

Introducción a la didáctica de la programación para un profesorado universitario en Informática

Virginia Brassesco¹²

¹ Departamento de Computación. FCEyN. Universidad de Buenos Aires,
CABA. Argentina

OrcID 0009-0005-7284-7563

virginia.brassesco@unipe.edu.ar

² Universidad Pedagógica Nacional
CABA. Argentina

Resumen La didáctica de las Ciencias de la Computación es una disciplina científica reciente que se encuentra en desarrollo. Si bien existen diferentes propuestas de plataformas y enfoques de enseñanza de la computación en las escuelas, el desafío está en formar futuros profesores y profesoras de informática para su dictado, en los colegios secundarios de la Argentina. En este artículo se describe la primera versión y experiencia de dictado de la materia "Didáctica de la Programación 1", en el marco de la nueva carrera: Profesorado en Informática, de la Universidad Pedagógica Nacional (UniPe). Esta materia utiliza material generado por las y los estudiantes en materias correlativas, para reflexionar sobre sus propios procesos de aprendizaje y las formas de enseñanza de distintos conceptos computacionales. Se busca que las y los estudiantes no sólo sean capaces de reproducir fichas de actividades existentes, sino que puedan crear nuevas y fundamentar desde distintas perspectivas teóricas sus producciones. En este artículo se presentarán los objetivos de la materia, su fundamentación dentro del profesorado y las estrategias de enseñanza y evaluación.

Abstract The didactics of Computer Science is a recent scientific discipline that is in development. Although there are different proposals for platforms and approaches to teaching computing in schools, the challenge is to train future teachers of computer science for its dictation in secondary schools in Argentina. This article describes the first version and dictation experience of the subject "Didactics of Programming 1", within the framework of the new career: Computer Science Teachers, from Pedagogic National University (UniPe). This subject uses material generated by the students in correlative subjects, to reflect on their own learning processes and the ways of teaching different computational concepts. It is sought that the students are not only capable of reproducing existing activity sheets, but also that they can create new ones and base their productions from different theoretical perspectives. This article will present the objectives of the subject, its foundation within the teaching staff and the teaching and evaluation strategies.

Palabras clave: Didáctica, Programación, Profesorado.

Keywords: Didactics, Programming, University Education

1 Introducción

La didáctica de las Ciencias de la Computación es una disciplina científica reciente que se encuentra en desarrollo. Si bien existen diferentes propuestas de plataformas y enfoques de enseñanza de la computación en las escuelas, el desafío está en formar futuros profesores y profesoras de informática para su dictado, en los colegios secundarios de la Argentina.

En este artículo se describe la primera versión y experiencia de dictado de la materia "Didáctica de la Programación 1", en el marco de la nueva carrera: Profesorado en Informática, de la UniPe. Esta materia utiliza material generado por las y los estudiantes en materias correlativas, para reflexionar sobre sus propios procesos de aprendizaje y las formas de enseñanza de distintos conceptos computacionales.

Se busca que las y los estudiantes no sólo sean capaces de reproducir fichas de actividades existentes, sino que puedan crear nuevas y fundamentar desde distintas perspectivas teóricas sus producciones. En este artículo se presentarán los objetivos de la materia, su fundamentación dentro del profesorado y las estrategias de enseñanza y evaluación.

El artículo está organizado de la siguiente manera: se comienza con una fundamentación de la materia dentro del profesorado, se continúa con los contenidos de aprendizaje y la secuenciación propuesta. Luego presentamos las diferentes propuestas de actividades de clases y la evaluación de la materia con una propuesta formativa. Por último, concluimos con una discusión sobre las primeras dos ediciones de la materia y próximos pasos.

2 Fundamentación y encuadre de la materia en el Profesorado

Una premisa para UniPe es promover una cierta relación con el conocimiento que comprenda una actitud investigativa en los estudiantes para generar una reflexión conceptual sobre los contenidos específicos que irán abordando, a la vez que preguntarse acerca de la validez del conocimiento que van desarrollando en su interacción con las prácticas.

Estamos interesados en formar profesionales comprometidos con su tiempo, que puedan intervenir en los problemas de la práctica sin caer en intervenciones desde una racionalidad técnica sino a partir de un pensamiento que considere las particularidades de cada situación y pueda recrear respuestas adecuadas para ellas. Buscamos que en las aulas de UniPe se evidencie la intención de ayudar y animar a nuestros estudiantes a aprender. Promovemos una formación a través de la cual se pueda influir positiva, sustancial y sostenidamente en

sus formas de pensar, actuar y sentir, partiendo de la consideración del alumno ciudadano en permanente formación.

Esta materia es la primera del área de Didáctica de las Ciencias de la Computación que deben cursar las y los estudiantes, una vez adquiridos ciertos conocimientos en programación. Por este motivo, se presenta como una asignatura introductoria en términos de didáctica escolar específica, que se desarrolla en torno a los contenidos disciplinares presentados en la materia correlativa anterior: Programación 1.

En estos últimos años, la programación ha sido una de las disciplinas que más desarrollo presenta en materia de didáctica de la computación. A través de diferentes iniciativas, organizaciones de todo el mundo (incluidas de nuestro país) vienen desarrollando a lo largo de estos años un amplio abanico de recursos afines, entre los que se encuentran: plataformas online, actividades con y sin computadoras, investigaciones, desarrollo de cursos y dictado de capacitaciones, manuales docentes, planificaciones anuales, etc.

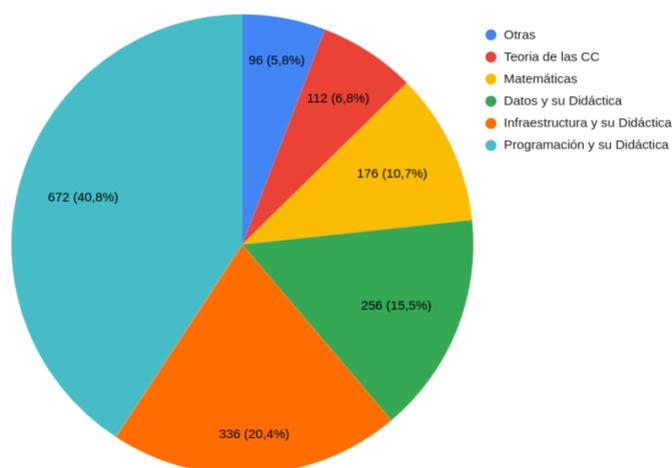


Fig. 1. Distribución de horas del Campo de Formación Disciplinar Específico, corresponden al 57% -1648 horas- de las horas totales -2856 horas- del profesorado. Dentro de éstas más del 40% son materias de Programación y su didáctica.

El profesorado en la UniPe tiene una duración de 4 (cuatro) años y una carga horaria total de 2856 horas. De las cuales más del 57% (1648 horas) son parte del total del Campo de la Formación Disciplinar Específica; y dentro de estas horas más del 40% son materias de Programación y su didáctica, ver Figura 1. Podemos ver que este profesorado le da importancia a la programación, y la pone de relevancia dentro de las materias específicas de su formación. Esto se enmarca dentro de la teoría de utilizar la programación como

herramienta para enseñar conceptos computacionales más complejos como por ejemplo la ciencia de datos, la inteligencia artificial o el diseño de sistemas de software complejos. Las y los egresados de la formación, tendrán así una sólida noción de un abanico de lenguajes computacionales y dominio de distintos entornos de programación. Además su formación en conceptos de programación, y no en un lenguaje específico, les permitirá mantenerse actualizados respecto de los avances y cambios de la disciplina.

Didáctica de la Programación 1, reúne una serie de conceptos, teorías y debates fundamentales que permiten conocer, entre otras cosas, las bases y discusiones que dieron inicio a la necesidad de incorporar este campo en el sistema educativo nacional, cuáles son y cómo se desarrollan las propuestas y debates actuales, el estudio, desarrollo e incorporación de este campo en otros sistemas educativos del mundo y cómo se integran las didácticas de las ciencias exactas y naturales en general. Se discuten algunos conceptos sobre la enseñanza e incorporación de las Ciencias de la Computación en el aula, los distintos debates que se desarrollan a partir del concepto de Pensamiento Computacional y criterios de análisis de distintas plataformas y entornos educativos de programación.

Por otro lado, la materia cuenta con un espacio dedicado al estudio y ejercitación sobre la enseñanza de la programación en el aula. Se trabaja sobre un subconjunto de contenidos presentados en la materia correlativa: *Programación 1*. La repetición condicional, la alternativa condicional o el énfasis en la creación y el uso de nuevos procedimientos son algunos ejemplos de temas que serían pertinentes abordar desde un enfoque didáctico para poder ser trabajados en el aula.

De forma conjunta, se desarrollan y analizan diferentes estrategias didácticas aplicables a la enseñanza de los contenidos a partir de las experiencias en otras áreas de las ciencias Exactas y Naturales y de los debates y experiencias desarrolladas en los últimos años sobre Didáctica de la Programación.

A través de la familiarización con diferentes recursos disponibles, se propone que las y los estudiantes puedan evaluar, criticar e integrar en sus prácticas docentes el uso de estos recursos con las estrategias de enseñanza presentadas.

3 Objetivos de aprendizaje

Podemos delinear los objetivos de aprendizaje de la materia en tres ejes principales: por un lado el análisis crítico de secuencias didácticas existentes, por otro la elaboración de actividades que desarrollen algún contenido específico y por último la posibilidad de evaluación y análisis sistemático de propuestas y recursos existentes.

Dentro del primer objetivo se busca que las y los estudiantes sean capaces de analizar y llevar a la práctica secuencias didácticas existentes y proponer recursos significativos que

se puedan acoplar a las mismas (fichas de actividades con y sin computadoras, software que permita exponer algún concepto, etc.). Para esto, durante la cursada, se dan a conocer y se van evaluando algunas propuestas didácticas desarrolladas en otras Ciencias Exactas y Naturales; se estudia y se debate sobre recursos existentes; y se reflexiona sobre las estrategias didácticas involucradas en el desarrollo de distintas secuencias didácticas.

En segundo lugar, buscamos que las y los estudiantes elaboren fragmentos de clases que desarrollen algún contenido específico y que, posteriormente, se puedan integrar en conjunto y llevar a cabo en el aula. Para esto seleccionamos contenidos significativos que se desarrollarán en una clase, utilizando lo aprendido en la materia anterior correlativa a esta, Programación 1. Dicha elaboración es llevada a la práctica en forma de microclases (Anijovich et al., 2009, 119-134) frente a sus compañeros y poder así conocer y desarrollar criterios para la exposición de contenidos.

Por último, con el fin de poder evaluar, integrar y criticar propuestas y recursos existentes sobre didáctica y enseñanza de la programación se indaga sobre iniciativas existentes de enseñanza de las Ciencias de la Computación en la escuela en distintas provincias de la Argentina y países de latinoamérica.

4 Desarrollo de los contenidos en el cuatrimestre

La materia tiene una carga de 6 horas semanales, dictada en dos días distintos. Esto permite distribuir el trabajo semanal con las y los estudiantes en el aula física y en el aula virtual. Se encuentra organizada principalmente en dos grandes bloques: diseño de actividades y conceptualización de nociones de didáctica de la programación.

Las clases dedicadas a diseño de actividades están a su vez divididas para permitir el trabajo de 3 conceptos computacionales distintos a lo largo del cuatrimestre. Además esto permite que, con el espaciado de casi un mes entre cada actividad, se puedan ir desarrollando más contenidos teóricos que se puedan aplicar en el siguiente diseño.

Respecto a la revisión bibliográfica, la materia se divide en 4 grandes tópicos:

- Materia en la escolaridad obligatoria:
 - Debate sobre la urgencia e importancia dentro de la escolaridad obligatoria.
 - Propuestas de incorporación a la currícula escolar (transversal, disciplinar, interdisciplinario, multidisciplinario, etc.).
 - Análisis de distintas iniciativas a nivel nacional y latinoamericano.
- Concepto de Pensamiento Computacional:
 - Definición

- Debates emergentes.
- Didáctica de la Programación:
 - Teorías y propuestas desarrolladas en los últimos años.
 - Familiarización con recursos didácticos disponibles: plataformas online, fichas de actividades, manuales a fines, actividades sin computadoras, etc.
 - Marco de referencia. Estrategias didácticas: Aprendizaje Basado en Problemas.
- Didáctica de las Ciencias Exactas y Naturales:
 - Uso Bases de Orientación.
 - Formas de evaluación formativa para el aprendizaje por comprensión.

La secuenciación de los contenidos fue organizada de forma tal que a lo largo de la cursada junto con el diseño de las actividades, se viesen reflejados los distintos conceptos históricos y teóricos de forma incremental.

Por ejemplo, uno de los primeros temas que se tratan en la materia es "qué es la programación" (Martinez Lopez, 2013, 23-35). Se debate en torno a las ideas de la construcción de programas y la posibilidad de usar distintos lenguajes de programación para poder contar una misma idea. De esta forma las y los estudiantes comienzan a tener una visión más criteriosa de la elección de los lenguajes de programación elegidos, y más adelante podemos discutir sobre la paradoja de la elección del lenguaje y el entorno para enseñar a programar: "el lenguaje y el entorno a utilizar no son importantes, pero sin embargo son importantes" (Martinez Lopez y Lopez, 2021, 102). Se introduce entonces la propuesta de análisis, mediante diversas dimensiones y facetas, de los entornos elegidos en las fichas subsiguientes. Las y los estudiantes, utilizan dichos criterios para defender sus propuestas de plataformas y comenzar a tener debates teóricos sobre la propuesta didáctica.

Otro de los temas que permite un debate rico a lo largo de la cursada es la noción de "Pensamiento Computacional". Las clases se pensaron primero con una construcción de nuestras propias definiciones analizando diferentes referentes y organizaciones del campo de las ciencias de la computación, para luego poder construir una crítica constructiva junto con un análisis histórico de la necesidad del constructo. Empezamos entonces analizando el artículo de Wing (2006), sentando las bases para iniciar el debate sobre el constructo, y siguiendo con la visión y diferencias entre la Asociación de Maestros en Ciencias de la Computación¹ de EEUU (2011) y la Fundación Sadosky (2013). Finalmente se propone un debate sobre el uso del término en la construcción del currículum en Ciencias de la Computación en las escuelas de Argentina. Utilizando como disparadores el artículo de Bonello y Schapachnik (2020) y material audiovisual de las Jornadas Argentinas de Didáctica de las Ciencias de la Computación 2021 (por ej: Experiencias de Ciencias de la

¹ <https://csteachers.org/page/standards-for-cs-teachers>

Computación en provincias argentinas²), se pone en cuestionamiento el constructo para recuperar miradas sobre temas emergentes en relación a la inclusión de la materia en el contexto escolar.

Por último, mencionar que desde el primer día se analizan diferentes miradas sobre la necesidad de enseñar Ciencias de la Computación en la escolaridad obligatoria. Se usan diferentes recursos audiovisuales y textos para poder enriquecer las visiones de esta necesidad social y política. A través de la lectura y escucha de entrevistas y ponencias de artículos, se van contraponiendo los distintos puntos de vista, para que las y los estudiantes puedan darle sentido a su importancia como futuros agentes de cambio en las instituciones educativas. Se analizan también las distintas legislaciones Argentinas y los cambios a lo largo del tiempo, en sus considerandos. Se analizan así distintos aspectos, por ejemplo el económico, viendo la inclusión de la Programación en la escolaridad obligatoria como fortaleza para "el desarrollo económico-social de la Nación" según el artículo 1 de la resolución 263 (2015) del Consejo Federal de Educación³, y a su vez citando evidencia en sus considerandos, de que las y los niños que aprenden a programar "mejoran su desempeño en otras áreas disciplinares, entre ellas matemática y lenguas extranjeras".

Desde esta concepción formativa, con visión histórica y análisis crítico de las distintas propuestas, las y los estudiantes pueden reconocer y asumir el sentido social y político de su tarea de enseñar, ubicada en un momento estratégico de cambio educativo en la Argentina -mediante la incorporación de los Núcleos de Aprendizaje Prioritarios (NAP) de Educación Digital, Programación y Robótica (2018)⁴, y que su futura labor, la de enseñar, constituye una intervención intencional y sistemática de valor pedagógico y social.

Cabe aclarar que los NAP, en la escuela, se refieren a un conjunto de saberes centrales, relevantes y significativos, que incorporados como objetos de enseñanza, contribuyen a desarrollar, construir y ampliar las posibilidades cognitivas expresivas y sociales de las y los estudiantes. Fueron creados para cumplir con la ley 26.206 (2006) de la República Argentina, y desarrollar las competencias necesarias para el manejo de los nuevos lenguajes producidos por las tecnologías de la información y la comunicación.

5 Actividades en clase

La propuesta central que integra y articula todos los contenidos es la creación de un *podcast*. A lo largo de la cursada van recolectando material y grabando distintas cápsulas

² <https://www.youtube.com/watch?v=fZPaBTdQQfw>

³ <https://cfe.educacion.gob.ar/resoluciones/res15/263-15.pdf>

⁴ <https://www.argentina.gob.ar/nivelesymodalidades/direccion-nacional-de-educacion-primaria-0/nucleos-de-aprendizajes-3>

que pueden ser usadas en la producción final. Los guiones y grabaciones son compartidas en el campus con el resto del curso para que puedan ser utilizadas libremente.

Se ejercita la escritura de la escaleta para el diagramado de cada cápsula, y llegar con dominio de la misma para el armado del *podcast* final. Además, entre ellos elaboran entrevistas para compartir las visiones de diferentes referentes. Comienzan mediante el uso de preguntas cuyas respuestas son deducibles del material usado en clase, y de a poco proponen respuestas con miradas críticas a las distintas temáticas mencionadas en el apartado anterior.

El trabajo en equipo es parte fundamental de la materia, el aprendizaje en conjunto y el compartir el conocimiento termina siendo un eje enriquecedor que nutre el uso de distintas herramientas informáticas para digitalizar las producciones. Por ejemplo, una integrante del curso puede conocer cómo utilizar *Audacity* y enseñar en clase al resto cómo se puede compaginar y usar sonidos para enriquecer la propuesta audítil.

Con la creación de los guiones ejercitan la escritura creativa y narrativa, deben contar una historia donde introduzcan los contenidos y logren capturar a los oyentes. Además de cada tema se les sugieren ciertas preguntas para que ellos puedan profundizar en los contenidos más allá de la bibliografía dada y poder también desarrollar su pensamiento crítico respecto a los distintos tópicos de la materia.

Para poder construir el *podcast* articulan los conocimientos usando la línea de tiempo que se construye clase a clase. En esta oportunidad, a sugerencia de las docentes, se utilizó la herramienta Padlet que permite de forma interactiva ir agregando tarjetas en la línea temporal, junto con imágenes ilustrativas y textos. En la línea temporal se van anotando las apariciones de las distintas plataformas educativas, desde la creación de *Logo* hasta por ejemplo la creación de *ChatGPT* y *Copilot* como asistente para programar.

La línea temporal les ayuda a visualizar cómo se conecta el desarrollo de las computadoras, con las propuestas didácticas para poder programarlas con formas cada vez más didácticas e interactivas. A su vez se realiza un análisis de la aparición de las computadoras vinculado con los hechos geopolíticos del momento, por ejemplo la segunda guerra mundial.

Por último, cada grupo debe proponer actividades para enseñar a programar tres conceptos de la programación imperativa: la división en subtareas y uso de procedimientos, la alternativa condicional y la repetición condicional. La repetición fija fue omitida de las opciones, pero los estudiantes la pueden usar como opción para proponer una secuencia en el caso de la división en subtareas o la alternativa condicional.

Existe un desafío en imaginar a sus posibles estudiantes, cuáles son sus conocimientos previos, qué contenido puedo asumir que saben para el andamiaje de los nuevos conceptos.

Para poder elaborar las actividades se trabaja primero utilizando sus bitácoras de la materia correlativa "Programación 1", donde ellos registraron cómo aprendieron, qué dificultades tuvieron a la hora de solucionar los desafíos, y cuáles podían ser distintas soluciones para el mismo problema. Además analizan cuáles fueron los desafíos introductorios de cada temática y qué variedad de recursos se utilizaron. Por ejemplo, si usaron una actividad desenchufada para empezar a indagar sobre "sensores booleanos" (decisiones en un mapa con distintos semáforos que permiten avanzar, girar a izquierda o derecha) y otra en computadora donde deben prender o apagar una lámpara.

Luego deben pensar el público y un problema interesante a resolver para ese estudiantado imaginario que ellos eligen. Por ejemplo estudiantes de primer año del colegio secundario interesados en el medio ambiente, un problema interesante puede ser el reciclaje y la reutilización.

Basándose entonces sobre las propuestas que ellos conocen construyen nuevos escenarios. La creatividad se despierta cuando analizan diferentes plataformas educativas para resolver los problemas. La reutilización de soluciones disponibles es un camino posible para que la curva de aprendizaje de la nueva tecnología no sea muy alta, y puedan a lo largo de tres semanas centrarse en mejorar y acotar la propuesta.

Durante la construcción de sus actividades van anotando la propuesta en unas fichas compartidas con el cuerpo docente que puede ir haciendo sugerencias de mejoras e inquietudes.

Finalmente se produce el momento esperado por todas y todos, la práctica del dictado de dicha actividad a sus compañeros y compañeras. Si bien la duración de este momento interactivo suele ser de unos 40 minutos, hemos adoptado la idea de microclase sugerida por Anijovich (2009, 45): "Los testimonios que los estudiantes producen sobre los momentos previos y posteriores a sus microclases permiten entrenar la capacidad de reflexionar sobre las decisiones tomadas antes, durante y después de la acción".

Durante el dictado de la microclase, si la cantidad de estudiantes es suficiente se asignan roles de observadores a algunos de ellos y otros serán participantes activos de la clase simulando ser estudiantes para poner a prueba la actividad. Las personas que observan la clase utilizan algún instrumento protocolo de observación previamente seleccionado por todos, por ejemplo, la tabla "Fortalezas, debilidades, evidencias, sugerencias" (Anijovich et al., 2009, 48). Cada clase se rotan las personas para que todas puedan pasar por las distintas instancias.

El momento postactivo es altamente enriquecedor para todas las personas ya que es un intercambio entre la pluralidad de voces, donde cada persona aporta con sus conocimientos

previos y sus percepciones de la clase. Las devoluciones de sus compañeros y de sus formadores y su propia percepción sobre lo ocurrido, le permiten al estudiante que dió la clase reflexionar sobre su propia percepción de lo vivido.

6 Evaluación formativa y sumativa de los contenidos

La acreditación de las materias universitarias son parte de los pasos requeridos para la obtención del título, y como tal son un hito clave en la formación de nuestras y nuestros estudiantes. Pero la acreditación, que en la mayoría de los casos requiere de una puntuación numérica, puede no ser reflejo del trabajo y aprendizaje de las y los estudiantes a lo largo de la cursada.

En el profesorado de informática de la UniPe se inscriben tanto personas que son primera generación universitaria, como también profesionales docentes con otras formaciones de base. Personas que recién terminan el secundario y otras que ya son adultos. No será entonces el mismo desarrollo que tendrá una persona que ya ejerce la docencia, y puede contar con una formación pedagógica previa, que otra que conoce cómo se ejerce la docencia por haber sido estudiante de instituciones educativas. En ambos casos buscamos relevar que el aprendizaje de los contenidos haya sido el apropiado para poder aprobar la materia pero además que ambos puedan desarrollar al máximo sus potenciales durante la cursada y beneficiarse de los conocimientos y preguntas de sus compañeros y compañeras.

Dentro de esta heterogeneidad, incluso hay personas que atravesaron otras formaciones del área de informática y tienen sus ideas de cómo se pueden enseñar los distintos contenidos. Es sumamente importante la labor pedagógica, del docente a cargo del curso, para poder dar lugar a las distintas voces y puedan permitirse indagar sobre distintas opciones de dictado.

Para la regularización de la materia se pide a los alumnos cumplan con las entregas -que son todas grupales, en tiempo y forma- y realicen aportes en las discusiones de las microclases presentadas. La asistencia es obligatoria sólo en los momentos de las microclases.

Para acreditar la materia deben presentar el guión y el *podcast* grabado en grupo, y cada integrante tiene un coloquio sobre dos temas elegidos por ellos. Previo al final se les dejan situaciones problemáticas con preguntas disparadoras para que puedan elegir los temas a conversar por ejemplo: "Si alguien en el colectivo te pregunta porqué hay que incorporar la materia Ciencias de la Computación en el jardín de infantes, y porqué es urgente hacerlo, ¿vos qué le responderías?" o "Si alguien del ministerio de educación de Formosa te consulta respecto a la forma de incorporar pensamiento computacional en su currícula escolar, ¿qué sugerencias le harías?". En la instancia de coloquio se espera que la persona pueda

responder sobre los temas elegidos y mantener una conversación acorde a los interlocutores del problema, hablando con soltura de los distintos conceptos aprendidos. Para esta instancia se utiliza una rúbrica de evaluación, como la sugerida en (Anijovich y Cappelletti, 2017, 105).

7 Discusión y próximos pasos

Esta materia cuenta con tan solo dos ciclos de dictado. Actualmente se está dictando por segunda vez en la sede de San Telmo y la de Pilar. El equipo de trabajo está conformado actualmente por tres docentes, dos de ellas se incorporaron este último cuatrimestre y aportan a la mejora del contenido y su dictado. Las materias en UniPe, usualmente son dictadas por un único docente. En este caso hemos distribuido las clases de la sede Pilar entre las dos nuevas docentes.

Encontramos en la materia una introducción a los temas de didáctica de la computación, con una mirada amplia tanto sobre aspectos teóricos de la disciplina como de aplicaciones prácticas para llevar a las aulas.

Vemos que las y los estudiantes se comprometen con la preparación de las actividades, indagando sobre al menos tres opciones de plataformas para la enseñanza de programación, con guía por parte del cuerpo docente. El estudiantado muestra un alto compromiso de cursada con la materia, siendo la entrega de los trabajos en tiempo y forma un indicador positivo de seguimiento del compromiso y de la retención.

La calidad de las entregas se ve mejorada por los propios conocimientos del uso de otras herramientas informáticas, por ejemplo, algoritmos de IA para mejorar los audios grabados de forma casera y adaptarlos a formatos de *podcast*. Y dentro del aula se favorece el intercambio y aprendizaje grupal para que cada estudiante aporte con sus conocimientos al resto del equipo.

La materia tiene un espacio lúdico didáctico donde las y los estudiantes se reencuentran con los conceptos de Programación, e indagan sobre nuevas plataformas educativas para visitar dichos contenidos con nuevas miradas. Fuerza a repensar los ejercicios que ya saben resolver, y les funcionan, para plantear más opciones de soluciones posibles para el mismo problema.

La creación de fichas de actividades, pensando en futuros colegas docentes que usarán sus ideas en sus aulas, e incluso ellos mismos, fomenta la escritura crítica y conceptual y un compromiso con el tiempo y forma de entrega.

Estamos trabajando en la participación de nuestras y nuestros estudiantes en jornadas donde puedan presentar sus propuestas y obtener un aporte de la comunidad docente activa. Desde la universidad reconocemos el valor de la comunidad y el compartir el conocimiento para seguir creciendo y construyendo más y mejores prácticas. En particular, podemos mencionar, que el dictado de la materia en la sede Pilar, se vió enriquecido por la colaboración coordinada con la docente de la materia de "Práctica Profesional" que se dicta en simultáneo, quien realizó observaciones de las microclases de las y los estudiantes.

Respecto a los datos cuantitativos actuales, la población que cursó la materia es muy baja para poder hacer un análisis estadístico, sin embargo, podemos mencionar que la tasa de abandono de la materia es casi nula y los motivos son externos al contenido o la forma de dictado de la misma. Siendo nuestro mayor indicador cuantitativo la entrega a tiempo de todas las instancias de evaluación propuestas periódicamente.

8 Conclusiones

En este trabajo presentamos el curso de Didáctica de la programación 1, su fundamentación y las primeras experiencias de trabajo en clase. Se enuncian algunas actividades realizadas en las clases como por ejemplo, el armado de una línea de tiempo, que es usada a lo largo de toda la cursada para construir criterio histórico y político respecto a la construcción del conocimiento a partir de distintas plataformas y propuestas curriculares existentes. Entendemos la enseñanza como un proceso complejo que implica decisiones acerca del conocimiento a desplegar: para qué se enseña, qué se requiere enseñar, y cómo podría hacerse; por eso fomentamos el análisis crítico en la construcción de las actividades mediante el uso de bases de orientación.

Las primeras ediciones creemos, de acuerdo al compromiso y nivel de aprobación, que fueron exitosas. Sin embargo, para tener mayores conclusiones, requerimos seguir analizando sucesivos dictados y contar con más indicadores para analizar métricas de mejora del mismo.

A partir del corriente dictado incorporamos evaluaciones anónimas con preguntas respecto a la percepción y sensaciones que ha despertado el curso en el estudiantado. De esta forma podremos realizar un análisis cualitativo respecto al impacto de la materia, en relación a la búsqueda del aprendizaje por el placer del conocimiento. Sumando estos resultados a lo que hemos podido medir con los índices de entrega de todas las instancias de evaluación. Viendo mejoras incrementales en cada trabajo presentado, y alta creatividad en las propuestas, aportando valor agregado con sus propias experiencias previas.

Referencias

- Anijovich, R., Capelletti, G., Mora, S., y Sabelli, M. J. (2009). *Transitar la formación Pedagógica*. PAIDOS.
- Anijovich, R., y Cappelletti, G. (2017). *La evaluación como oportunidad*. Paidós.
- Bonello, B., y Schapachnik, F. (2020). Diez preguntas frecuentes (y urgentes) sobre pensamiento computacional. *Virtualidad, Educación y Ciencia*, 11(20), 156-167.
<https://program.ar/wp-content/uploads/2021/06/Diez-preguntas-frecuentes-y-urgentes-sobre-Pensamiento-Computacional.pdf>
- Dapozo, G. (2020). Estrategias de evaluación formativa en la enseñanza de programación en modalidad remota de emergencia. *XXVI Congreso Argentino de Ciencias de la Computación - CACIC 2020*, 1(1), 690-699. <http://sedici.unlp.edu.ar/handle/10915/114458>
- Fundación Sadosky. (2013). CC – 2016 [Una propuesta para refundar la enseñanza de la computación en las escuelas Argentinas]. *Informe Ejecutivo*. Fundación Sadosky.
<https://www.fundacionsadosky.org.ar/wp-content/uploads/2014/06/cc-2016.pdf>
- Martinez Lopez, P. F. (2013). *Las bases conceptuales de la Programación: Una nueva forma de aprender a programar*. Universidad de Quilmes.
- Martinez Lopez, P. F., y Lopez, M. J. (2021). ¿Scratch, Python, o qué? Criterios para elegir un entorno para enseñar a programar a principiantes. *Jornadas Argentinas de Didáctica de las Ciencias de la Computación*, 1(1), 101-114.
<https://jadicc.program.ar/wp-content/uploads/2022/03/actas-jadicc-2021.pdf>
- Wing, J. (2006). Computational Thinking. *Communications of the ACM*, 49(3), 33-35.
10.1145/1118178.1118215