplios en relación al plan anterior con el formato tradicional.

Como trabajo a futuro, se planea continuar realizando el seguimiento a los estudiantes en las asignaturas siguientes de

Algoritmos y Estructuras de Datos así como implementar las oportunidades de mejora detectadas en el curso.

AGRADECIMIENTOS

Para la implementación del cambio del diseño de los cursos de Programación I y Programación II, se contó con el apoyo del “Fondo de Fomento a las Ingenierías” de la Agencia Nacional de Investigación e Innovación (ANII) [27].

REFERENCIAS

- [1] R. Pessoa, G. Lisboa y T. Pontual, “A Systematic Literature Review on Teaching and Learning Introductory Programming in Higher Education”, *IEEE Trans. of Education*, Vol 62, No. 2, Mayo 2019
- [2] P. Feijóo-García y C. Ortiz Buitrago, “The Godparent Plan: A Pedagogical Strategy for CS1 Accompaniment and CS2 Pedagogical Enhancement”, *International Journal of Engineering Pedagogy*, 8(1), 43-55, 2018.
- [3] J. Bergmann y A. Sams, “Flip your classroom: reach every student in every class every day”, USA: ISTE ASCD, 2012
- [4] L. Zhuo y G. Qi-xian, “The application of hybrid flipped classroom in the course of Java Programming”, 2015 Int. Conf. on Information Technology in Medicine and Education, 2015
- [5] I. Kereki y A. Adorjan, “Flipped classroom in a CS1 course”, *Proc. de IEEE Educon 2020*, Portugal, 2020.
- [6] Y. Wei, Y. Shi, H. Hao y J. Liu, “Blended learning versus traditional learning: a Study on student’s learning achievements and academic press”, *Int. Symp. on Educational Technology*, p. 219-223. DOI 10.1109/ISET.2017.57, 2017
- [7] J. Bishop y M. Verleger, “The Flipped Classroom: a Survey of the Research”, 120th ASEE Annual Conference & Exposition, 2013
- [8] B. Prevalla y H. Uzunboylu, “Flipped Learning in Engineering Education”, *TEM Journal*, V 8, Issue 2, DOI 10.18421/TEM82-46, Mayo 2019
- [9] C. Reidsema, L. Kavanagh, R. Hadgraft y N. Smith (editores), “The Flipped Classroom, practice and practices in Higher Education”, Springer Nature Singapore, DOI 10.1007/978-981-10-3413-8-1, 2017
- [10] H. J. Kung y L. Kung, “Flipped Learning Activities for an introduction to Java Course”, *Issues in Information Systems*, 18(3), 29-39, 2017
- [11] W. Guerrero, H. Sánchez y M. Rico, “Aula invertida y el aprendizaje basado en problemas en la enseñanza universitaria”, 2017 IEEE Int. Conf. on Information Systems and Computer Science, 2017
- [12] T. Indi, “An experience report of flipped classroom strategy implementation for Java Programming Course”, 2016 IEEE 8th Int. Conf. on Technology for Education, 2016
- [13] B. Tyler y M. Abdrakhmanova, “Flipping the CS1 and CS2 classrooms in Central Asia”, 46th Annual Frontiers in Education Conference, FIE 2016 - Erie, United States, 2016
- [14] G. Lee y P. Lee, “Data structures in Flipped Classroom: Students’ effort and Preference”, 2015 Int. Conf. on Learning and Teaching in Computing and Engineering, DOI 10.1109/LaTiCE.2015.28, 2015
- [15] Y. Hayashi, K. Fukamachi y H. Komatsugawa, “Collaborative Learning in Computer Programming Courses that adopted the Flipped Classroom”, 2015 Int. Conf. on Learning and Teaching in Computing and Engineering, DOI 10.1109/LaTiCE.2015.43, 2015
- [16] NetBeans, <https://netbeans.apache.org/>
- [17] J. Morgan, “The best Java IDEs and What to Expect From Them” <https://dzone.com/articles/the-best-of-java-ides-amp-what-to-expect-from-them>
- [18] Java, <https://www.oracle.com/technetwork/java/javase/downloads/index.html>
- [19] Moodle, <https://moodle.org/>
- [20] Kahoot, <https://create.kahoot.it>
- [21] Quizizz, <https://quizizz.com/>
- [22] T. Weerasinghe, “An evaluation of different types of blended learning activities in higher education”, 2018 IEEE Int. Conf. on Advanced Learning Technologies, 2018.
- [23] Hackerrank, <https://www.hackerrank.com/>
- [24] Hackerrank Jobs, <https://www.hackerrank.com/jobs/search>
- [25] Shobu game, <https://www.smirkandlaughter.com/shobu>
- [26] International Collegiate Programming Contest - ICPC <https://icpc.baylor.edu/regionals/abouticpc>
- [27] ANII: Agencia Nacional de Investigación e Innovación de Uruguay, <https://www.anii.org.uy/noticias/104/las-cuatro-facultades-de-ingenieria-apuestan-al-desarrollo-del-pais-aumentando-la-cantidad-de-ingenieros-egresados/>