

Aplicación del modelo pedagógico «Flipped Learning» para el aprendizaje de programación

Una experiencia en la Universidad Nacional de Luján

Juan M. Fernandez¹, Rosana Matuk¹, Matías Rodríguez¹, Mario Quiroga¹

¹Departamento de Ciencias Básicas, Universidad Nacional de Luján
{jmfernandez, rmatuk, mrodriguez, mquiroga}@unlu.edu.ar

Resumen

La pandemia declarada en el año 2020 por la propagación de un nuevo coronavirus, el SARS-CoV-2, generó una situación en extremo preocupante desde el punto de vista sanitario a nivel mundial. Sin embargo, provocó un momento de disrupción en los sistemas educativos de todos los niveles y en todo el mundo, los cuales debieron adoptar, sin ningún tipo de planificación previa ni preparación, la modalidad remota para continuar con el proceso educativo. A partir de este momento, y paulatinamente conforme la preocupación por la pandemia comenzó a disminuir, junto con la disminución del número de muertes y el avance de la vacunación, propició un proceso de reflexión que permitió abordar la discusión sobre el modelo netamente transmisivo instituido en la mayoría de los sistemas educativos y dar lugar a estrategias de enseñanza y aprendizaje que pongan al estudiante en el centro del proceso.

En este marco, tomaron mayor fuerza las denominadas metodologías activas de aprendizaje. En particular, existe un modelo pedagógico dentro de este conjunto de metodologías denominado de aula invertida o *flipped classroom*, cuya idea básica consiste en promover que el alumno trabaje por sí mismo y fuera del aula los conceptos teóricos a través de diversas herramientas que el docente pone a su alcance y el tiempo de clase se aproveche para resolver dudas relacionadas con el material proporcionado, realizar prácticas y abrir foros de discusión sobre cuestiones controvertidas.

En este trabajo, a partir de la sistematización del marco conceptual del modelo pedagógico

de *aula invertida* y el abordaje de una experiencia particular, se introduce una propuesta de organización y planificación de un curso para el aprendizaje de programación en el ámbito universitario bajo el paradigma de metodologías activas de aprendizaje.

Introducción

La pandemia declarada en el año 2020 por la propagación de un nuevo coronavirus, el SARS-CoV-2, que causa la enfermedad denominada COVID-19, obligó a todos los niveles educativos a adoptar, sin ningún tipo de planificación previa ni preparación, la modalidad remota para continuar con el proceso educativo [10]. Mientras que hasta fines de 2019 se podía estimar el número de usuarios de la Educación a Distancia en cerca de 200.000, hacia mediados de 2020, de manera formal o informal, se generalizó el uso de las TIC para mantener activa las actividades de enseñanza-aprendizaje, alcanzando los 2 millones de usuarios de educación a distancia en las universidades argentinas hacia mediados de 2020 [14].

Esta situación en extremo preocupante desde el punto de vista sanitario a nivel mundial generó un momento de disrupción en los sistemas educativos de todos los niveles y en todo el mundo. A partir de este momento, y paulatinamente conforme la preocupación por la pandemia comenzó a disminuir junto con la disminución del número de muertes y el avance de la vacunación, permitió a los equipos docentes reflexionar sobre la experiencia, los conocimientos y herramientas adquiridas durante la pandemia.

Este proceso de reflexión permite, junto con

las herramientas y conocimientos incorporados, profundizar la discusión sobre el modelo netamente transmisivo instituido en la mayoría de los sistemas educativos y dar lugar a estrategias de enseñanza y aprendizaje que pongan al estudiante en el centro del proceso.

Este tipo de metodologías responden a las denominadas de aprendizaje activo o «active learning». El aprendizaje activo consiste en actividades breves individuales o en grupos pequeños relacionadas con el curso que todos los estudiantes de una clase deben realizar, alternando con intervalos dirigidos por un instructor en los que se procesan las respuestas de los estudiantes y se presenta nueva información [9].

En particular, existe un modelo pedagógico dentro de este conjunto de metodologías denominado de aula invertida o *flipped classroom*, cuya idea básica consiste en promover que el alumno trabaje por sí mismo y fuera del aula los conceptos teóricos a través de diversas herramientas que el docente pone a su alcance y el tiempo de clase se aproveche para resolver dudas relacionadas con el material proporcionado, realizar prácticas y abrir foros de discusión sobre cuestiones controvertidas [5, 6].

En este contexto, este trabajo, a partir la sistematización del marco conceptual del modelo pedagógico de *aula invertida* y el abordaje de una experiencia particular, introduce la definición de una propuesta de organización y planificación de un curso para el aprendizaje de programación en el ámbito universitario bajo el paradigma de metodologías activas de aprendizaje. Además del marco metodológico de organización del curso, se propone y fundamenta la elección de un *stack tecnológico* particular para la implementación efectiva.

Marco teórico

La transición desde un modelo educativo presencial y basado en la transmisión vertical de conocimientos, hacia un modelo más horizontal e interactivo de aprendizaje, y realizado en forma remota, supone un gran “cambio cultural” para la Universidad como institución educativa. Entre los pilares fundamentales de dicho cambio se encuentra la llamada “renovación metodológica”. Aunque los resultados de diferentes

investigaciones muestran que no existe un método “mejor” que otro de forma absoluta, sí aportan algunas conclusiones interesantes y a tener en cuenta: para los objetivos de bajo nivel, por ejemplo, adquisición y comprensión de la información, cualquier método es adecuado y equivalente. Para los objetivos superiores, por ejemplo, desarrollo del pensamiento crítico y aprendizaje autónomo, los métodos centrados en los alumnos son más adecuados y eficaces [15].

Así, se puede afirmar que los métodos de enseñanza con participación del alumno, donde la responsabilidad del aprendizaje depende directamente de su actividad, implicación y compromiso son más formativos que meramente informativos, generan aprendizaje más profundos, significativos y duraderos y facilitan la transferencia a contextos más heterogéneos [15].

En cuanto a los métodos de enseñanza, pueden ser situados en un continuo. En este sentido, Brown y Atkins [3] realizan una especie de tipificación en la que clasifican los diferentes métodos de enseñanza siguiendo este criterio. En un extremo están las lecciones magistrales en las cuales la participación y el control del estudiante son mínimos. En el otro extremo, estaría el estudio autónomo en el cual la participación y control del profesor son usualmente mínimas.

Por otro lado, transcurre la era digital y sus características hacen imprescindible adaptar las técnicas pedagógicas a las nuevas realidades y a los nuevos alumnos. Actualmente existen lo que algunos autores [4] denominan «e-alumnos», esto es, personas que dentro y fuera de las aulas emplean las nuevas tecnologías como herramientas para su aprendizaje. Se trata de alumnos muy visuales y acostumbrados a la multitarea, esto es, alumnos que revisan su correo electrónico mientras están en clase o que ven decenas de vídeos al día en canales como YouTube, y que muestran dificultades para prestar atención al profesor durante la hora u hora y media que expone su tradicional discurso magistral [4].

De hecho, existen estudios que cuantifican esta situación. Por ejemplo, un estudio en particular [21] muestra que, de las 200 palabras por minuto que puede hablar un profesor, el alumno capta alrededor de la mitad; los alumnos retienen el 70% de lo que se explica en los diez primeros minutos de clase y tan sólo un 20% de

lo explicado en los diez últimos, permaneciendo atentos sólo alrededor del 40 % del tiempo que dura la clase.

Por ello, los equipos docentes deben analizar cuál es actualmente la mejor manera de transmitir sus conocimientos para conseguir que el alumno asimile adecuadamente los contenidos y saque el mayor provecho a las horas que invierte en su aprendizaje.

En este contexto, el aula invertida o *flipped classroom*, es un modelo pedagógico que, bien utilizado, puede contribuir en gran medida a este fin [4].

El modelo de aula invertida surge en el contexto de un nuevo paradigma o conjunto de metodologías educativas denominadas aprendizaje activo o «active learning». El aprendizaje activo consiste en actividades breves individuales o en grupos pequeños relacionadas con el curso que todos los estudiantes de una clase deben realizar, alternando con intervalos dirigidos por un instructor en los que se procesan las respuestas de los estudiantes y se presenta nueva información [9].

Puntualmente *Flipped Classroom* o aula invertida en español, nace como modelo pedagógico de la mano de Jonathan Bergmann y Aaron Sams, dos profesores de química de la Woodland Park High School en Colorado (EEUU) [5]. La motivación que dió origen a esta metodología fue conseguir que los alumnos, que por diversos motivos no habían podido asistir a clase, fueran capaces de seguir el ritmo del curso y no resultaran perjudicados por la falta de asistencia. Para ello, decidieron grabar los contenidos docentes a través de un software que permitía capturar en vídeo las presentaciones narradas, y distribuir las entre sus alumnos. Sin embargo, poco a poco se dieron cuenta de que las grabaciones no sólo las utilizaban aquéllos que no habían podido ir a clase, sino la generalidad de sus estudiantes. De este modo comenzaron a invertir su método de enseñanza remitiendo vídeos de las lecciones para que las visualizaran en casa antes de la clase y reservando las horas presenciales para realizar proyectos con los que poner en práctica los conocimientos adquiridos y resolver dudas relacionadas con la materia explicada.

Según estos autores -principales dirigentes de la Red de Aprendizaje Flipped- la «Flipped

Classroom» (aula invertida), o en términos más generales el «Flipped Learning» (aprendizaje invertido o aprendizaje al revés) es «un enfoque pedagógico en el que la instrucción directa se mueve desde el espacio de aprendizaje colectivo hacia el espacio de aprendizaje individual, y el espacio colectivo se transforma en un ambiente de aprendizaje dinámico e interactivo en el que el educador guía a los estudiantes a medida que se aplican los conceptos y puede participar creativamente en la materia» [6].

Otros autores, señalan que se trata de «un modelo didáctico en el cual los estudiantes aprenden nuevo contenido a través de videotutoriales en línea, habitualmente en casa; y lo que antes solían ser los “deberes” (tareas asignadas), se realizan ahora en el aula con el profesor ofreciendo orientación más personalizada e interacción con los estudiantes» [20].

La idea básica inherente a este modelo educativo consiste en promover que el alumno trabaje por sí mismo y fuera del aula los conceptos teóricos a través de diversas herramientas que el docente pone a su alcance, principalmente vídeos o *podcasts* grabados por su profesor o por otras personas (pero no exclusivamente), y el tiempo de clase se aproveche para resolver dudas relacionadas con el material proporcionado, realizar prácticas y abrir foros de discusión sobre cuestiones controvertidas. Y es casi cuatro veces más efectivo que los estudiantes tengan la oportunidad de practicar sus habilidades en clase con el *feedback* formativo del maestro, que la realización de tareas o deberes fuera del aula, porque en este último caso los profesores tienen pocas oportunidades de supervisión [21].

Es importante señalar que aunque con este método pedagógico el alumno trabaja de forma autónoma, nunca lo hace solo porque el equipo docente actúa de guía en su proceso de aprendizaje, seleccionando los contenidos que debe estudiar, asimilar y retener, poniéndolos a su disposición a través de diversos medios y estando en constante comunicación con él. No obstante, esta metodología propone un cambio de roles respecto al modelo tradicional ya que el alumno debe colaborar activamente en su propio aprendizaje [4].

Otra de las ventajas que posee el modelo pedagógico de *flipped learning* es que resulta ade-

cuado para el rango etario de estudiantes que actualmente están ingresando a la Universidad, entre los que se encuentran los denominados *millennials* [19]. Los estudiantes *millennials*, denominados «nativos digitales» [18], han estado expuestos a la tecnología de la información desde una edad muy temprana. El acceso de los *millennials* a la tecnología, la información y los medios digitales es mayor que el de cualquier generación anterior. Las características de los *millennials* incluyen el acceso a la información las 24 horas del día, los 7 días de la semana, una preferencia por entornos que apoyan la multitarea y una inclinación hacia la actividad grupal y la apreciación de los aspectos sociales del aprendizaje. Esta generación se distingue por su acceso a experiencias tecnológicas y colaborativas. Los estudiantes *millennials* impulsan el cambio en los entornos de aprendizaje de todo el mundo [19].

Desarrollo

Tradicionalmente, la organización más clásica de los cursos universitarios versa en una clase teórica expositiva, donde el profesor transmite a los estudiantes los conocimientos que estos deben asimilar, seguida por algún espacio de puesta en práctica de estos saberes. No obstante, si bien la práctica más habitual y característica en la enseñanza universitaria es la clase teórica, esta estrategia, por sí sola, no es muy recomendable para el fomento de aprendizaje autónomo de los alumnos [17].

En este tipo de modelos bien tradicionales, como el modelo didáctico tradicional o transmisivo, los aspectos metodológicos, el contexto y, especialmente, el alumnado, quedan en un segundo plano. El conocimiento sería una especie de selección divulgativa de lo producido por la investigación científica, plasmado en los apuntes universitarios [16].

Desde el punto de vista de la implementación de estos modelos, tradicionalmente, como se plasma en la Figura 1, el equipo docente provee al alumnado un material bibliográfico de referencia o de producción propia que sustenta las clases expositivas periódicas del docente, donde los espacios de consulta y participación por parte de los estudiantes se circunscriben a pe-

queños espacios temporales de esa misma clase o una clase de corte más aplicada o práctica donde se invita a los estudiantes a aplicar los saberes transmitidos. No hay un espacio virtual que sustente esta actividad, y si lo hay es utilizado a modo de repositorio de archivos o sitio web y la interacción docente-estudiante se reduce básicamente al docente hablando al estudiante [16].



Figura 1: Esquema del flujo de intercambios en un modelo educativo transmisivo

Contexto de la propuesta

En este trabajo, se utiliza como caso de estudio la asignatura Introducción a la Programación, la cual forma parte del primer cuatrimestre del Plan de Estudios de la Carrera de Licenciatura en Sistemas de Información de la Universidad Nacional de Luján, que posee los siguientes descriptores o contenidos mínimos: «Resolución de problemas y algoritmos. Lenguajes interpretados y compilados. Estructuras de control: secuencial, alternativa e iterativa. Algoritmos elementales. Tipos de datos simples».

Anualmente se inscriben unos 600 estudiantes a esta asignatura, que se dicta en las Sedes de Luján y Chivilcoy. El equipo docente está conformado con 5 módulos simples de profesores (con un profesor con dedicación exclusiva y otro con dedicación simple), 4 módulos simples para cargos de Jefe de Trabajos Prácticos (dos docentes con dedicación equivalente a semi-exclusiva), 4 módulos simples de Ayudantes de primera (dos docentes con una dedicación simple y uno con una dedicación semi-exclusiva) y 4 ayudantes alumnos de carácter rentado, los cuales poseen -por reglamentación vigente- dedicación simple.

La asignatura objeto de estudio previo a esta experiencia estaba planificada bajo los preceptos metodológicos vertidos en los párrafos precedentes. La misma contaba inicialmente con dos clases semanales presenciales, la primera de

carácter teórico, donde se abordaban los contenidos teóricos y la segunda, de carácter práctico en laboratorio de computación, donde los estudiantes ejercitaban con la resolución de enunciados prácticos por medio de la formulación de algoritmos en el lenguaje procedural Pascal, que se resolvían por el docente previo a la culminación de la clase. A su vez, en los últimos años el equipo docente había desarrollado un apunte propio, basado en un libro de cabecera [8]. Para la evaluación del conocimiento y posterior calificación de los estudiantes existían dos parciales teórico-prácticos con posibilidad de recuperar uno de ellos.

Presentación de la propuesta

A lo largo del tiempo se han ido propiciando muchos cambios en relación a la forma de enseñanza de la programación, coexistiendo varios enfoques y tendencias. Aún hoy se puede verificar que no hay un consenso en los métodos a utilizar. Algunas de las razones son que no existe un único método para la resolución de algoritmos así como tampoco un enfoque didáctico para materias introductorias que se haya impuesto por sobre otros o demostrado una indiscutible efectividad. Se observa que hay métodos de enseñanza que se fundamentan a partir de un paradigma de programación en particular. A su vez, dentro de un paradigma determinado se visualizan varios enfoques para enseñar: algunos enseñan a programar en un lenguaje de programación particular, utilizando su sintaxis y su semántica, y otros emplean un lenguaje algorítmico lo bastante general como para permitir su traducción posterior a cualquier lenguaje de programación [11].

En un trabajo que aborda la revisión de la literatura respecto a la enseñanza de la programación [7] se identificaron cinco factores de éxito para enseñar cómo programar con un enfoque lúdico: la motivación intrínseca del estudiante, la integración y la participación en clase, el enfoque centrado en el alumno, la interacción y retroalimentación, y la fluida integración del contenido educativo en el juego [2, 13].

A partir de estos cinco factores de éxito, identificados también a partir de la evidencia empírica en el desarrollo de un curso de programación, se inició una transición metodológica en relación

al diseño del curso, la cual se plasma esquemáticamente en la Figura 2.

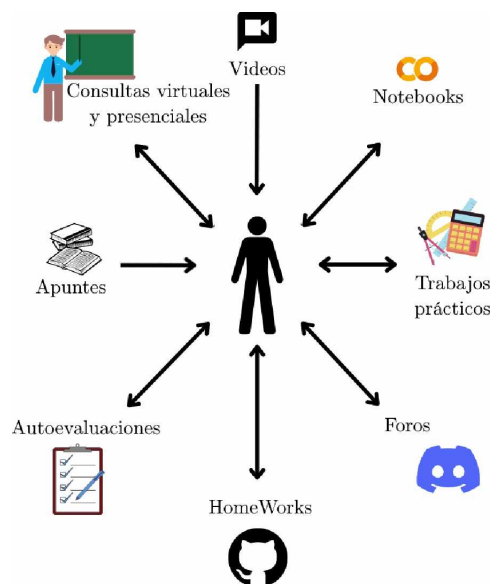


Figura 2: Esquema del flujo de intercambios en el modelo de aula invertida propuesto

En primer lugar, se decidió cambiar el lenguaje de programación, pasando a Python¹. Si bien Pascal [24] se muestra como un lenguaje muy adecuado para la enseñanza de la programación, se escogió Python por dos cuestiones: por un lado es un lenguaje ampliamente utilizado a lo largo de toda la Carrera, en diferentes áreas como la programación distribuida, aprendizaje automático y programación web; por el otro, es sumamente expresivo y con una sintaxis básica fácil de aprender. Sin embargo, la asignatura no es un curso de enseñanza de Python, sino de programación, lo que significa que los conceptos se materializan en este lenguaje pero se enseñan desde los fundamentos y no desde su sintaxis particular.

Luego, desde el punto de vista de organización de los materiales y componentes pedagógicos que conforman la propuesta, se estableció el Aula Virtual de la asignatura -única para las 8 comisiones de la asignatura- como el centro del ecosistema de aprendizaje y el espacio natural donde el estudiante accede a todos los conte-

¹Sitio oficial: www.python.org

nidos. Aquí, por un lado, se definió un apartado con todos los aspectos de organización de la asignatura, entre ellos el programa, una presentación de los miembros del equipo docente, el cronograma del curso, el acceso a la rúbrica o planilla de seguimiento de entregas, las condiciones de evaluación y las credenciales de acceso a los espacios virtuales que más adelante se detallan. A su vez, se define el recorrido didáctico separando a la asignatura en clases o unidades que se van habilitando semana a semana para establecer un ritmo común en la cursada y no sobrepasar a los estudiantes con la gran cantidad de material disponible. Estas clases semanales, a su vez, tienen secciones donde se presentan los diferentes videos embebidos en el aula con una introducción y una conclusión general, como se puede observar en la Figura 3.

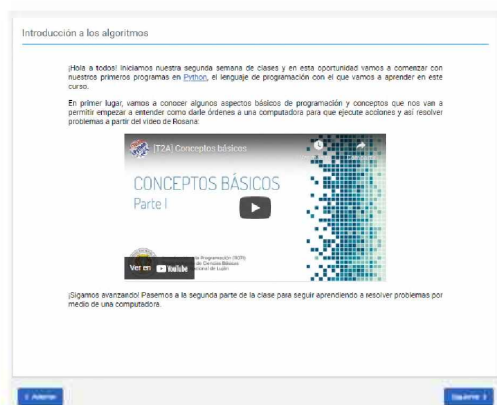


Figura 3: Ejemplo de clase en el Aula Virtual

En segundo lugar, las clases teóricas tradicionales se reemplazaron por videos y un apunte de clases propio de la asignatura. En este último caso, se tomó como referencia el apunte «Algoritmos y Programación I, Aprendiendo a programar usando Python como herramienta», 2da. Edición, de la cátedra de Algoritmos y Programación I de la Facultad de Ingeniería de la Universidad de Buenos Aires [23].

A su vez, las destrezas adquiridas durante la pandemia en cuanto a la grabación de videos permitieron grabar el material audiovisual, que se aloja en un canal de YouTube propio de la asignatura. Para la grabación de estos videos previamente se diseñó una guía con los temas a abordar en función del apunte de clases, se di-

vidió la tarea en el equipo docente y se definieron una serie de criterios para homogeneizar la producción, entre ellos la utilización de un *template* común, la definición de 15 minutos como el tiempo estándar de duración así como la idea de omitir referencias temporales del tipo «la próxima clase» o «la semana pasada» en el material.

Por su parte, se intentó horizontalizar la relación entre docentes y estudiantes a partir de la redefinición del rol docente y el cambio de perfil en la conformación del equipo. Por un lado, las clases magistrales pasaron a ser espacios de consulta, tanto presenciales como virtuales -otro aprendizaje de la pandemia-, donde los docentes fomentan la participación de los estudiantes a partir del planteo de dudas y aportes. Por otro lado, las coberturas generadas por licencias o bajas de integrantes del equipo se cubrieron con una gran cantidad de ayudantes alumnos, buscando acortar las distancias entre el plantel docente y los estudiantes y generando una dinámica y forma de comunicación más cercana.

A su vez, la vinculación entre docentes y estudiantes no se reduce únicamente a los espacios anteriores. Se definió un Servidor de Discord² de la asignatura con el objetivo de responder dudas que puedan surgir, generando distintos canales de acuerdo a las temáticas, tanto de texto como de audio y video. Por sus características, Discord es una aplicación muy popular entre los ingresantes a la Carrera, lo cual permite prescindir de la necesidad de aprender una nueva aplicación para comunicarse, al tiempo que fomenta su uso debido a la sensación de familiaridad para los estudiantes. Este espacio también sirve para generar sinergia entre los propios estudiantes que, conforme avanza la cursada, comienzan a responderse dudas entre ellos bajo la supervisión del plantel docente. Como particularidad del Servidor, entre los canales propuestos, se definió uno denominado «Patio» para alentar a que los estudiantes realicen intercambios *off-topic* en ese lugar, en el marco de la asignatura.

Para la ejercitación, actividad fundamental en la enseñanza de la programación, se decidió con-

²Discord es una plataforma de distribución digital y una aplicación de VoIP gratuita diseñada para comunidades de videojuegos, que se especializa en la comunicación de texto, imágenes, video y audio entre usuarios en un canal de chat [12].

tinuar con consignas de trabajos prácticos no entregables que buscan el desarrollo de un algoritmo para la resolución de un problema concreto. Asimismo se incorporaron notebooks Jupyter³ diseñadas por el equipo docente, alojadas en *Google Colaboratory*. Estas herramientas permiten escribir y ejecutar código de Python en el navegador prescindiendo de configuración, con acceso gratuito a GPU y facilidad para compartir código⁴. Las notebooks de Colab permiten combinar código ejecutable y texto enriquecido en un único documento, junto con imágenes, HTML y LaTeX entre otros. A su vez, es posible almacenar estas notebooks en una cuenta de Google Drive o editarlas de forma colaborativa. En el marco de la asignatura, se diseñó una *notebook* por cada unidad, a efectos que los estudiantes puedan ver implementados en código los conceptos de cada unidad, ejecutar el código y modificarlo de forma interactiva.

El nuevo esquema propuesto para la asignatura se completa con la implementación de mecanismos de evaluación formativa de dos tipos: autoevaluaciones y *homeworks*.

Las autoevaluaciones son cuestionarios de opción múltiple que son accedidas desde el Aula Virtual y tienen como objetivo pedagógico repasar los principales conceptos teórico-prácticos de cada unidad. Tienen la particularidad que se corrigen automáticamente una vez entregadas y asignan un puntaje a los estudiantes.

Por su parte, los *homeworks* consisten en un conjunto de ejercicios de programación que deben ser resueltos por los estudiantes. Para cada ejercicio de programación, el equipo docente define además del enunciado del ejercicio, un conjunto de casos de tests. El estudiante tiene la posibilidad de correr los tests en forma local en su computadora, para probar si hizo bien el ejercicio, y también puede correr los tests en forma remota en la plataforma *GitHub*. Para la organización e implementación de los *homeworks* se utiliza *GitHub Classroom*⁵. Esta aplicación es una plataforma educativa desarrollada por *GitHub*, que posee una interfaz de usuario para que los docentes publiquen tareas de programación para un curso en un aula virtual y

luego es posible recopilar las resoluciones de los repositorios de los estudiantes. El proceso comienza cuando el docente crea una página del curso en el *GitHub Classroom* y crea asignaciones o tareas. Para cada tarea, el docente envía un enlace de invitación a los estudiantes y, una vez que el estudiante acepta la invitación de la tarea, el sistema crea automáticamente un repositorio para almacenarla. De forma predeterminada, estos repositorios son públicos, lo que significa que los estudiantes pueden ver los repositorios de los demás. Sin embargo, los docentes pueden solicitar el uso de repositorios privados enviando sus credenciales educativas a GitHub [1]. Cuando el estudiante entrega las actividades mediante esta plataforma, la misma corre los *test* de prueba asignando un puntaje y registrando automáticamente esta situación así como la entrega. De esta forma, el estudiante puede saber en forma instantánea el puntaje que sacó en el ejercicio. Además, se da oportunidad a los estudiantes de que entreguen tantas veces como necesiten cada *homework* hasta que este sea aprobado, con restricción únicamente del plazo de entrega propuesto para esa actividad. Esta herramienta permitió al equipo docente la posibilidad de contar con evaluaciones formativas relacionadas con las destrezas adquiridas por los estudiantes para programar sin la necesidad de corregir las entregas manualmente, lo cual sería prácticamente imposible de implementar si se considera el alto cociente entre estudiantes y docentes. Al mismo tiempo, permite el acercamiento temprano de los estudiantes a los sistemas de versionado de código que luego utilizarán en su etapa profesional, para lo cual el equipo docente puso a disposición material audiovisual en el cual se abordan las nociones básicas de estas herramientas.

Por último, para la evaluación del conocimiento y posterior calificación de los estudiantes se definió un único examen parcial de carácter teórico-práctico para quienes hayan cumplimentado todas las evaluaciones formativas. Este parcial cuenta con la posibilidad de ser recuperado en una instancia y a su vez, previo al examen se realiza, a mitad de cursada, un simulacro de examen al sólo efecto que los estudiantes conozcan la forma de evaluación y les resulte menos estresante esta actividad.

³Sitio oficial: jupyter.org

⁴Sitio oficial: colab.research.google.com

⁵Sitio oficial: classroom.github.com

Resultados de la implementación

La implementación de la metodología propuesta se realizó de forma paulatina a partir del año 2018. Este cambio persigue resultados *cualitativos*, en términos de la forma en que los estudiantes se apropian de las competencias para programar, *motivacionales*, en el sentido que busca un aprendizaje activo, y *cuantitativos*, dado que se busca que más estudiantes ingresen a la Carrera, aprendan y avancen en la propuesta formativa. En este sentido, es interesante analizar los resultados cuantitativos en términos de la tasa de aprobación registrada en la asignatura en los últimos años así como la evolución de la cantidad de inscriptos. Para sus estadísticas oficiales [22], la Universidad Nacional de Luján define en el manual técnico de indicadores la tasa de aprobación (*TA*), que se calcula de la siguiente manera:

$$TA = \frac{\text{Regulares} + \text{Promovidos}}{\text{Inscriptos} - \text{Ausentes}} \quad (1)$$

Se puede observar que en el cálculo de la *TA* se detraen los estudiantes ausentes debido a que los mismos no participan de la cursada al no haber desarrollado ninguna de las evaluaciones de la asignatura. Esta cuestión resulta apropiada particularmente para el contexto actual dado que reduce las distorsiones producidas por la situación de pandemia atravesada en los últimos años.

Cuadro 1: Evolución de la Tasa de aprobación en Introducción a la Programación 2018-2021 [22]

Cohorte	Insc.	TA	Aprob.
2018	228	53.55 %	83
2019	288	44.60 %	95
2020	372	80.71 %	113
2021	540	85.77 %	223
2022	673	-	-

En el Cuadro 1 se observa que la tasa de aprobación -salvo en el año 2019 que se produjo una disminución de la *TA*- creció de forma sostenida, pasando del 53.85 % en 2018 al 85.77 % en el año 2021. A su vez, la cantidad de inscriptos a la Carrera se triplicó entre 2018 y 2022, razón por la cual resulta vital la automatización de la corrección de las evaluaciones formativas a través

de la implementación de las autoevaluaciones y *homeworks*.

Conclusiones

En este trabajo se introduce una propuesta de organización para un curso de grado cuyo objetivo es la enseñanza de la programación a nivel universitario. En esta propuesta, se parte de la base de una asignatura organizada bajo los preceptos de un modelo transmisivo tradicional, el cual constituye el esquema adoptado mayoritariamente en el Sistema Universitario.

Se presentan las tecnologías utilizadas y los cambios implementados, para transformar la enseñanza de la asignatura al paradigma de *flipped learning* con el objetivo principal de motivar a los estudiantes y ponerlos en el centro del proceso de enseñanza.

Los cambios introducidos han permitido enseñar los contenidos de la asignatura en forma totalmente remota en tiempos de pandemia, elevar la motivación y participación de los alumnos, así como facilitar la enseñanza a alumnos que están a grandes distancias físicas de la Universidad, generando un ecosistema de sociabilización entre los alumnos e introduciendo la enseñanza horizontal entre pares.

A su vez, la implementación de prácticas masivas de programación con corrección automática permitió contar con evaluaciones formativas en una asignatura en la cual la cantidad de estudiantes supera altamente las capacidades físicas y de recursos humanos de la Universidad.

Entre las potenciales desventajas del enfoque propuesto es posible señalar que es necesario que los estudiantes cuenten con conectividad para poder acceder al ecosistema de aprendizaje definido. También es cierto que se requiere algún tipo de dispositivo para la programación, pero este último es un requerimiento también vigente en el esquema tradicional de enseñanza.

Otra cuestión a señalar son los riesgos que se corren cuando los estudiantes, acostumbrados a un enfoque transmisivo, no se posicionan en el centro del proceso de enseñanza tomando un rol activo. Esta cuestión puede llevar a que acudan, o se conecten, a los encuentros de intercambio y consultas sin haber consumido los contenidos semanales, lo cual lleva a que los estudiantes no

puedan interactuar y participar, desaprovechando estos encuentros, generando el atraso en la cursada y poniendo en riesgo la aprobación de la asignatura. En este sentido, el equipo docente intenta mitigar los riesgos fomentando la comunicación constante entre los docentes y estudiantes, así como también entre los propios estudiantes, a partir de los foros en Discord y el envío de noticias.

Por último, el *stack* tecnológico es más amplio que en los cursos tradicionales, lo cual requiere que estudiantes de primer año deban aprender una serie de herramientas adicionales como las *notebooks* Jupyter, *Discord* y *GitHub*. En este sentido, en el marco de la asignatura se han diseñado espacios especiales, como el canal de voz de Discord «Servicio Técnico», para la puesta a punto de las instalaciones necesarias y el uso de *GitHub*, donde los estudiantes suelen demostrar mayores dificultades.

Referencias

- [1] ANGULO, M. A., AND AKTUNC, O. Using github as a teaching tool for programming courses. In *2018 Gulf Southwest Section Conference* (2019).
- [2] ATER-KRANOV, A., BRYANT, R., ORR, G., WALLACE, S., AND ZHANG, M. Developing a community definition and teaching modules for computational thinking: accomplishments and challenges. In *Proceedings of the 2010 ACM conference on Information technology education* (2010), pp. 143–148.
- [3] ATKINS, M., AND BROWN, G. *Effective teaching in higher education*. Routledge, 2002.
- [4] BERENGUER-ALBALADEJO, C., ET AL. Acerca de la utilidad del aula invertida o flipped classroom. *XIV Jornadas de Redes de Investigación en Docencia Univeritaria* (2016).
- [5] BERGMANN, J., AND SAMS, A. *Flip your classroom: Reach every student in every class every day*. International society for technology in education, 2012.
- [6] BERGMANN, J., AND SAMS, A. *Flipped learning: Gateway to student engagement*. International Society for Technology in Education, 2014.
- [7] DAPOZO, G., GREINER, C., PETRIS, R., GODOY, M. V., AND GUGLIELMONE, M. C. E. Enseñanza de programación en la universidad. estrategia basada en programación por bloques. *Innovation & Practice in Education*, 89.
- [8] DE GIUSTI, A. E., MADOZ, M. C., AND BERTONE, R. A. *Algoritmos, datos y programas con aplicaciones en Pascal, Delphi y Visual Da Vinci*. Prentice Hall, 2001.
- [9] FELDER, R. M., AND BRENT, R. Active learning: An introduction. *ASQ higher education brief* 2, 4 (2009), 1–5.
- [10] FERNÁNDEZ, J. M., OLORIZ, M. G., AND PUGGIONI, N. A. Efectos de la modalidad remota en el desempeño académico según campo disciplinar. In *Congresos CLABES* (2021).
- [11] FERREIRA SZPINIAK, A., AND ROJO, G. A. Enseñanza de la programación. *TE & ET* (2006).
- [12] FONSECA CACHO, J. Using discord to improve student communication, engagement, and performance.
- [13] HEININGER, R., SEIFERT, V., PRIFTI, L., UTESCH, M., AND KRCMAR, H. The playful learning approach for learning how to program: a structured lesson plan. *BLED* (2017).
- [14] LINDO, A. P. Prospectiva de la universidad argentina 2030. *Revista Argentina de Educación Superior*, 21 (2020), 160–175.
- [15] MARCH, A. F. Metodologías activas para la formación de competencias. *Educatio siglo XXI* 24 (2006), 35–56.
- [16] MAYORGA FERNÁNDEZ, M. J., AND MADRID VIVAR, D. Modelos didácticos y estrategias de enseñanza en el espacio europeo de educación superior.

- [17] MIGUEL DÍAZ, M. D. Cambio de paradigma metodológico en la educación superior: exigencias que conlleva.
- [18] PRENSKY, M. Digital natives, digital immigrants part 2: Do they really think differently? *On the horizon* (2001).
- [19] ROEHL, A., REDDY, S. L., AND SHANNON, G. J. The flipped classroom: An opportunity to engage millennial students through active learning strategies. *Journal of Family & Consumer Sciences* 105, 2 (2013), 44–49.
- [20] TOURÓN, J., AND SANTIAGO, R. col.(2013).“the flipped classroom” española: experiencias y recursos para dar ‘la vuelta’ a la clase.
- [21] TOURÓN, J., AND SANTIAGO, R. *El modelo Flipped Learning y el desarrollo del talento en la escuela: Flipped Learning model and the development of talent at school*, vol. 368. Ministerio de Educación, 2015.
- [22] Universidad Nacional de Luján, Departamento de Estadísticas Educativas. <http://www.estadisticaseducativas.unlu.edu.ar>.
- [23] WACHENCHAUZER, R., MANTEROLA, M., CURIA, M., MEDRANO, M., AND PAEZ, N. Algoritmos y programación i-aprendiendo a programar usando python como herramienta. *Recuperado el 17* (2015).
- [24] WIRTH, N. The programming language pascal. *Acta informatica* 1, 1 (1971), 35–63.