Exactas Programa: llevando la programación a cada rincón de la ciencia

Matías Lopez-Rosenfeld, Esteban Mocskos[0000-0002-6473-7672], Mariano González Lebrero, José Crespo[0000-0002-0074-6924], Mehrnoosh Arrar[0000-0001-6404-0552], Inés Caridi[0000-0001-6695-0971], and Mariela Sued

Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires, Argentina
exactasprograma@dc.uba.ar

Resumen La computadora se ha instalado en casi cualquier disciplina en los ámbitos científicos, gubernamentales e industriales como una herramienta fundamental. Sin embargo, se nota una falta de aprovechamiento del potencial de la utilización de la computadora dentro de los contenidos que se dictan en las diferentes carreras de otras instituciones del país, como la Facultad de Ciencias Exactas y Naturales de la Universidad de Buenos Aires. Si saber programar para resolver o analizar situaciones es una actividad que realizan los graduados de todas las disciplinas, ¿como es posible que no se les enseñen los conocimientos necesarios para hacerlo de manera eficiente y sin que represente un escollo para su desarrollo profesional?. Con este objetivo en mente, un grupo de docentes y estudiantes de doctorado de distintas disciplinas propusimos un taller orientado a brindar a todos los alumnos, más allá de la carrera que estudie, los elementos básicos de uso de la programación para resolver problemas. *Exactas Programa* es una iniciativa multidisciplinaria que combina la propuesta de diferentes actividades que motivan la utilizacion de la computadora como asistente en el análisis y resolución de distintos desafíos. En este trabajo, compartimos la estructura de nuestra propuesta, los detalles de cada una de las actividades que lo constituyen y las distintas situaciones que se consideraron para su adaptación luego de haberse dictado en cinco oportunidades.

Keywords: Enseñanza Programación · Ciencia · Actividades Lúdicas.

1. Introducción

La utilización de la computadora como una herramienta ha permeado, en mayor o menor grado, todos los ámbitos de las ciencias e ingeniería. Hoy en día, se ha transformado en uno de los pilares que sustenta el desarrollo y la innovación. Pero, además, la programación constituye una herramienta invalorable en el proceso de aprendizaje en prácticamente todas las áreas del conocimiento. Una persona capaz de trabajar con algoritmos dispone de una formación que lo prepara para mucho más que escribir buenos programas, dispone de un instrumento de propósito general que será definitorio en su desarrollo.

Suele decirse que un tema realmente se comprende cuando existe un otro a quien enseñárselo. Desarrollar un programa es una forma de enseñar a una computadora. El aspecto clave es que, para poder escribir un programa, se requiere de una profunda comprensión del problema. Además, durante el mismo proceso de desarrollo, el programador está obligado a ordenar y clarificar los pasos necesarios para resolver, lo que ayuda a ordenar el pensamiento.

En el caso de una persona dedicada a la actividad científica, la capacidad de programar resulta ser una condición altamente valorada. Inclusive, en varias áreas del conocimiento, se vuelve imprescindible para poder desempeñarse.

Convencidos de los beneficios que la programación ofrece, un grupo de docentes, investigadores y doctorandos de distintos departamentos de la Facultad de Ciencias Exactas y Naturales de la Universidad de Buenos Aires (FCEN-UBA) diseñamos una experiencia piloto con el objetivo de que distintos miembros de nuestra comunidad aprendieran a resolver problemas utilizando un lenguaje de programación.

El objetivo último es que, a mediano plazo, este taller favorezca la incorporación de prácticas computacionales dentro del contenido de las diferentes asignaturas de las carreras dictadas en la FCEN-UBA. Buscamos que aprender a programar sea una herramienta al alcance de todos los estudiantes y que su utilización sea transversal a todas las disciplinas.

Esta idea es compartida en diversos ámbitos de la sociedad, y tal es así que en 2015, el Consejo Federal de Educación de la República Argentina declaró para ese año a *la enseñanza y el aprendizaje de la programación como de importancia estratégica para el sistema educativo argentino durante la escolaridad obligatoria, con el fin de fortalecer el desarrollo económico y social de nuestro país* [1]. Como esta norma no alcanza al ámbito universitario –son pocas las currículas universitarias que incluyen contenidos relacionados con la programación– y la mayoría de los nuevos ingresantes no tuvo programación en sus escuelas, planteamos una propuesta que sustente una aproximación a la resolución de diferentes tipos de situaciones utilizando una computadora de manera que pueda ser realizada por cualquier ingresante a la FCEN-UBA.

En este trabajo describimos este proyecto, *Exactas Programa*, un taller para resolver problemas de distintas disciplinas científicas usando la computadora. A continuación explicamos cuáles son los objetivos, cómo concebimos las actividades y la manera en la que las implementamos.

2. Nuestra propuesta

El objetivo fundacional de *Exactas Programa* es incorporar la programación en la formación de nuestros alumnos desde el mismo inicio de sus carreras. Buscamos presentar herramientas computacionales que les permitan resolver diferentes problemas, promoviendo el razonamiento algorítmico y la implementación de código propio en un lenguaje de programación. Esta aproximación facilitará posteriormente la inclusión de nuevas prácticas basadas en la utilización de programación dentro de las materias dictadas en la FCEN.

Para ello diseñamos un taller que consiste en nueve encuentros de cuatro horas cada uno, a razón de tres veces por semana. A la fecha, se han dictado cinco ediciones durante los recesos de verano e invierno, a partir de la experiencia piloto lanzada en el verano de 2018. Cabe destacar que el curso no otorga puntaje a los estudiantes ni es obligatorio, por lo cual la asistencia del público está motivada sólo por el interés en la temática.

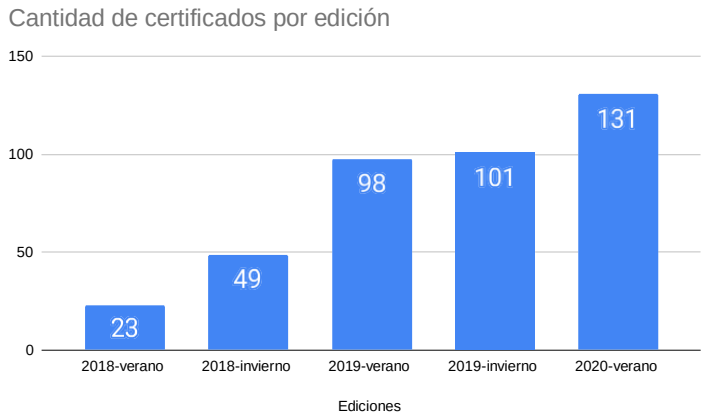

Figura 1. Cantidad de estudiantes que finalizaron la actividad en las distintas ediciones del taller. Esta propuesta se diseñó para que pueda dictarse durante los recesos (invierno y verano).

El curso no se plantea como una introducción a la programación, se introducen nociones básicas de programación como medio para resolver distintas situaciones desafiantes. La modalidad de las clases es teórico - práctica y se desarrollan en un laboratorio con computadoras donde algunos de los estudiantes utilizan las computadoras ya instaladas mientras que otros lo realizan usando su propia computadora portátil. En cualquier caso, una de las condiciones críticas es que cada participante utilice una computadora y realice las actividades individualmente, de forma tal que el aprendizaje sea activo y que cada asistente se vea en la necesidad de escribir su propio código.

Los alumnos reciben asistencia personalizada del equipo de docentes durante el desarrollo de las actividades de manera de transformar sus ideas de resolución, en código que pueda responder las preguntas planteadas. Para poder asistir y acompañar las necesidades de los participantes, contamos con un docente cada ocho alumnos; una relación poco habitual en ámbitos como el nuestro pero que ha resultado fundamental para transitar de forma satisfactoria el vertiginoso camino que propone el taller. La relación docente-alumno es clave para poder lograr que un alumno sin conocimientos previos en programación pueda estar escribiendo programas relativamente complejos ya en la segunda clase.

A lo largo de los nueve encuentros, se plantean siete actividades en la forma de problemas/desafíos:

i) Jugar al Black Jack.
ii) Simular el llenado de un álbum de figuritas.
iii) Modelar una dinámica de incendio de bosques.
iv) Modelar y visualizar la órbita de la tierra alrededor del Sol.
v) Simular la evolución de una avalancha.
vi) Modelar la interacción predador-presa en tiempo y espacio.
vii) Simular una dinámica de la interacción de partículas en dos dimensiones.

Cada una de estas actividades es presentada con un interrogante inicial que da origen a una serie de preguntas. Luego de una breve presentación, queda clara la necesidad de tener un programa que nos ayude a responder las preguntas de interés. Por ejemplo, ¿cuál es la probabilidad de llenar el álbum del mundial comprando 80 paquetes de figuritas? [1]

Las siete actividades mencionadas están formuladas de manera atractiva incluyendo, en cada caso, una actividad lúdica que permite a los estudiantes acercarse al problema de forma tangible, ya sea jugando o experimentando con distintos materiales *unplugged* : papeles, bolitas, cartas, dados, caramelos, etc. La posibilidad de experimentar y representar en vivo da un fuerte soporte para desarrollar luego los algoritmos y escribir los programas que puedan resolver las consignas propuestas, siguiendo y resolviendo durante el resto de la clase una guía práctica cuidadosamente elaborada. La parte lúdica ayuda a tener en claro no sólo la pregunta importante sino también la dinámica del sistema que se quiere estudiar. Además de servir como forma de entender los detalles del problema que se está atacando, esta sección también ayuda a crear un lazo de cordialidad en el grupo, mejorando el clima de estudio y el intercambio de ideas. Finalmente ayuda para que los docentes, durante las consultas, puedan referirse a momentos concretos durante las dudas (por ejemplo, “¿te acordás que cuando jugamos...?”).

La tabla 1 resume el plan general de la mayoría de las clases. Los encuentros tienen una serie de actividades programadas que permiten presentar la actividad, entender los detalles del sistema a estudiar y programar. Los encuentros que no siguen esta estructura son la primera clase y la última. En el caso de la primera, se brindan los conceptos necesarios de programación de manera que un estudiante sin conocimientos previos, pueda escribir sus primeros programas. La última clase también es distinta ya que se realiza un cierre de la actividad con todo el grupo.

El equipo docente tiene una composición interdisciplinaria que otorga una gran riqueza de enfoques y muestra las distintas maneras en las que cada disciplina puede valerse de la programación, dejando en evidencia que esta no es

[1] Este interrogante suele ser abordado cada cuatro años en los medios de comunicación: Mundial 2014 https://www.pagina12.com.ar/diario/contratapa/13-250187-2014-07-06.html; Mundial 2018 https://www.lanacion.com.ar/2125275-rusia-2018-cuantos-sobres-de-figuritas-hacen-falta-para-llenar-el-album-del-mundial

Tabla 1. Estructura general de las clases del taller. Los encuentros tienen una serie de actividades programadas que permiten presentar la actividad, entender los detalles del sistema a estudiar y realizar la implementación.

Momentos	Producto esperado	Materiales disponibles	Duración
Cierre clase actividad anterior	Marcar situaciones típicas y errores frecuentes	Pizarrón y diapositivas	30 minutos
Presentacion del problema	Reconocer los objetos involucrados	Diapositivas	10 minutos
Planteo de preguntas		Diapositivas y pizarrón	20 minutos
Experimentacion *parte lúdica*	Comprensión de la dinámica involucrada en el problema	Dados, lápiz, papel, caramelos, etc.	15 minutos
Cierre de la experimentacion	Identificar cómo se modelan los objetos involucrados	Pizarrón	10 minutos
Implementacion	Programa que permita responder las preguntas	Guía de ejercicios + computadoras + docentes	2,5 horas

una herramienta exclusiva de los especialistas en computación sino que brinda inmensas posibilidades a todas las áreas. Esta riqueza en la variedad de la formación del equipo docente permite que todos se sientan convocados por alguna de las actividades propuestas, relacionadas con las áreas del conocimiento transversales a las carreras que se enseñan en la Facultad.

La propuesta didáctica *Exactas Programa* se basa en la utilización del lenguaje de programación Python, por ser un lenguaje de código abierto y de creciente uso tanto en la comunidad científica como en la industria. Sin embargo, dado que el foco del curso está en cómo utilizar la computadora para resolver problemas y no en detalles técnicos del lenguaje, no se incluyen conceptos avanzandos típicos de Python (como listas por comprensión, iteradores, etc.). De esta forma, las habilidades adquiridas en el curso en el lenguaje elegido resultan bastante aplicables a otros (R, Matlab, Fortan, etc).

3. Actividades

En el primer encuentro del curso se realiza una presentación de Python, explicando brevemente su sintaxis, los conceptos de asignación, ciclo, condicionales, funciones y algunas nociones básicas de tipos de datos (`int`, `float`, `string`) y listas. Esta clase comienza con un desafío: *¿A qué no terminan la clase programando?*. Desde el principio, se busca complementar los conceptos que se muestran o explican con pruebas realizadas por los alumnos en computadora. Para evitar problemas de instalación de herramientas y demoras en el comienzo, se optó por la utilización de un entorno web que ofrece la posibilidad de escribir y probar programas en Python de manera muy intuitiva. Python Tutor (www.pythontutor.com) es un servicio web que permite escribir, ejecutar y analizar el comportamiento de programas en Python sin la necesidad de tener ninguna herramienta instalada, solo se necesita un navegador y conexión a Internet, ambas disponibles en los laboratorios de computadoras de la FCEN-UBA.

Durante el resto del curso, las actividades pueden ser resueltas mayoritariamente combinando las herramientas vistas en esta clase, con excepción de matrices (para lo que se usa el paquete numpy) y gráficos (con matplotlib) más algunos módulos que se detallan en las actividades. Es importante destacar que durante el curso se hace un gran énfasis en pensar *qué* representa cada variable y lista utilizada, dándole un valor muy fuerte a la semántica con la que se interpretan (por ejemplo, una lista de ceros va a estar representando un álbum de figuritas vacío y esa será la forma de llamarlo, ver más detalle del álbum en el apartado 3.2).

Este taller fue concebido para alumnos sin necesidad de contar con alguna experiencia en programación. Las primeras clases representan un importante desafío en transmitir los conceptos fundamentales de programación de manera sencilla para poder utilizarlos inmediatamente en la resolución de problemas interesantes. Luego, en los restantes encuentros se profundizan conceptos y se proponen desafíos más complejos. Al cabo de pocos encuentros los estudiantes logran visualizar que, con pocos elementos, son capaces de recrear escenarios y conceptos complejos.

A continuación se detallan dos de las actividades propuestas a lo largo del curso. Para cada una de ellas se describe el modelo con el que se va a trabajar, las preguntas que se intentan responder con este modelo, el momento lúdico y qué conceptos de programación son necesarios para completarla.

NanoJack

En esta actividad se simula una simplificación del juego de cartas Black Jack: el juego consiste en que cada jugador reciba cartas cuyos valores están entre 1 y 13 y va sumando sus valores; si en un momento alcanza exactamente el valor 21 gana pero, si se pasa, pierde. Luego de recrear la dinámica del juego para un solo jugador se incrementa la cantidad de jugadores. La pregunta que se busca responder es qué proporción de jugadores ganan a este juego.

Antes de empezar a programar, en el contexto de la parte lúdica de la clase, se arman grupos de 4/5 estudiantes que, junto con un docente que hace de crupier, juegan en vivo al NanoJack. El material necesario es un mazo de cartas francesas por grupo.

Si bien cada grupo define primero cómo va a realizar el juego, una de las opciones es que el crupier le va dando cartas de a una al primer estudiante de la ronda hasta que se cumple con la condición de corte (alcanzar o superar los 21), luego con los demás estudiantes, y a medida que avanza en la ronda, se invita a que los estudiantes relaten lo que está sucediendo expresando en palabras lo que luego tendrá que implementarse en el código. Esta puesta en común inicial es de muy alto nivel y con poco detalle sobre cómo implementarla pero les permite a los estudiantes visualizar sobre un sustrato concreto para luego poder llevarlo a la abstracción del código.

En esta actividad, los estudiantes deben ejercitar listas (creación dinámica y recorrido), ciclos anidados con contadores (en la creación de un mazo de cuatro palos) y ciclos con condiciones booleanas (la suma alcanza o supera los 21), así

como el uso del módulo random para utilizar la función shuffle. Además, ya comienzan a experimentar en cómo probar y saber si su programa hace lo que esperan, realizar el debug de un programa es una tarea esencial que cualquier programador se enfrenta constantemente.

Figuritas

Llenar un álbum de figuritas es una actividad que la gran mayoría de los estudiantes y docentes, sino todos, afirma haber realizado en algún momento de sus vidas. A lo largo de este encuentro proponemos una serie de preguntas sobre este problema y procuramos dar respuesta a las mismas mediante el uso de la simulación computacional. Si bien este es un tema largamente abordado desde la matemática, mencionamos apenas algunas referencias al respecto, sin incursionar en este camino para resolver los desafíos planteados.

Podemos resumir la actividad de juntar figuritas de la siguiente manera: se adquiere un álbum (libro vacío con lugares numerados que hay que completar), se van comprando paquetes de figuritas de a uno, cada paquete contiene figuritas numeradas que deben ser pegadas en el álbum en el lugar correspondiente a su número, se gana cuando se completa el álbum.

El hecho de que las figuritas estén tan difundidas ayuda a que la tarea no requiera explicaciones complejas, y que además suela traer algunos recuerdos en cada uno de los estudiantes a partir de su experiencia: la existencia de *"las difíciles"*, la composición de los paquetes, y muchas preguntas que surgen de manera espontánea por parte de la audiencia.

Después de un breve debate, consensuamos en dar respuesta a las siguientes preguntas: en promedio, ¿cuántos paquetes de figuritas hay que comprar para llenar el álbum? ¿Cuál es la probabilidad de llenar el álbum si puedo comprar a lo sumo 700 paquetes?

La figura 2 presenta el modelo simplificado de álbum usado durante la parte lúdica. Se entrega un álbum con capacidad para seis figuritas y se emula el azar de comprar de a una figurita tirando un dado. Los alumnos deben calcular cuántas *figus* tuvieron que comprar para llenar el álbum y luego obtener el promedio de la cantidad necesaria. El material didáctico consiste en un álbum de papel con figuritas para llenar y un dado por cada grupo de dos ó tres estudiantes.

Para alcanzar el modelo final (álbum grande y paquetes) se realiza una primer implementación donde las *figus* se compran de a una (siguiendo la lógica del juego realizado), y luego se lo extiende a figuritas en paquetes.

En conjunto, estas dos actividades descriptas representan el núcleo de la lógica del curso y donde los estudiantes incorporan la mayor cantidad de herramientas.

4. Resultados y Discusión

Desde el inicio del taller piloto en el verano de 2018 planteamos el ambicioso objetivo de que, en el largo plazo, todos los ingresantes de nuestra Facultad

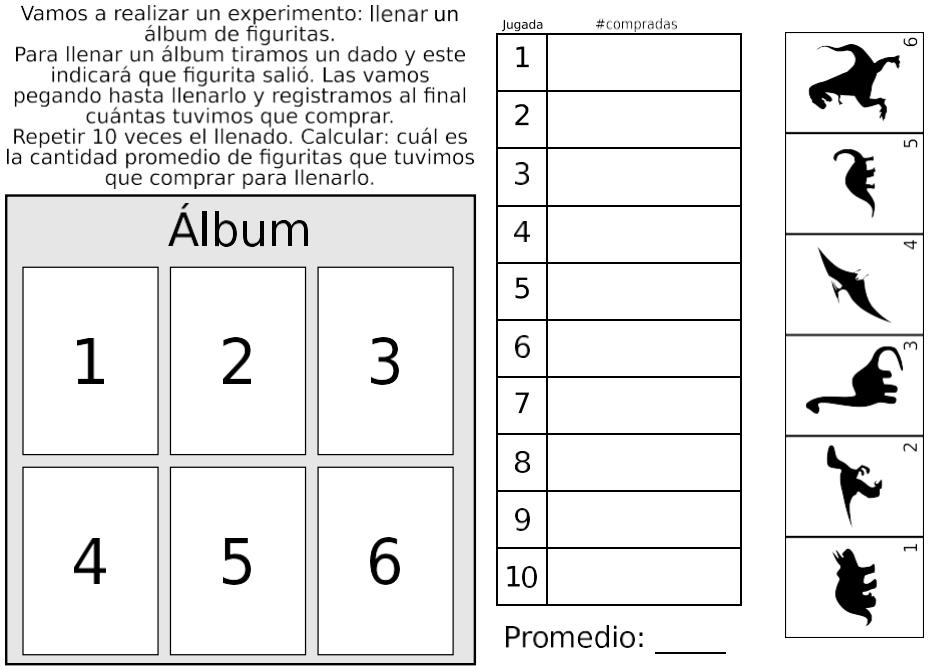

Figura 2. Álbum y figuritas usados en la parte lúdica de la actividad. Los alumnos pueden realizar la mecanica de llenar un álbum y entender los conceptos principales relacionados con ese proceso.

asistan al taller. Luego de estas cinco primeras ediciones, nos seguimos encontrando con un interés que parece ser insaciable, tanto de nuestra Facultad como de otras instituciones. Seguimos en el camino de lograr que este taller no sea sólo de mucho interés, sino que esté al alcance de alumnos de cualquier carrera, y que les sea de utilidad. Abajo detallamos algunos aspectos del taller que fuimos trabajando durante estas cinco ediciones para acercarnos a este objetivo.

Plantel docente multidisciplinario

El taller *Exactas programa* fue concebido por un plantel docente multidisciplinario, con el objetivo de brindar una herramienta poderosa a estudiantes de cualquier carrera de nuestra Facultad. Las actividades del taller están inspiradas en problemas de interés en estas diferentes disciplinas, pero siempre con el enfoque puesto en el desafío de la implementación computacional.

A lo largo de las distintas ediciones, hemos observado la necesidad de mantener esta diversidad en el plantel docente pero con un núcleo de docentes que permanezcan de forma estable durante los nueve encuentros. Los alumnos van encontrando sus referentes en las primeras clases, eligen con quiénes se entienden mejor preguntando y resolviendo dudas, y esos v´ınculos se sostienen y afianzan a lo largo del taller.

Dentro de esta visión multidisciplinaria y, al aumentar la cantidad de turnos en simultáneo que se fueron incorporando, se buscó mantener este espíritu y a la vez, asegurarse de que hubiera un referente técnico sólido en cada turno. Es normal que los alumnos planteen dudas o inquietudes que pueden exceder los conceptos mostrados en el taller o que ocurra algún error que pueda resultar complejo de solucionar. La presencia de un referente técnico con sólidos conocimientos en Python en cada turno probó ser más que suficiente para poder responder o solucionar cualquier inconveniente técnico que surgiera.

Actividades lúdicas

El hecho de que cada desafío empiece con una actividad lúdica y una puesta en común en pseudocódigo permite, por un lado, entender el problema y, por el otro, ayuda a entrenar el pensamiento computacional-algor´ıtmico antes de la implementación. Estas actividades lúdicas requieren de la identificación de qué objetos componen el modelo y una división de tareas necesariamente repetidas cierta cantidad de veces que permite descomponer la dinámica del sistema en sus reglas básicas.

Equilibrio entre creatividad y estructura

La presencia de docentes de diferentes disciplinas ha ayudado no sólo a identificar problemas de intereses diversos, sino también nos permitió ver la diversidad en posibles maneras de pensar y resolver los problemas que planteamos. Una decisión inicial que tuvimos que tomar era cuánto guiar o sesgar la resolución del

problema para no perder a los alumnos que no saben cómo empezar pero tampoco confundir a otros que hubiesen pensado o resuelto el problema de otra manera.

Para dar un ejemplo, retomamos la actividad del álbum de figuritas. En la parte lúdica, los alumnos rápidamente entienden las reglas básicas del juego: compramos una figurita y la ubicamos en el álbum, una y otra vez hasta completar el álbum entero. Observamos, sin embargo, una gran diversidad en las maneras en que los alumnos modelan esta dinámica, que se van ramificando en cuatro decisiones clave: cómo elegir aleatoreamente una figurita, cómo representar el álbum vacío, cómo ubicar la figurita en el álbum, y cómo definir la condición para seguir iterando o no.

En la primera edición de *Exactas Programa*, tanto en esta como en muchas de las otras guías intentamos no sesgar las soluciones de los alumnos, permitiéndoles creatividad total, dándoles algunos comandos útiles, como por ejemplo `random.random()` o `random.randint(a,b)` y una consigna del estilo "Simule el llenado de un álbum de seis figuritas". En base a devoluciones de alumnos, y con el objetivo de disminuir la tasa de deserción luego de esta actividad (la segunda del taller), empezamos a estructurar más la resolución del problema, agregando ejercicios del estilo "Defina una función que toma una lista `album` y un elemento `figu` como parámetros, y devuelve `True` o `False` si `figu` está o no en la lista `album`. De la misma manera, en la puesta en común luego de la actividad lúdica y antes de empezar a programar, discutimos diferentes maneras de abarcar el problema, y unificamos en un pseudocódigo en el pizarrón una forma de resolver el problema, explicitándo (por ejemplo, que el álbum vacío será una lista de ceros, y el álbum lleno una lista de unos). En cada edición de *Exactas Programa*, solemos ajustar ejercicios en las guías para seguir buscando este equilibrio entre mayor estructura (menor frustración) y menor estructura (mayor creatividad).

Escalabilidad y portabilidad

En su primera edición, *Exactas programa* era un taller dictado por docentes y doctorandos de nuestra facultad. Cada docente se encargó de elaborar, y luego dictar, una actividad relacionada a su disciplina. Debido al amplio interés de nuestra comunidad en asistir al taller, y gracias al apoyo de nuestra Facultad, hemos crecido rápidamente en tamaño, duplicando la cantidad de turnos en las dos ediciones siguientes. Durante esta etapa se hizo evidente la necesidad de una coordinación general del taller: se desarrollaron una página web y formulario de inscripción. Además, la asignación de cupos en los cuatro turnos, la distribución del plantel docente y la designación de un coordinador por turno ayudaron a desarrollar el taller satisfactoriamente.

Hoy en día disponemos de un repositorio con todo el material del curso, al cual todos los docentes tienen acceso. Esto permite unificar el material, como por ejemplo los cambios en los ejercicios y las diapositivas para introducir cada actividad. Para poder homogeneizar la resolución de los problemas y asegurar que un docente de cualquier área pueda guiar cualquiera de las actividades, nos beneficiamos mucho con la confección de una guía docente para cada actividad.

Buscamos ajustar el material del curso para continuar escalando la cantidad de cupos del taller, y también ayudar a su exportación a otras instituciones, como por ejemplo ha sucedido a la Universidad Nacional de San Martín en el verano del 2020.

Devoluciones desde diversas perspectivas

Consideramos que un taller con audiencia y docentes de áreas tan diferentes requiere de una búsqueda casi continua de devoluciones. Al término de cada encuentro los asistentes deben completar un formulario que nos permite conocer sus impresiones sobre la actividad propuesta; de modo de obtener realimentación en aspectos como cuánto pudieron terminar de la actividad en el tiempo de la clase, cuán difícil les pareció, etc. También cada alumno debe enviar el código que produjo durante la clase. Este material permite estudiar los diferentes recorridos en el proceso de aprendizaje de cada participante, y ayuda al docente construir un cierre de la actividad para la clase siguiente. En este mismo sentido, siempre se reserva un tiempo del último encuentro para discutir personalmente con los alumnos sobre sugerencias e impresiones generales del taller. Enviamos además una encuesta final a todos los asistentes persiguiendo el mismo objetivo.

Por otro lado, nos hemos beneficiado de una reunión docente luego de la finalización del taller, para discutir resultados generales e impresiones de los diversos turnos, así como ideas para mejorar el curso. Cada docente también completa formularios para proponer cambios específicos a las actividades del taller. Como resultado de la incorporación de nuevos docentes al plantel, hemos confeccionado tres guías básicas donde resumimos los principales conceptos abordados en las dos primeras actividades y proponemos algunos ejercicios relacionados a los mismos.

5. Conclusiones

Exactas programa es una experiencia sensorial, presencial. Hay música, teatralización, tortas que traen docentes y alumnos. Ponemos en práctica leyes de convivencia fundamentales promovidas, entre otros, por Greg Wilson (ver sección Rules de su libro [2] https://teachtogether.tech/#). De esta manera generamos un ambiente donde la empatía resulta ser la herramienta fundamental para compartir y generar conocimiento.

Generamos vías de comunicación (un grupo de Whatsapp por turno) donde los asistentes plantean sus dudas y, generalmente, un compañero consigue dar respuesta. Los docentes también se involucran en este terreno. Este espacio es importante tanto por lo dinámico que resulta para asistir a los alumnos ante pequeñas complicaciones, como por la red de colaboración que se establece entre los participantes, incluyendo a los docentes.

Por último queremos mencionar que a lo largo de las diferentes ediciones hemos conseguido disminuir la deserción, que se producía principalmente después del primer o segundo encuentro. Hemos tomado varias medidas: reformulamos

las clases, intensificamos el acompañamiento de manera casi insistente, confeccionamos guías ad-hoc resumiendo los principales conceptos que se abordan en estos encuentros. No hemos estudiado el impacto de cada una de estas medidas, pero en conjunto han resultado efectivas para contener a los alumnos.

6. Reflexión final

Durante los últimos 10 años, se produjo una explosión en el campo de la Educación Computacional: trabajos publicados, doctorados y, en algunos casos, muchos recursos financieros acompañando diferentes iniciativas. Pese a ello, los resultados no han sido los esperados y los especialistas en el área han estado reflexionando críticamente al respecto, promoviendo nuevas prácticas. Nuestra propuesta inicial, concebida por un plantel docente con muchos años de experiencia trabajando en nuestra Facultad pero poca formación teórica en el campo, contempla muchas de ellas: (1) integrar las computación con otras disciplinas (no sólo matemática, como ocurre habitualmente), presentando un amplio abanico de problemas a resolver; (2) proponer actividades que requieren una participación activa por parte de todos los asistentes; (3) no buscar eficiencia en las resoluciones, los docentes escuchan y acompañan las trayectorias que proponen los alumnos procurando apenas esclarecer errores conceptuales graves, (4) promover el pensamiento computacional, mas allá de la implementación, a la hora de pensar en nuevos desafíos.

Esperamos que nuestra experiencia pueda ser de ayuda para quienes quieran promover este tipo de iniciativas.

Referencias

1. Consejo Federal de Educación: Resolución Nro. 263/15 (2015), https://cfe.educacion.gob.ar/resoluciones/res15/263-15_01.pdf, visitada el 28 de agosto de 2020
2. Wilson, G.: Teaching Tech Together: How to Make Your Lessons Work and Build a Teaching Community around Them. CRC Press (2019)

A. Material Suplementario

Incendio Forestal

En esta actividad se propone simular la dinámica de un incendio forestal, modelando un bosque representado por diferentes etapas (brotes de árboles, caída de rayos, propagación del fuego, limpieza de árboles quemados).

El bosque propuesto es lineal, con N celdas o lugares. En cada celda puede haber un árbol o no, y en caso de que haya un árbol, podrá estar quemado o no. Vamos a decir que dos árboles son *vecinos* cuando están uno al lado del otro, en dos celdas consecutivas. Las condiciones de borde de nuestro bosque son abiertas, tanto la primera celda como la última tendrán un único vecino,

mientras que todas las celdas del medio tendrán siempre dos vecinos (uno a la derecha y otro a la izquierda).

El estado del bosque se representa en cuatro etapas:

- Época de brotes: Cada celda vacía tiene probabilidad p de que le brote un árbol.
- Época de caída de rayos: con probabilidad f caen rayos en cada celda y si había un árbol donde cayó un rayo, este árbol se quema (por lo que el bosque resultante tendrá lugares vacíos, árboles vivos y árboles quemados).
- Época de incendios: se propaga el incendio todo lo posible. Cada árbol quemado propaga el fuego a los árboles vecinos vivos inmediatos (el de la derecha y el de la izquierda). Nota: la propagación termina cuando no queda ningún árbol quemado que pueda propagar el fuego.
- Época de limpieza: se tiran abajo los árboles quemados y esas celdas pasan a estar vacías nuevamente.

Al cabo de las cuatro etapas, comienza nuevamente tomando al estado final del bosque como la condición inicial para la primer etapa siguiente.

Las preguntas que se intentan responder en esta actividad son: ¿Qué fracción de árboles resultan quemados para ciertos valores de los parámetros p (probabilidad de brotes) y f (probabilidad de caída de rayos)? Además, para una probabilidad fija f de caída de rayos, ¿cuál es el valor de p que maximiza la producción del bosque?

Para el momento lúdico, se les reparte a cada estudiante tres triángulos de cartulina: uno rojo (para representar un árbol quemado), uno verde (para representar un árbol sano) y uno blanco (para representar una celda vacía). Para la conformación del bosque, estudiantes y docentes participamos formando un bosque longitudinal, en el que todos quedamos conectados. Cada persona va a representar una celda del bosque. El bosque es abierto (no se cierra sobre sí mismo), es decir que tendrá dos bordes, como el que se implementa en la computadora.

Inicialmente, cada persona (que representa una celda) genera un número al azar en Python, para decidir si en su lugar va a brotar o no un árbol. En cada celda va a nacer un árbol con probabilidad $p = 0{,}8$. Las personas que representan celdas donde nació un árbol (obtener un número menor a 0,8) levantan la cartulina verde. Después de este paso habrá algunas personas levantando cartulinas verdes y otras levantando cartulinas blancas (los que representan celdas vacías). Luego, representamos la caída de rayos. Aquellos que representan celdas en las que hay árboles (los que tienen levantada la cartulina verde) vuelven a generar un número aleatorio. Si el número es menor que $f = 0{,}2$ entonces pasan a ser árboles quemados y cambian la cartulina levantada verde por una roja (representando la situación de árbol que se quemó). Los árboles que no se quemaron siguen con la cartulina verde levantada. Después de esta instancia va a haber algunas personas levantando cartulinas verdes, y algunas, rojas, y otros blancas (las celdas vacías).

Luego, representamos la propagación del fuego. Si un árbol sano tiene un vecino quemado, entonces se quema. Por el contrario, si un árbol quemado tiene

como vecinos celdas vacías se detiene la propagación en ese lugar. En este punto queda en evidencia la importancia de las celdas vacías que actúan como cortafuegos. El docente a cargo de coordinar el juego, ayudar a verificar si se terminó o no la propagación. Después de esta instancia, habrá personas con una cartulina verde levantada, otros con una roja y otros con una blanca (las celdas vacías). Luego, al llegar el verano, los árboles quemados se caen (los que tenían cartulina roja la bajan) y pasan a ser celdas vacías. Al llegar la primavera, en una nueva época de brotes las celdas vacías pueden dar lugar a árboles con probabilidad $p = 0,8$. En aquellas celdas que deban brotar árboles levantan la cartulina verde y se suman a las cartulinas verdes del ciclo anterior (evidenciando el estado final del bosque en el ciclo anterior como el bosque inicial del siguiente).

Es interesante que en el momento lúdico se parte de una versión inicial donde cada estudiante representa una celda que va tomando las decisiones por si sola de manera independiente, a una versión centralizada coordinada por un docente que se comporta como el orquestador de todos los movimientos. Este pasaje de sistemas autónomos que realizan acciones a un sistema centralizado que sea programable muestra que escribir el guión de este coordinador alcanza para representar una dinámica compleja.

En esta actividad se busca afianzar el manejo de *listas*, acceder a sus elementos y modificarlos, en base al valor de otros elementos de la lista (los vecinos inmediatos), generar números al azar, para poder simular la caída de los rayos. También se trabaja con la definición de *funciones*, implementando condiciones y ciclos. En esta guía por primera vez se grafican puntos y vs x usando la librería matplotlib.

Durante la implementación, los alumnos tienen que pensar cómo podemos usar la computadora para representar la siguiente situación: un evento ocurre con probabilidad p. Se discute acerca de números pseudoaleatorios y cómo operar con un número generado por la computadora entre 0 y 1 para resolver el problema. Finalmente, un desafío es pensar un algoritmo que reproduzca la dinámica que se quiere simular (en particular, la propagación del fuego en un bosque longitudinal).

Órbita Planetaria

Esta actividad presenta una fracción del sistema solar, cuya simplificación consta de dos cuerpos: el Sol, que se lo toma como centro de referencia y la tierra que orbita a su alrededor. El objetivo de la actividad es poder calcular esta trayectoria. Para calcular esta órbita se utiliza el algoritmo de Verlet[2]. Este algoritmo utiliza la posición actual y la posición en un instante anterior para calcular la próxima. Los datos iniciales para realizar la simulación son obtenidos de la página de la NASA. Este modelo incluye distintos conceptos como: discretización del tiempo, y descomposición en dos coordenadas de objetos en un plano.

[2] El algoritmo de Verlet es un procedimiento para la integración numérica de ecuaciones diferenciales ordinarias de segundo orden con valores iniciales conocidos

El objetivo es lograr calcular la órbita terrestre de manera de poder encontrar 3 puntos en dicha trayectoria que determinen días especiales partiendo del día de hoy:

- el próximo perihelio (las distancia mínima entre la Tierra al Sol),
- el próximo afelio (las distancia máxima entre la Tierra al Sol),
- el próximo cumpleaños de cada estudiante

Como herramienta pedagógica se realiza Verlet en un tiro vertical, donde se va realizando una lista de las posiciones que va teniendo un objeto arrojado hacia arriba. Se busca hacer foco en qué ítems de la lista de valores ya calculados se utilizan para el próximo, haciendo explícito el cuerpo del ciclo que luego tendrán que programar.

En esta actividad los estudiantes deben ejercitar listas donde se al almacenan los valores correspondientes a las dos dimensiones en las cuales se mueve la tierra (x e y). La novedad con respecto a las actividades previas es que las posiciones de la lista que se consultará en los próximos pasos aun no se ha creado. Es importante destacar que deben comprender el cambio de semántica de cada posición que en una iteración se considera futura, en la siguiente se considera la presente y en las siguientes ya son posiciones históricas.

Avalancha

Esta actividad consiste en modelar un terreno al que le va cayendo nieve y al acumularse cierta cantidad, se desmorona a algunas celdas adyacentes. El terreno es representado como una grilla (o matriz), donde se va a ir arrojando un copo de nieve a la vez. Cuando una posición llega a tener 4 copos de nieve se desborda trasladando un copo a cada uno de sus vecinos (celdas que comparten un borde en la grilla). De este modo, la celda que se desborda queda con cero copos de nieve.

Para ejemplificar cómo es la dinámica, la propuesta es armar una matriz humana de 4×4 donde cada estudiante que participa está ubicado en una grilla y tiene identificado quienes son sus vecinos. El docente a cargo de presentar la actividad va “tirándole” copos de nieve (representados con caramelos de menta) a una posición determinada. La persona que está en dicha posición acumula esos copos de nieve. Cuando supera la cantidad (sumando cuatro copos o caramelos) le desborda uno a cada vecino y se queda sin ninguno. Así mismo, antes de tirar un nuevo copo, se deben realizar todos los desbordes que sean necesarios hasta que no exista posición en la grilla que supere el límite. Cabe aclarar que las casillas de los bordes tienen menos vecinos, así que cuando desbordan los copos que corresponderían a vecinos que no existen, desaparecen.

En esta actividad los estudiantes deben utilizar por primera vez una matriz, aprender a recorrerla, y para cada posición poder construir las coordenadas que representan sus adyacentes.

Predador Presa

El modelo de predador-presa representa interacción biológica en la que un individuo de una especie animal (el predador) caza a un individuo de otra especie (la presa) para subsistir. En nuestro caso se instancia en Leones (como predador) y Antílopes (como presa) en un terreno representado por una grilla. Al igual que la actividad de Incendio Forestal (ver A.1) la dinámica tiene varias etapas que componen un ciclo que se repite tantas veces como el estudiante quiera. En este caso, son tres etapas que se describen a continuación:

- **Alimentación**: si un León tiene un Antílope vecino se lo come.
- **Reproducción**: si dos animales de la misma especie son vecinos, se reproducen dando lugar a un nuevo individuo.
- **Movimiento**: todos los individuos se desplazan.

Para resolver estas etapas se utiliza la lógica de recorrer una grilla vista en Avalancha (ver A.3) y esto trae como consecuencia que el tablero se va dividiendo en dos: la zona ya resuelta de dicha etapa, y la no resuelta. Se suceden casos en los que un individuo, al ser desplazado, se ubica en la zona no resuelta de la grilla pudiendo ejecutar una acción varias veces en un mismo ciclo. Esta situación antiintuitiva funciona como desafío al finalizar este modelo para realizar un modelo más complejo que resulte más realista.

Gases

En esta actividad se modela un sistema de partículas que interactúan entre sí y están contenidas en una caja. Por cuestiones de simplicidad y de mejor visualización se trabaja en dos dimensiones. Se incorpora la necesidad de escribir los resultados a un archivo para ser visualizados por un programa externo. Si bien es un modelo bastante simple, es muy preciso y permite modelar procesos reales muy complejos cómo cambios de fase.

Por ser la última actividad, el pautado del desarrollo es algo menor permitiendo una mayor variedad en el código. En primer lugar, se realiza un código de una partícula que se mueve limitada por una caja. La trayectoria se calcula mediante un algoritmo de Verlet casi igual al que se implementó en la práctica de órbita planetaria. El desafío es mantener la partícula dentro de una caja modelando choques elásticos con las paredes.

En segundo lugar, se incluyen n partículas contenidas en la misma caja pero que no interactúan entre sí. Luego se incorpora una interacción repulsiva y de corto alcance entre las partículas lo que habilita a que choquen entre sí e intercambien energía. El esquema del código se puede resumir en el diagrama de la figura 3.

El desarrollo de la simulación necesita de múltiples ciclos anidados, ya sea mediante `for` o `while`, que si bien ya fueron usados en casi todas las prácticas, requieren de un nivel de abstracción mayor: por ejemplo la fuerza sobre cada partícula se calcula por la suma de las interacciones con todas las otras, lo que requiere de dos ciclos anidados.

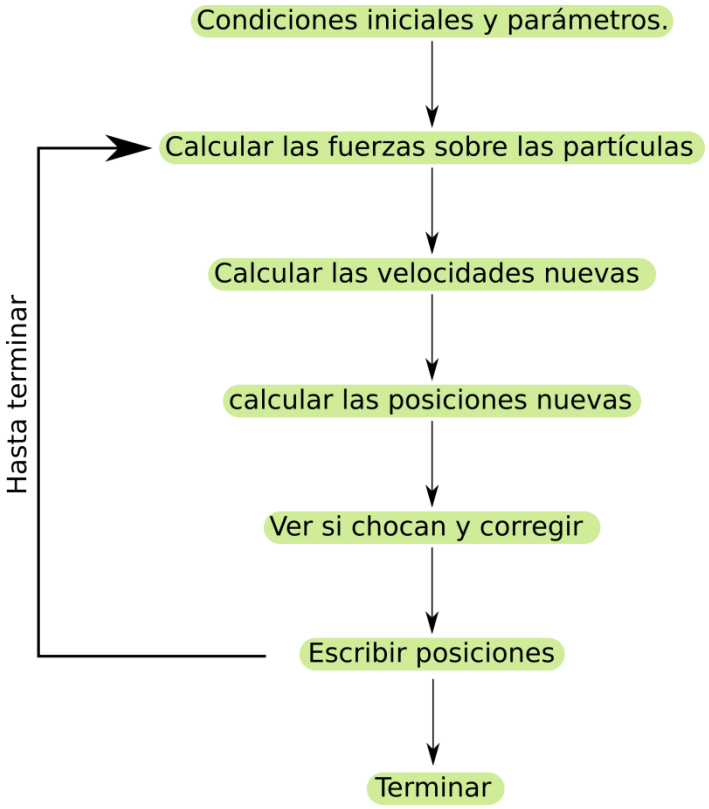

Figura 3. Esquema de la logica empleada en la realización del código de modelado

La mayor novedad es la escritura en un archivo con un formato específico que pueda ser leído por un programa de visualización de dinámicas moleculares llamado VMD [3]. Este programa es muy utilizado por la comunidad cient´ıfica, pero se podría usar otro.

Como broche final y para verificar el código generado, se calcula la energía cinética y la energía potencial en cada paso de simulación y se comprueba que la energ´ıa total se conserva.

Si bien en este punto la actividad se considera completa, existen una variedad de extensiones que alumnos motivados pueden usar para seguir jugando/aprendiendo.

Algunas extensiones son:

- Agregar una interacción atractiva.
- Hacer dinámicas manteniendo la temperatura (energía cinética) constante.
- Calcular la distribución de velocidades y comprobar que se corresponden con los predicho por Maxwell-Boltzman mucho antes de la existencia de las computadoras.
- Incluir tipos de partículas diferentes con interacciones distintas según el tipo.
- Pasar de partículas monatómicas a diatómicas (unidas por un potencial armónico). s

Esta actividad, al igual que otras planteadas en este talle, tiene diferentes posibles etapas de culminación permitiendo que estudiantes inquietos puedan desarrollar su capacidad e interés realizando ejercicios optativos de mayor complejidad.

[3] https://www.ks.uiuc.edu/Research/vmd/