

Escalando la enseñanza de programación en tiempos de pandemia: desafíos y oportunidades*

Manuela Cerdeiro, José Crespo,*** Oscar Filevich,******

Rafael Grimson*** y Matías López-Rosenfeld*******

Resumen

En el marco del proyecto de Inteligencia Artificial Interdisciplinaria de UNSAM se lanzó, en marzo de 2020, la materia Programación en Python. La pandemia de COVID-19 nos tomó por sorpresa y esa cursada se transformó en virtual. Para el segundo cuatrimestre ya contábamos con experiencia y capacidad de previsión lo que permitió cambiar radicalmente el enfoque del

* Enviado: 23-02-21. Aceptado: 26-05-21.

** Licenciada (2010) y Doctora (2015) en Ciencias Matemáticas (UBA). Actualmente se desempeña como Jefa de Trabajos Prácticos con dedicación exclusiva en la UBA y con dedicación simple en la UNSAM. Trabaja como matemática aplicada realizando transferencia e investigación en temas que incluyen el análisis de datos y la optimización (Departamento de Matemática, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires). Correo: cerdeiro@dm.uba.ar

*** Licenciado en Ciencias Biológicas (UBA, 2007) y Doctor en Ciencias Biológicas (UBA, 2011). Es investigador del CONICET con lugar de trabajo en el Instituto de Ecología, Genética y Evolución de Buenos Aires (UBA-CONICET) y Jefe de Trabajos Prácticos en el Departamento de Ecología, Genética y Evolución de la Facultad de Ciencias Exactas y Naturales (UBA). En la actualidad se encuentra trabajando en el estudio de termotolerancia de insectos con importancia económica o médica y en el desarrollo de dispositivos para la optimización de la toma de datos automáticas. Correo: crespo@ege.fcen.uba.ar

**** Licenciado en Ciencias Biológicas (UBA, 2006) y Doctor en Química Inorgánica, Química Analítica, y Química Física (UUBA, 2011). Es Investigador del CONICET con lugar de trabajo en el Instituto de Tecnologías Emergentes y Ciencias Aplicadas (UNSAM). Su tema de investigación es la neurociencia de circuitos, y la interacción glía-neurona. Es además profesor adjunto en la Escuela de Ciencia y Tecnología de la UNSAM. Correo: ofilevich@unsam.edu.ar

***** Licenciado en Ciencias Matemáticas (UBA, 2001), realizó una maestría en la École Polytechnique (Paris, Francia, 2005) y se doctoró en la Universidad de Hasselt en Bélgica en el 2010. Desde su regreso a Argentina trabaja en el desarrollo de modelos matemáticos y computacionales para el estudio de problemáticas ambientales. Es investigador del CONICET con lugar de trabajo en el Instituto de Investigaciones e Ingeniería Ambiental de la UNSAM y profesor adjunto en la Escuela de Ciencia y Tecnología de la UNSAM. Autor corresponsal de este artículo. Correo: rgrimson@unsam.edu.ar

***** Matías Lopez-Rosenfeld es Licenciado en Ciencias de la Computación (UBA, 2012), y Doctor en Ciencias de la Computación (UBA, 2017). Es investigador del CONICET con lugar de trabajo en el Instituto de Ciencias de la Computación (UBA-CONICET) y profesor adjunto en el Departamento de Computación de la Facultad de Ciencias Exactas y Naturales (UBA). En la actualidad se encuentra trabajando en el desarrollo de estrategias y herramientas computacionales para el aprendizaje de la programación. Correo: mlopez@dc.uba.ar

curso, y llevar adelante una completa virtualización de la metodología pedagógica. Este escrito relata la experiencia de docencia a distancia detallando herramientas y resultados observados en un curso con cientos de alumnos de grado y posgrado, de Argentina y del extranjero. Culmina discutiendo sobre necesidades pendientes y posibilidades a futuro. En particular, si bien la pandemia de COVID-19 ha destrozado muchas de nuestras mejores costumbres y prácticas, también ha acelerado una verdadera disrupción pedagógica generando nuevos espacios para la labor docente, que pueden aprovecharse para dictar cursos más horizontales, federales e inclusivos.

Palabras clave: Python. Programación. Enseñanza. Grado. Posgrado.

Abstract

In March 2020, within the interdisciplinary artificial intelligence project from UNSAM, “Python Programming” was launched. COVID-19 pandemics surprised us and we were forced to transform a previously course thought as face-to-face into completely virtual. For the following semester, we were already experienced and with sufficient prevision we were able to completely change the focus of the course and adequate it to the new conditions. We managed to accomplish this by adding teachers from different disciplines which led us to successfully provide a completely virtual pedagogic methodology. This text resumes our virtual teaching experience detailing our pedagogic approach and observed results in a course with hundreds of graduate and undergraduate students from Argentina and abroad. We end with a discussion on pending needs and future possibilities. Although the COVID-19 pandemics has trashed our best practices and customs, it has also hastened a true pedagogical disruption creating new spaces for teachers that should be seized for more horizontal, federal and inclusive courses.

Keywords: Python. Programming. Teaching. Graduate. Undergraduate.

Resumo

No âmbito do projeto de Inteligência Artificial Interdisciplinar da UNSAM, a disciplina Programação em Python foi lançada em março de 2020. A pandemia COVID-19 nos pegou de surpresa e aquele curso, que deveria ser ministrado pessoalmente, tornou-se virtual. Já no segundo semestre tínhamos experiência e capacidade de antevisão, o que nos permitiu mudar

radicalmente o enfoque do curso, adaptando-o às novas condições: somando professores de outras áreas, conseguimos fazer uma virtualização completa da metodologia pedagógica . Este texto relata a experiência do ensino a distância, detallando ferramentas e resultados observados em um curso com centenas de alunos de graduação e pós-graduação, da Argentina e do exterior. Termina discutiendo as necesidades pendientes e as posibilidades futuras. Em particular, enquanto a pandemia COVID-19 destruiu muitos de nossos melhores costumes e práticas, também acelerou una verdadera ruptura pedagógica, gerando novos espacios de ensino, que podem ser usados para ministrar cursos mais horizontais, federais e inclusivos.

Palavras-chave: Python. Programação. Ensino. Grau. Pós-graduação.

Introducción

El 11 de marzo de 2020 la Organización Mundial de la Salud (OMS) declaró el brote del nuevo coronavirus como una pandemia y apenas ocho días después el gobierno nacional decretó el Aislamiento Social Preventivo y Obligatorio. Esto impuso condicionantes a la docencia pero a su vez aceleró el surgimiento de nuevas herramientas para la interacción entre docentes y alumnos que parecían casi futuristas y tenían grandes resistencias (García Aretio, 2017). Pocos días antes, en el marco del proyecto de Inteligencia Artificial Interdisciplinaria, la Universidad Nacional de San Martín (UNSAM) había lanzado la materia de grado y posgrado “Programación en Python”, de carácter optativo tanto para las siete ingenierías como para el doctorado en ciencias aplicadas, carreras que se dictan en la Escuela de Ciencia y Tecnología de la UNSAM. La pandemia nos tomó por sorpresa y esa cursada, que se diseñó y comenzó en modo presencial, se transformó en virtual. Para el segundo cuatrimestre ya contábamos con algo de experiencia y capacidad de previsión por lo que decidimos cambiar radicalmente el enfoque de la materia y adecuarla a las nuevas condiciones. Diferentes docentes se sumaron a la experiencia, y en conjunto diagramamos una metodología de trabajo y la llevamos adelante.

En el mes de julio del 2020 anunciamos en diversos medios de difusión académica así como en redes sociales que la materia “Programación en Python” se dictaría de manera completamente virtual. La preinscripción excedió ampliamente nuestras expectativas llegando casi a los 1200 preinscriptos, incluyendo un 40% de personas que residen fuera del Área

Metropolitana de Buenos Aires (AMBA). Estas personas, que están fuera del alcance natural de nuestra universidad, eran principalmente del interior del país y algunas incluso residentes de otros países (cfr. Figura 1).



Figura 1. Geolocalización de inscriptos.

Originalmente, la materia había sido pensada para dictarse en un laboratorio de computación y las dimensiones del mismo imponían un cupo de 50 alumnos, por lo que este número de preinscriptos significó un cambio de escala. Se nos presentó entonces la disyuntiva entre excluir a muchos estudiantes, o modificar la metodología de enseñanza de manera radical para poder admitir a todos. La decisión fue no poner cupos. Como único requisito se les pidió a los estudiantes que hicieran la inscripción definitiva si poseían un mínimo conocimiento de programación en algún lenguaje (en particular, sobre los conceptos de variables, ciclos y condicionales) con el objetivo de tener un punto de partida común sobre el cual construir.

Dictar un curso virtual y con tantos inscriptos nos presentó un escenario al que hasta ahora no nos habíamos enfrentado nunca. A pesar de que la tecnología está disponible desde hace varios años, la educación virtual masiva ha enfrentado fuertes resistencias en el ámbito de la enseñanza universitaria Argentina y su crecimiento en estas instituciones ha sido lento y dispar (González, 2017; Marotias, 2018; Rivero, 2020).

El contexto de virtualidad forzada por la pandemia generó una verdadera disrupción pedagógica (García Aretio, 2017). Surgió la necesidad imperiosa de adaptarse a esta nueva modalidad repentinamente, tanto para los docentes como para los estudiantes. En particular, las posibilidades técnicas y contextos familiares o generales de los estudiantes afectaron profundamente su capacidad de adaptación a esta disrupción. El éxito de un estudiante en este nuevo contexto estuvo sin dudas modulado por sus costumbres anteriores y su nivel de apropiación de la tecnología necesaria para seguir el curso. En este sentido, es probable que esta modalidad haya dejado en el camino a aquellos estudiantes con ciertas dificultades técnicas (Pérez-López, 2021). Es adecuado mencionar que el dictado de un curso de programación en Python en línea no conforma una experiencia original en sí misma: existen numerosos cursos online masivos y abiertos (o en inglés MOOC, *Massive Online Open Courses*) de esta temática (por ejemplo, Coursera, EDX, MiríadaX, entre otros). Sin embargo, estos cursos presentan varias diferencias sustanciales con el que hemos dictado:

- Los cursos masivos y abiertos son mayoritariamente pagos. Si bien existen ayudas financieras y facilidades para estudiantes de países pobres, de todas formas requieren tener una tarjeta de crédito y asociarla al usuario con lo que se excluyen importantes segmentos de la población estudiantil. Además, aquellos que pueden ingresar una tarjeta de crédito quedan expuestos a importantes gastos en moneda extranjera por no seguir exactamente las condiciones estipuladas en las bases de estas promociones.
- Los cursos masivos y abiertos se dictan mayoritariamente en inglés. Aunque se está avanzando en la generación de cursos en castellano, la oferta no es suficiente aún y los que existen son principalmente traducciones de cursos en inglés o están confeccionados para estudiantes de otros países e incluyen referencias culturales ajenas como “el día de acción de gracias”, llamar primavera a la estación que comienza con el equinoccio de marzo, etc. y dejan de lado temáticas locales incluídas en este curso como el juego de Truco, la Generala, o referencias geográficas a nuestro territorio.
- El curso ofrecido por la UNSAM tiene el aval de una universidad pública nacional, con un programa que especifica entre otras cosas los doctores que están a cargo del curso, la carga horaria semanal y cuatrimestral, los mecanismos de evaluación (dos exámenes parciales para todos más un examen final para los estudiantes de posgrado), y otros detalles que resultan fundamentales para que el curso sea aceptado por casi cualquier universidad como materia electiva de grado y posgrado.

En resumen, el curso que hemos dictado tiene características que lo hacen único: es gratuito (sin requerir pagos ni tarjetas de crédito), abierto (cualquiera puede inscribirse), masivo (no tiene cupo), se dicta en castellano y es válido como materia electiva de grado y posgrado en casi cualquier universidad nacional. En particular no conocemos ninguna experiencia previa de este tipo generada desde una Universidad Nacional.

Metodología

Programa y organización del curso

El objetivo original del curso era enseñar los fundamentos del lenguaje Python orientado al manejo de datos, a la escritura de scripts y a una organización adecuada de los programas; enseñar algunos rudimentos de la teoría de algoritmos, incluyendo conceptos básicos de la teoría de la complejidad y algunas estructuras de datos no triviales; e introducir la programación orientada a objetos. Nos propusimos que al final del curso los estudiantes pudieran escribir sus propios programas sencillos en Python así como leer y modificar programas escritos por otras personas. Enseñamos bajo la convicción de que una adecuada enseñanza de la programación debe incluir algunos conceptos teóricos sobre algoritmos, complejidad computacional y estructuras de datos. El advenimiento de la pandemia y la consecuente virtualización de los contenidos nos forzó a una reorganización completa del material, incluyendo el diseño y la redacción de material escrito y la confección de nuevas guías de ejercicios. Sin embargo, y a pesar de haber modificado la modalidad, la metodología de enseñanza, el cronograma y el público que podía acceder al curso, mantuvimos intactos tanto los objetivos del curso como sus contenidos mínimos ya que su programa había sido aprobado formalmente con anterioridad por el Consejo de Escuela de la Universidad.

Las presentaciones de los temas y los ejercicios propuestos fueron organizados de forma que en distintas instancias se retoma el hilo de lo visto anteriormente, proponiendo una modificación, generalización o extensión de un problema ya resuelto. En este sentido, la estrategia utilizada apunta a lograr un aprendizaje iterativo incremental (cfr. Brennan y

Resnick, 2012).¹

A modo de ejemplo podemos mencionar un ejercicio que consiste en leer un archivo CSV que presenta el contenido de un camión que provee a una verdulería. Inicialmente, la lectura se hace de forma manual (leer archivo línea por línea, campo por campo); después se utilizan herramientas propias de Python para trabajar en este formato (`csv.reader`); luego se escribe un programa para ejecutarlo desde una terminal, incorporando herramientas que permiten el pasaje de argumentos desde la línea de comandos a un programa. Como estructura donde se guardan estos datos se utilizan primero listas de cadenas de caracteres, luego diccionarios, luego DataFrames de la biblioteca Pandas y finalmente se realiza la tarea con Objetos mediante el diseño de Clases apropiadas. Además, el mismo ejemplo es usado para discutir problemas relacionados con búsquedas, ordenamientos, manejo de errores, y otros temas.

El camino planteado fue organizado en doce unidades. Estas unidades se organizaron en doce semanas, agregando dos semanas de repaso y dos semanas de evaluación, conformando un curso de 16 semanas de duración. En la Tabla 1 puede verse el contenido de cada unidad.

El contenido de este curso y sus ejercicios se basan en experiencias previas nuestras y de otros docentes. En particular tomamos ideas del curso histórico de algoritmos y programación de la Facultad de Ingeniería de la UBA (cfr. Wachenchauzer *et al.*, 2018), así como el enfoque práctico que propone Beazley (2020) en su curso en inglés de Python práctico. De ambos cursos tomamos fragmentos para construir el nuestro también bajo una licencia Creative Commons,² que permite gestionar el conocimiento fuera de la lógica del mercado, en sistemas sociales autoorganizados.

Para el análisis de datos, modelado y generación de gráficos usamos diversas bibliotecas de Python y los ejemplos en varios casos están tomados de la documentación oficial de estos paquetes así como de las clases y tutoriales provistos por SciPy (Varoquaux, 2015 y Virtanen, 2020). Nuevamente, esta documentación es compartida con licencias Creative Commons o BSD.³ También nos basamos en ideas y ejemplos del proyecto Exactas Programa⁴ (Arrar *et al.*, 2018 y Lopez-Rosenfeld *et al.*, 2020), que fueron desarrollados por docentes de diversas

1 Cfr. <https://www.agilealliance.org/agile101/agile-glossary/>

2 Cfr. <http://www.creativecommons.org.ar/>

3 Cfr. https://es.wikipedia.org/wiki/Licencia_BSD

4 Cfr. <http://exactasprograma.exactas.uba.ar/>

carreras de la Facultad de Ciencias Exactas y Naturales de la Universidad de Buenos Aires y cuyos autores han compartido generosamente.

Unidad	Contenido
Introducción a Python	El intérprete interactivo de Python; Crear, editar y ejecutar pequeños programas; Algunos tipos de datos de Python: número enteros, números de punto flotante, cadenas y listas.
Funciones y estructuras	Archivos; Funciones; Tipos y estructuras de datos; Contenedores.
Trabajando con datos	Secuencias; Contadores; Formatos de impresión; Uso de un entorno de desarrollo integrado; Caso de estudio: Arbolado porteño.
Crear y operar con listas	Errores y excepciones; Listas; Comprensión de listas; El problema de la búsqueda en una lista; La búsqueda lineal; Todo es un objeto; Caso de estudio: Arbolado porteño.
Aleatoriedad	El debugger de Python; El módulo random; El módulo numpy; Caso de estudio: Album de Figuritas; Algo sobre gráficos en Python.
Complejidad y Organización de programas	Scripting; Funciones; Módulos; Complejidad de algoritmos: la Búsqueda lineal vs. la búsqueda binaria; Más sobre gráficos.
Diseño, especificación, documentación y estilo.	Control de errores; El módulo principal; Cuestiones de diseño; Contratos: Especificación y documentación; Estilos de codeo; El módulo matplotlib.
Fechas, Carpetas y Pandas	Manejo de fechas y horas; Manejo de archivos y carpetas; Ordenar archivos en Python; El módulo pandas. Optativo: Series temporales.
Clases y objetos	Clases y Objetos; Herencia; Métodos especiales; Caso de estudio: Pilas y colas. Optativo: Modelos de ecología de poblaciones.
Generadores e iteradores	El protocolo de iteración; Iteración a medida; Productores, consumidores y cañerías. Optativo: Teledetección.
Recursión y regresión	Intro a la Recursión; Diseño de algoritmos recursivos. Optativo: Regresión lineal.
Ordenamiento	Ordenamientos sencillos de listas; Divide y reinarás; El algoritmo merge-sort. Optativo: Algoritmos de clasificación.

Tabla 1. Las doce unidades del curso "Programación en Python".

Dinámica del curso

El volumen de estudiantes que se anotaron hacía imposible dictar clases sincrónicas interactivas: no sólo a nivel didáctico (es imposible responder preguntas de 1000 personas) sino también tecnológico (las plataformas de teleconferencia admiten entre 100 y 500 miembros por sesión). Para poder dictar los contenidos en forma virtual fue necesario pensar en una versión mayormente asincrónica.

La filosofía del curso se basó en la idea de que en Ciencias de la Computación (y probablemente en otras disciplinas y artes) los estudiantes realizan un aprendizaje más agradable y profundo al aplicar los conceptos de manera práctica por ellos mismos que solamente leyendo o viendo ejemplos confeccionados por otros (Koedinger, 2015; Arrar *et al.*, 2018).

El soporte elegido para el material del curso fue el de un apunte exhaustivo publicado en línea en un repositorio público (Lopez-Rosenfeld *et al.*, 2020).⁵ Este material incluye las explicaciones teóricas de cada unidad intercaladas con ejemplos y ejercicios sencillos para acompañar la lectura y facilitar la comprensión de los temas, así como una ejercitación más profunda al final de cada sección, con algunos ejercicios seleccionados para una entrega semanal obligatoria.

El material escrito se complementó con encuentros virtuales semanales de dos horas de duración por videoconferencia. Las grabaciones de estos encuentros quedaron disponibles en YouTube, dando la posibilidad de realizar toda la cursada de modo completamente asincrónico, a excepción de los exámenes parciales que fueron en tiempo real.

Consideramos que la comunicación con los estudiantes es fundamental en todo curso y más aún en un contexto de virtualidad donde no es posible siquiera ver sus caras para poder percibir si entienden o no cada explicación. Por este motivo, fue fundamental contar con diversos canales de comunicación. Estos canales sirvieron no sólo para evacuar dudas sino también para saber cómo se vivía el curso desde el lado de los estudiantes. Sobre esto comentaremos en la próxima sección.

⁵ Cfr. https://github.com/python-unsam/Programacion_en_Python_UNSAM/blob/master/Notas/Contenidos.md

Diseño y presentación del material

Teniendo en cuenta que el material redactado sería leído por los alumnos en la computadora, optamos por un diseño continuo del texto. Descartamos la diagramación del texto que veníamos usando pensado para el tamaño de una hoja impresa A4, escrito en LaTeX⁶ y compilado a PDF y optamos por pensar el texto para ser leído en una pantalla, sin cortes de página y con una manipulación cómoda de los hipervínculos. Es por eso que el material fue escrito y editado en el lenguaje Markdown y compartido en páginas web.⁷

Para el almacenamiento y la difusión del material, utilizamos un repositorio GitHub.⁸ Esto hizo posible que, a diferencia de cómo se trabajaba anteriormente, el equipo docente pudiera realizar una escritura y corrección colaborativa del material de forma concurrente. Al revisar el texto o los ejercicios, un docente puede realizar cambios en tiempo real y de forma simultánea con otros docentes, a medida que se encuentran errores o se añaden contenidos.

Consultas e interacción

Tratamos de crear un clima que permitiera poner en práctica leyes de convivencia fundamentales promovidas, entre otros, por Greg Wilson (2019). De esta manera generamos un ambiente donde la empatía es una de las herramientas fundamentales para compartir y generar conocimiento, creando una comunidad de aprendizaje que brinda apoyo horizontalmente.

Debido a la disparidad en la relación docentes-estudiantes (inicialmente contábamos con un docente cada 250 estudiantes), fue necesario tener algunas estructuras autogestionadas de trabajo que funcionaran como primer lugar de consulta y asistencia entre estudiantes. En este sentido, en la presentación del curso les recomendamos a los estudiantes que crearan foros y redes entre ellos para discutir los problemas (sin compartir soluciones) y las dudas que surgieran en la materia. Esta invitación fue tomada instantáneamente y se crearon múltiples grupos en diversas plataformas (WhatsApp, Telegram, Discord) que funcionaban extraoficialmente y en los cuales no había docentes participando. Estos grupos funcionaron de acompañamiento y gran apoyo entre pares.

6 Cfr. <https://www.latex-project.org/>

7 Cfr. <https://www.markdownguide.org/>

8 Cfr. <https://github.com/>

Fue gratificante comprobar que cumplieron con la consigna de no pasar soluciones, y que además utilizaron estas plataformas para intercambiar métodos para acercarse a soluciones. Frases comunes eran “a mí me sirvió pensarlo así” o “fíjate qué pasa si considerás tal cosa”.

En un segundo escalón en la estructura de la comunicación, oficialmente teníamos un espacio de trabajo en la plataforma Slack para las consultas que no pudieran responderse en estos foros de estudiantes. Completaron la inscripción e ingresaron a la plataforma Slack algo más de ochocientos estudiantes.

La propuesta de que trataran de evacuar sus dudas en los grupos pequeños y llegaron a Slack las dudas remanentes funcionó muy bien. Si bien dedicamos muchísimo tiempo a responder consultas, la dinámica jerarquizada logró reducir en un orden de magnitud la cantidad de consultas que tuvimos que responder, permitiéndonos así dar abasto con las mismas.

La organización del Slack permitió además que las consultas estuvieran ordenadas de manera tal que otros estudiantes pudieran buscar y revisarlas fácilmente, aprovecharlas, incluso comentar sobre ellas, lo que evitó que se repitieran las consultas. Para esto, cada unidad tenía asociado un canal propio dentro de Slack. Además, la plataforma organiza las preguntas y sus respuestas en hilos (o *threads*) de forma que las respuestas a una pregunta quedan supeditadas a la misma y es sencillo diferenciarlas de otros temas. Finalmente, dentro de la plataforma Slack, se armaron también canales sobre los ejercicios optativos o temas más avanzados que fueron utilizados por grupos minoritarios de estudiantes.

Toda la comunicación oficial de la materia fue por medio de Slack, exceptuando las calificaciones que fueron comunicadas mediante emails individuales.

Los encuentros sincrónicos

Los encuentros sincrónicos se realizaban utilizando la plataforma Zoom y a su vez realizando *streaming* a YouTube. Cada encuentro se dividía en dos partes: la primera parte era de repaso de la unidad anterior y resolución de los ejercicios que habían suscitado más dudas, y la segunda de presentación de la unidad siguiente.

Para la elección de los temas o ejercicios a resolver en la primera parte utilizamos un

sistema de votación para poder identificar las dudas más comunes (usando la plataforma Slido).⁹ Cada estudiante podía proponer un ejercicio, alguna duda o un tema para que se retomara y además podía votar sobre las propuestas existentes (cfr. Figura 2). Al momento del encuentro se abordaban las preguntas más votadas entre todas las preguntas propuestas. Estas soluciones se exponían programando en vivo (*live-coding* compartiendo pantalla) lo que permitía explicar el razonamiento detrás del código a medida que se lo desarrolla.



Figura 2. Ejemplo de las dudas planteadas para discutir en clase.

En la segunda parte de la clase sincrónica, se realizaba una presentación somera de los temas nuevos que encontrarían durante la semana siguiente para que pudieran anticipar de qué

⁹ Cfr. <https://www.sli.do/>

iba a tratar la unidad y la introducción de los nuevos conceptos fuera más amable.

Sistema de evaluación

Entendimos que la modalidad usual de evaluación consistente en exámenes parciales con ejercicios a desarrollar no resultaría adecuada para una cursada a distancia y con tantos estudiantes. Por este motivo y para fomentar un aprendizaje basado en la ejercitación (Koedinger, 2015) planteamos un sistema de evaluación combinado a lo largo de la cursada.

Por un lado, realizamos una evaluación permanente con entregas semanales (cfr. Blesa, 2015). Fueron entre cuatro y seis ejercicios de entrega obligatoria por semana llegando en total a 56 ejercicios seleccionados a lo largo de todo el curso. Además de evaluar estas entregas semanales, tomamos dos exámenes parciales de forma virtual y sincrónica con modalidad de opción múltiple lo que permitió una corrección automatizada acorde al volumen de alumnos. Las semanas previas a estos exámenes funcionaron también como un período de repaso y de puesta al día, por lo que les dimos a los alumnos la posibilidad de entregar ejercicios atrasados pendientes. Para aprobar la materia resultó necesario haber realizado adecuadamente el 70% de los ejercicios obligatorios y haber aprobado ambos exámenes parciales, lo que garantizó un nivel elevado de conocimiento por parte de los aprobados.

Dado que era inviable dar una devolución de cada uno de los ejercicios entregados de una semana a la siguiente, establecimos también un mecanismo optativo de revisión entre pares. Seleccionamos de cada unidad un ejercicio que fuera conveniente para este fin. Al momento de realizar la entrega semanal, cada estudiante podía optar por participar de la revisión por pares de la semana. En caso de participar, además de enviar su solución para que sea revisada, se comprometía a revisar, en el transcurso de la semana, las resoluciones de dos estudiantes. Estas revisiones se plantearon inicialmente con el método de doble ciego, pero se daba la posibilidad de que el revisor decidiera firmar su revisión para poder continuar un diálogo sobre los ejercicios una vez terminado el proceso de revisión. Para las entregas de ejercicios y su evaluación por pares diseñamos un sistema de envíos basado en Google Apps Script.

Ejercicios optativos

Dada la diversidad de conocimientos previos en programación e intereses que presentaron los alumnos, y considerando que se trata de una materia para alumnos de diversas carreras de grado y posgrado, además de los ejercicios de resolución obligatoria que llegaban a cubrir conceptos básicos de la teoría de algoritmos, así como su análisis y la utilización de algunas estructuras de datos no triviales, incluimos ejercicios que presentaban aplicaciones de lo aprendido en la materia a temas más avanzados incluyendo modelos de regresión lineal, el análisis de series temporales, el procesamiento de imágenes satelitales, metodologías de clasificación en el marco del aprendizaje automático y modelos sencillos de ecología de poblaciones.

Estos ejercicios los presentamos semanalmente durante la segunda mitad del curso y su resolución era opcional. Los mismos fueron diseñados para aumentar la motivación de los estudiantes bajo la convicción de que la motivación es uno de los grandes facilitadores del aprendizaje de la programación (Jenkins, 2002).

Resultados

Valoración cuantitativa de la cursada

En este sistema nuevo de dictado de la materia, al igual que en todas las modalidades, existió cierto nivel de deserción. Evolución de la cantidad de estudiantes:

- Preinscriptos: 1287.
- Inscriptos: 887.
- Inscriptos que entregaron al menos un ejercicio: 682.
- Aprobaron primer parcial: 334.
- Aprobaron la materia completa: 273.

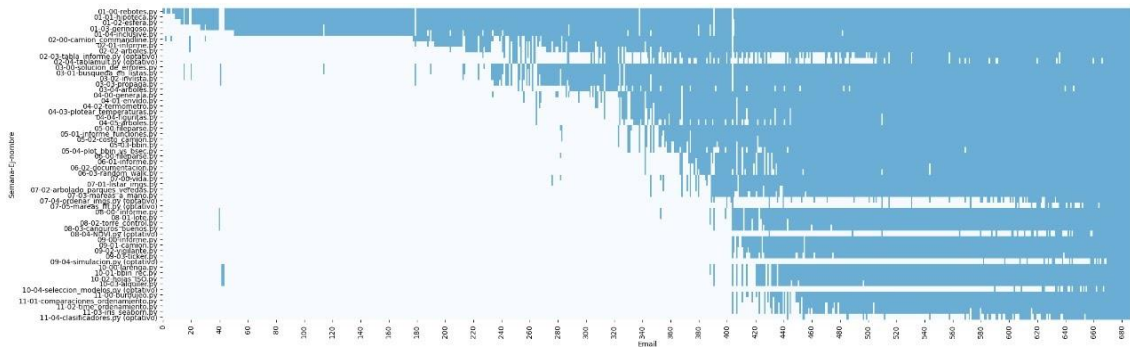


Figura 3. Entregas de ejercicios por parte de los 682 alumnos que realizaron al menos un envío. En azul las entregas realizadas. Eje X: estudiantes. Eje Y: ejercicios. Se observa un marcado abandono especialmente en las primeras unidades.

A fines realistas, podemos considerar que comenzaron la materia 682 estudiantes que fueron los que entregaron al menos un ejercicio. De estos, un 50% aproximadamente aprobó el primer parcial y solo el 40% logró cumplir completamente con los requisitos de aprobación de la materia. La Figura 3 muestra las entregas realizadas por los estudiantes a lo largo del cuatrimestre donde se puede observar una gran deserción luego de la primera unidad.

A continuación presentamos una tabla con los números de aprobados discriminados por institución (Escuela de Ciencia y Tecnología de la UNSAM, otra unidad académica de la UNSAM o no perteneciente a la comunidad UNSAM) y tipo de estudios cursados (cfr. Tabla 2). Se aprecia una gran cantidad de estudiantes de doctorado e investigadores, así como de estudiantes de fuera de la comunidad UNSAM.

Aprobados	ECyT UNSAM	Otro UNSAM	Otro	Total
Grado	50	7	62	119
Doctorado	6	7	90	103
Postdoc, Investigador o Profesional	2	2	47	51
Total	58	16	199	273

Tabla 2. Aprobados según estudios previos.

Entusiasmo

En dos oportunidades fue notable el entusiasmo y el agradecimiento de los estudiantes por poder participar de esta experiencia. La primera oportunidad se dio al ingresar al espacio virtual en Slack (a principios de cuatrimestre). En este primer contacto con el curso y sus pares, los alumnos espontáneamente se presentaban mencionando brevemente qué estudiaban, de dónde eran, qué les interesaba y por qué estaban allí. La variedad geográfica, etaria y ocupacional resultó atrapante y el entusiasmo con el programa y la modalidad propuesta llamaban la atención.

La segunda oportunidad se dio al cerrar la materia, antes del último examen. Las expresiones espontáneas de agradecimiento no sólo reconocieron y valoraron lo que habían aprendido sino que resaltaron la importancia de la comunidad de aprendizaje, la colaboración entre pares y el trabajo de los docentes que impulsamos esta experiencia.

Discusión y conclusiones:*Demanda de cursos e interior del país*

Se observa una gran demanda, tanto por parte de alumnos de grado como de posgrados en diversas disciplinas, de una materia que les permita aprender a usar la programación como herramienta científica y que sea reconocida por sus casas de estudio como válidas en el marco de sus estudios. Existen algunos talleres y cursos que van en esta dirección (como por ejemplo el Workshop en Técnicas de Programación Científica, el curso de Exactas ya mencionado o los Talleres de Rladies) que suelen llenar sus cupos, dejando demanda insatisfecha. Los cambios tecnológicos y didácticos impulsados por la pandemia nos permitieron ofrecer el curso sin cupos y observar esta enorme demanda insatisfecha existente.

Los cambios mencionados también nos permitieron llegar a estudiantes a los que nunca habíamos llegado ni pensado en llegar desde la UNSAM. El 38% de las personas que aprobaron el curso residen fuera del AMBA, que es la zona natural de influencia de nuestra universidad. Muchos estudiantes del interior del país nos han transmitido gran satisfacción por haber podido cursar a distancia. Hasta el advenimiento de la pandemia, les resultaba en muchos casos

necesario trasladarse físicamente hasta una gran ciudad para realizar sus cursos de posgrado, con todas las limitaciones y dificultades que esto acarrea.

Consideramos que más cursos a distancia de programación son necesarios, y que cualquier proyecto serio de federalización de la ciencia debe tener en cuenta la necesidad de ofrecer cursos de diferentes niveles académicos (pregrado, grado, posgrado) con buen nivel a residentes del interior del país. En este sentido, utilizando las palabras de García Aretio (2007), la educación a distancia puede contribuir a la realización de una educación sin distancias.

El problema de la deserción

Uno de los grandes problemas que nos planteamos resolver en el futuro es la deserción durante las primeras unidades. Creemos que una parte del problema corresponde a gente que tenía expectativas diferentes del curso, otra es atribuible a personas con dificultades técnicas o de conectividad insalvables. Estos dos orígenes escapan a nuestra capacidad de dar respuestas. Sin embargo, una porción de la deserción parece responder a personas que no tuvieron la contención suficiente durante las primeras unidades y puede resolverse con un mayor acompañamiento docente o comunitario.

La materia fue llevada adelante con solo dos cargos docentes rentados. Consideramos que un equipo docente más numeroso es necesario para poder atender adecuadamente a las diferentes personas y sus diferentes necesidades, especialmente durante las primeras unidades.

Finalmente, consideramos que una parte de la deserción se debe a estudiantes que no contaban con los conocimientos previos suficientes para afrontar este curso. En esos casos, creemos que resultaría conveniente impartir cursos más elementales de introducción a la informática con un espíritu similar al aquí expuesto.

El rol de la comunidad

Creemos que parte del éxito del curso se debió a que se formó una comunidad de pares entre los estudiantes. Esto permitió que se respondieran dudas entre ellos, que se dieran contención y que intercambiaran información de una manera horizontal. Estas estrategias ayudaron a bajar la deserción considerablemente luego de algunas unidades en parte debido al

establecimiento de esta comunidad (Wilson, 2019).

Consideramos que fueron tres los principales dispositivos que favorecieron la creación y establecimiento de esta comunidad de pares: la evaluación semanal que forzó una compenetración con el material desde el principio del curso, la propuesta por parte del equipo docente del doble sistema de grupos de consultas (uno de pares y otro con docentes) y las evaluaciones por pares que fomentaron el diálogo más allá de los vínculos naturalmente establecidos por los estudiantes.

Interdisciplinariedad

El equipo docente estuvo compuesto por investigadores formados de diversas disciplinas incluyendo las ciencias de la computación, la matemática, la biología, las neurociencias y las ciencias ambientales. Creemos que esta diversidad nos permitió tratar los diferentes temas desde un enfoque a la vez práctico y profundo, orientando las unidades hacia contenidos adecuados para estudiantes de grado y posgrado de diversas disciplinas científicas e ingenierías y promoviendo la participación de los diversos estudiantes en el curso.

Es sabido que la pandemia de COVID-19 ha destrozado muchas de nuestras mejores costumbres y prácticas. Esta experiencia muestra que también ha abierto la posibilidad de nuevas prácticas docentes, más horizontales, federales e inclusivas.

Continuación 2021

El primer cuatrimestre de 2021 estamos volviendo a abrir la inscripción a la materia. La Universidad Nacional de San Martín ha podido asignar tres nuevos docentes auxiliares a la materia y la cantidad de inscriptos ha incluso superado la del cuatrimestre anterior.

Bibliografía

- Arrar, M. *et al.* (2018). “Taller de resolución de problemas computacionales para ingresantes a carreras de Ciencias Exactas” (ponencia). Jornadas Argentinas de Didáctica de la Programación.
- Beazley, D. (2020). *Practical Python*. Disponible en: <https://github.com/dabeaz-course/practical-python>

[Fecha de consulta: 08/06/2021].

Blesa, M. J., Duch, A., Gabarró, J., Petit, J. y Serna, M. (2015). "Continuous assessment in the evolution of a CS1 course: The pass rate/workload ratio" (313-332). En *International Conference on Computer Supported Education*. Springer: Cham.

Brennan, K. y Resnick, M. (2012). "New frameworks for studying and assessing the development of computational thinking" (ponencia). *Proceedings of the 2012 annual meeting of the American educational research association*. Vancouver.

García Aretio, L. (2017). "Educación a distancia y virtual: calidad, disrupción, aprendizajes adaptativo y móvil". *RIED. Revista Iberoamericana de Educación a Distancia*, 20 (2): 9-25.

García Aretio, L., Ruiz Corbella, M. y Domínguez Figaredo, D. (2007). *De la educación a distancia a la educación virtual*. Buenos Aires: Ariel.

González, A. H. y Martín, M. M. (2017). Educación superior a distancia en Argentina: tensiones y oportunidades. *Trayectorias universitarias*, 3 (4): 3-11.

Jenkins, T. (2002). "On the difficulty of learning to program". *Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences*, 4: 53-58.

Koedinger K., Kim K., Zhuxin Jia, J., McLaughlin, E. y Bier N. (2015). "Learning is Not a Spectator Sport: Doing is Better than Watching for Learning from a MOOC". *Proceedings of the Second (2015) ACM Conference on Learning*: 111-120.

Lopez-Rosenfeld, M. et al. (2020). "Exactas Programa: llevando la programación a cada rincón de la ciencia" (ponencia). *Simposio Argentino de Educación en Informática. Jornadas Argentinas de Informática*.

Marotias, A. (2018). "El rol de la educación a distancia en la universidad pública argentina (1986-2016)" (ponencia). *V Simposio Argentino sobre Tecnología y Sociedad (STS 2018)-JAIIO 47*. Buenos Aires

Pérez-López, E., Atochero, A. V. y Rivero, S. C. (2021). "Educación a distancia en tiempos de COVID-19: Análisis desde la perspectiva de los estudiantes universitarios". *Revista Iberoamericana de Educación a Distancia*, 24 (1): 331-350.

Rivero, M. A., Ortiz, R. F. (2020). "Educación a distancia: evolución del contexto normativo en la República Argentina y del marco institucional en la Universidad Nacional del Sur. *Revista Perspectivas de las Ciencias Económicas y Jurídicas*". *RIDCA*. Disponible en: <http://repositoriodigital.uns.edu.ar/handle/123456789/4761> [Fecha de consulta: 08/06/2021].

Varoquaux, G., Gouillart, E., Vahtras, O., Haenel, V., Rougier, N. et al. (2021). *Scipy Lecture Notes: One document to learn numerics, science, and data with Python*. Disponible en: <https://scipy-lectures.org/> [Fecha de consulta: 08/06/2021].

Virtanen, P. et al. (2020). "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". *Nature Methods*, 17 (3): 261-272. Disponible en: <https://www.nature.com/articles/s41592-019-0686-2> [Fecha de consulta: 08/06/2021].

Wachenchauer, R., Manterola, M., Curia, M., Medrano, M., Paez, N., Essaya, D., Simó, D. y Santisi, S. (2018). "Algoritmos y Programación I". Disponible en: <https://drive.google.com/file/d/0B0KKEIBDHL7tdEQ3bFZ2M3VrZzA/view> [Fecha de consulta: 08/06/2021].

Wilson, G. (2019). *Teaching Tech Together: How to Make Your Lessons Work and Build a Teaching Community around Them*. Florida: CRC Press. Disponible en: <https://teachtogether.tech/> [Fecha de consulta: 08/06/2021].