

Misceláneas

De Exactas Programa a Pensamiento Computacional Una historia compleja con final feliz escrita a cuatro manos

Ignacio Ojea* y Mariela Sued†

*Dep. Matemática - FCEN y Ciclo Básico Común, UBA; IMAS (CONICET-UBA)

† Universidad de San Andrés - CONICET



Prólogo (Esteban Mocskos)

Resulta un tanto extraño salirse de la escritura de los textos científicos habituales para escribir una reflexión acerca de un proceso que, visto desde la perspectiva actual, podemos considerar un éxito en varios aspectos. Vamos a contarles cómo la necesidad de avanzar en el dictado de contenidos computacionales, impulsó a un grupo de docentes de la Facultad de Ciencias Exactas y Naturales de la UBA a diseñar, dictar, mejorar y disfrutar un curso que enseña los elementos iniciales de programación. En esta descripción, a veces teñida de un cierto romanticismo alentado por el paso del tiempo que suaviza los tropiezos, dos protagonistas de esta experiencia, Ignacio y Mariela, nos brindan sus perspectivas.

Esperamos que compartir esta historia impulse a otros colegas a concebir experiencias similares, atendiendo a las demandas específicas de sus entornos. Estamos a disposición para colaborar con nuevas iniciativas en este sentido, compartiendo nuestra experiencia y el material elaborado, como venimos haciéndolo.

Mariela Sued: ¿Cómo comenzó todo esto?

El histograma es una representación gráfica que nos permite visualizar de qué manera se distribuye un conjunto de datos numéricos. El *estimador no paramétrico de la densidad* puede ser considerado una versión suavizada del histograma. Junto con regresión no

paramétrica, este es uno de los temas preferidos por muchos docentes amigos para contar: se observa fácil algo lindo y hay muchas maneras de presentarlo según la formación del auditorio. Además, se ve claramente el famoso compromiso sesgo-varianza y la matemática necesaria para analizar la consistencia del estimador es muy accesible. En este enlace se puede acceder a visualizaciones interactivas, incluidas en [este libro](#), de García – Portugués.

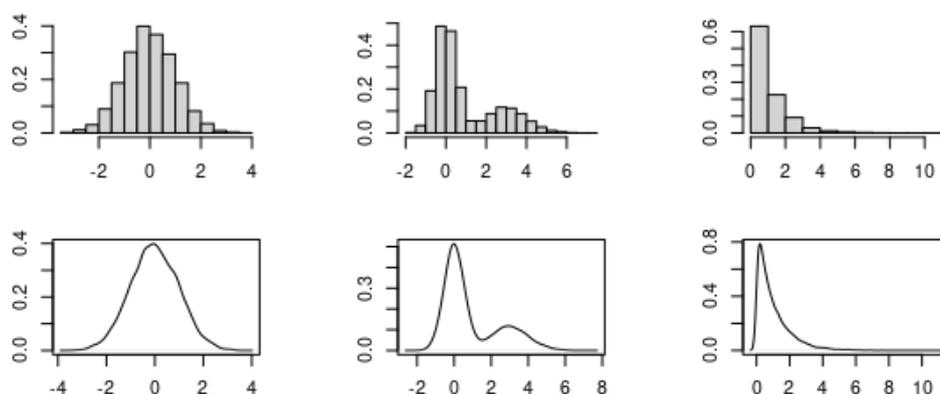


Figura 1: En la fila 1 se muestran histogramas de diferentes conjuntos de datos, y abajo de cada uno se grafica el estimador no paramétrico de la densidad.

Este es uno de los temas que presentamos en *Ciencia de Datos con R - Fundamentos Estadísticos*, una materia optativa en la Facultad de Ciencias Exactas y Naturales de la UBA. El auditorio suele ser muy variado: muchos estudiantes de Biología, algunos de Computación, pocos de Matemática, un par de Ciencias de la Atmósfera y algunos otros que no identifico de memoria. En la primera edición, terminé con mi presentación del estimador, mostré algunos gráficos, discutimos en algunos ejemplos qué pasaba con el tamaño de la ventana, qué pasaba con la elección del núcleo. Todos contentos. - ¿Se entendió?, pregunté. Algunas cabezas se movieron indicando que sí. El famoso resorte que los estudiantes tenemos en el cuello y se pone en acción cuando alguien pregunta si se entiende (me suena que esta es una idea de Paenza). Como estos muñequitos. Genial, fuimos entonces a las computadoras. Consigna. Caos. La cara de mi *biólogo favorito* cuando me preguntó qué había que hacer con h no me la olvido más. Fue un día **fundacional** en mi vida. Si bien en esta materia se pedía familiaridad con \mathbb{R} , nos dimos cuenta de que *familiaridad* es un término bastante poco específico. Una cosa es cargar datos, ejecutar algún comando y *más o menos* interpretar el resultado. Otra cosa es implementar una solución propia, escribir una función en la *compu*.

De paso, aprovecho para compartir un temita no menor. Una función en computación es algo muy diferente a lo que llamamos función en matemática. ¡Que desgracia! ¿No? Y eso mismo pasa con tantas otras cosas. ¿Es necesario definir qué es una probabilidad? ¿Quién no lo sabe? Y en eso, llegó Kolmogorov. Y nos pusimos a *definir* la probabilidad. Esto de apropiarnos de términos públicos y resignificarlos con definiciones complejas, es una verdadera desdicha. Algo que merece ser problematizado en el aula. La apropiación lingüística.

Volviendo a aquella clase: me di cuenta de que una cosa es creer entender y otra muy diferente es poder implementar. Entonces, cambié mi definición de entender. Parece ser que la frase *si lo puedes entender, lo puedes calcular* no es una cita textual de Richard Feynman, como alguna vez me dijeron. Pero refleja de manera inmejorable lo que pienso después

de aquella experiencia. La computadora como laboratorio para hacer, en particular en mi caso, estadística. Una nueva **herramienta didáctica** donde el estudiante se transforma en protagonista. Donde necesita **entender** para **implementar**; donde puede **experimentar**. Donde la **lógica** es implacable. Donde la **abstracción** juega un rol fundamental.

Mientras constataba que los estudiantes de la Facultad de Ciencias Exactas y Naturales no contaban con las habilidades necesarias para utilizar la computadora en la resolución de problemas, en el país se anunciaban, una vez más, grandiosas reformas educativas. Mucho se hablaba de la robótica y la programación. *Por primera vez llega la alfabetización digital al Nivel Inicial*, anunciaban las autoridades del momento. Era junio de 2019 y nosotros ya habíamos comenzado con Exactas Programa: un taller destinado a resolver problemas utilizando la computadora. Una oportunidad magnífica; una verdadera necesidad.

Mariela Sued: Surgimiento de Exactas Programa

Durante los almuerzos del primer cuatrimestre de 2017, en la mesa del viejo Instituto de Cálculo, nos preguntábamos cómo resolver este dilema: hacer que los chicos lleguen a las diferentes materias con las herramientas computacionales necesarias para poder resolver problemas interesantes y no enfrentarse a la necesidad de tener que enseñar a programar antes de poder dedicarse a los temas de la currícula. Entonces, salimos a buscar aliados, entendiendo que esta problemática era común a varias disciplinas de la Facultad.

“Mariela nos preguntó a varios si nos parecía bien que hubiera graduados de la Facultad que no tuvieran idea de programación. A los que decíamos que no, nos pedía que la ayudemos con la idea de armar un taller. Así se formó un grupo de docentes, investigadores, investigadoras y estudiantes de doctorado de distintos departamentos que cubrieron una buena parte de las disciplinas científicas.”

Este recorte forma parte de una nota publicada en El Cable, un semanario de difusión interno de la Facultad de Ciencias Exactas y Naturales, donde Esteban Mocskos sintetiza aquellos inicios de una manera insuperable (Esteban consigue siempre mejorar cualquier iniciativa).

En las primeras reuniones éramos muchos, todos con buenas intenciones pero con pocos acuerdos. Con el correr de las semanas, algunos fueron quedando en el camino. Hasta que, un buen día, decidimos ponernos en acción y nos repartimos tareas entre quienes seguíamos entusiasmados con el proyecto. Cada equipo quedó a cargo de diseñar una propuesta vinculada a su disciplina que incluyera un desafío computacional interesante. Así fue como el curso —que luego se llamaría **Exactas Programa**— empezó a tomar forma en torno a problemas concretos. Todas las clases empezaban con la formulación de un problema y una breve experimentación lúdica sin medios digitales. Después, sí: había que escribir código en la computadora. Las consignas ofrecían una guía general de qué funciones implementar para construir, de manera modular, una posible solución. Las primeras dos clases incluían un buen rato de exposición para presentar las herramientas elementales de programación imperativa: ciclos, condicionales y funciones. Pero fuera de eso la enorme mayoría del tiempo se destinaba a que cada alumno escribiera código en su computadora (sí, una compu por persona y cada una metiendo mano en su teclado).

Y... ¡funcionó! Gracias a Inés Caridi (IC), José Crespo (EGE), Juan Pablo Pinasco (DM), Esteban Mocskos (DC), Mariano Camilo González Lebrero (QI), Federico Sevlever (DF), Juan Pablo Di Bella (DF) y Mehrnoosh Arrar (IC) pudimos largar la primera edición de Exactas Programa, en el verano de 2018. Un éxito total.

Ignacio Ojea: Un docente más al equipo

Un día me crucé con Juan Pablo Pinasco en la cocinita del Departamento de Matemática. - Che, ¿sabés algo de Python?- me preguntó. - No - le contesté. - ¿Y no querés dar un curso?-. - Dale - le dije. Así es como me sumé como docente a Exactas Programa.

Había sido incontables veces JTP de Elementos de Cálculo Numérico, la materia en la que aspiramos a enseñarles a programar a estudiantes de Matemática y Física. Además, claro, de enseñarles métodos numéricos para ecuaciones diferenciales ordinarias, sistemas lineales, ecuaciones no lineales, interpolación, cuadrados mínimos, integración numérica, etc. También había estado muchas veces a cargo de la práctica de Análisis Numérico, una materia más avanzada, en la que había podido constatar el déficit de la formación estrictamente computacional que dejaba Cálculo Numérico.

Exactas Programa no era un curso de Python. Tampoco era una materia. Era la solución al problema con el que me venía encontrando recurrentemente: ¿cómo conseguir que los estudiantes incorporen la computadora como una herramienta para resolver problemas, para experimentar, para probar ideas, para ganar intuición? La definición que supimos conseguir dice que Exactas Programa es *un curso de resolución de problemas usando la computadora*. Allí se dicen un par de cosas: ostensiblemente se evita la expresión *curso de programación* mientras que se introduce en primer plano el elemento práctico: *resolución de problemas*.

En criollo: es un curso que *va a los bifés*, evita meterse en (por lo demás, interesantísimos) meandros teóricos y procura dar herramientas para que luego cada uno use la computadora como una aliada en su vida académica cotidiana. Sin embargo, la definición también esconde casi todos los elementos que hacen de Exactas Programa una auténtica joya. Cualquiera que haya creado, modificado, o simplemente dictado una materia dedicándole un poco de reflexión a cómo hacerlo de la mejor manera, sabe que entre la lista de contenidos mínimos y la realidad hay un abismo. Una materia no es una lista de temas, sino los temas que efectivamente se dan, el ángulo desde el que se los enfoca, los énfasis que se hacen, la manera en que se evalúa y, por supuesto, la impronta particular de cada docente. Exactas Programa tenía un verdadero regimiento de docentes (6 u 8 auxiliares cada 50 estudiantes) y los docentes proveníamos de todas las carreras de la Facultad. Cada clase presentaba un problema con una consigna que guiaba en el modelado general. Además, la actividad empezaba con un juego concreto, fuera de la computadora: con cartas, dados, lápiz y papel o armando filas físicamente. Esa etapa promovía en los estudiantes la localización de los elementos del problema y la secuencia de pasos fundamentales para resolverlo *antes* de empezar a programar.

Los problemas se relacionaban con todas las disciplinas científicas que se enseñan en la Facultad, de modo que había algo para el gusto de cada uno. Además, a diferencia de lo que ocurre con la enorme mayoría de los manuales introductorios a la programación, siempre se aspiraba a resolver o simular una situación verdaderamente interesante y no sólo un juguete didáctico. Cualquiera que elija al azar una clase de Exactas Programa, tome la consigna y la resuelva, sentirá esa euforia peculiar que da el haber resuelto un problema por uno mismo (euforia a la que todo estudiante de ciencias debería aspirar*).

*¿Sólo de ciencias? Quizá recuperar algo del espíritu lúdico en los estudiantes sea la reforma educativa más revolucionaria que se pueda imaginar y, quizá, no dependa tanto de lo que ocurre en el aula.

El equipo docente

Los docentes de Exactas Programa no éramos expertos, sino gente que programaba para resolver problemas dentro de sus disciplinas específicas. Como suele ocurrir, enseñando aprendimos muchísimo y ahora programamos mejor. Pero nuestro objetivo central no era que los estudiantes escribieran EL código, sino acompañarlos para que pudieran escribir *su* código. Por supuesto, hacíamos algunas correcciones de estilo y fomentábamos buenas prácticas elementales, pero siempre partíamos del código del estudiante, intentábamos tomar su idea y ayudarlos a implementarla. Si la lógica de su enfoque no servía o no era la más adecuada la experimentación era el mecanismo natural para hacerlo notar. En la medida de lo posible, debíamos guiar al estudiante para que se diese cuenta cuando estaba encarando el problema por un camino erróneo. Esto solo es posible con gran cantidad de docentes por alumno. También se apoya de manera decisiva en un elemento que puede parecer lateral: Exactas Programa era un curso optativo, sin evaluación. Los participantes estaban ahí porque querían y estaban dispuestos a poner cuatro horas de su día en ese juego. Esto es difícil de replicar de manera generalizada en materias obligatorias. Pero vale la pena intentarlo.

El entusiasmo de los estudiantes era una respuesta natural al entusiasmo y la dedicación de todo el equipo. Mariela, con su empuje arrollador, había logrado formar un grupo de profesores muy comprometido con la organización del curso (trabajaron ad-honorem desde el inicio). Ese compromiso inmediatamente se contagió a los auxiliares. Para la realización de las distintas ediciones de Exactas Programa se consiguieron algunos fondos: los auxiliares cobraban un cargo de ayudante por un mes. Pero en la mayoría de los casos, lo que hacían excedía largamente esa instancia de docencia eventual: pensaban actividades nuevas y les daban forma, sugerían ajustes en función de la experiencia, se hacían cargo de la coordinación de varios cursos, etc. Mucha gente se cansó o fue tomando otros compromisos que hicieron que ya no les fuera posible dedicarle tiempo a un proyecto hecho a pulmón, pero también se fue sumando gente nueva, con un gran impulso y siempre compartiendo el trato cordial y la dedicación al avance de cada estudiante.

Mariela Sued: Organizando la Fiesta

El jueves 6 de octubre de 2022 recibí un montón de mensajes de gente amiga, todos ellos en lugares estratégicos de la conducción de nuestra Facultad, consultándome sobre un proyecto que se discutiría al día siguiente en una reunión. La propuesta era tan buena que resultaba insuperable: incluir una materia de computación en el CBC. Ingeniería había conseguido ese milagro y Exactas tenía que resolver cómo adecuar su propio CBC. La materia tenía nombre y todo: **Pensamiento Computacional**. Sigo sin tener del todo claro quién elaboró el documento, pero me honraba encontrar ejemplos de ejercitación que habíamos preparado años antes para materias de probabilidad. *La ruina del jugador* estaba ahí. Es un ejemplo de cómo resolver una linda pregunta sin necesidad de plantear una solución analítica. En [este enlace](#) se encuentra el documento completo.

No participé de la reunión y, por lo que me contaron, fue dura. No se habían generado los consensos necesarios para poder avanzar en un plan de semejante envergadura. En lo personal, no sabía siquiera que fuera posible.

Entonces, me presentaron a Santiago Ceria, el abanderado de la idea. Una persona excepcional. Comprometida y sensible a la vez. Y desde ese momento, me sumé a la causa. Incluimos en el documento *Todas las voces, todas*, disponible en [este enlace](#), la visión de Santiago sobre esta historia.

Conseguimos un primer gran avance institucional. Después de una presentación a los directores de departamento (disponible en [este enlace](#)) organizada por las secretarías académicas (♥), se aprobó Pensamiento Computacional como materia del CBC para las carreras de Matemática, Computación y Ciencia de Datos. *Solo* restaba armar la materia.

El proceso fue arduo. Muchas opiniones, muchas críticas a lo que veníamos haciendo en Exactas Programa. Pero pocas ideas concretas para llevar al aula.

Una matemática (como yo) formando parte de la diagramación de una materia de pensamiento computacional para el CBC resultó ser una idea más disruptiva de lo que muchos estábamos preparados para tolerar. Por suerte, la casa está en orden y los expertos siguen custodiando la disciplina. En lo personal, no fue fácil sobrevivir a tanto hostigamiento. Muchas críticas pero pocas propuestas. Eso sí: en el proceso, todos aprendimos mucho. En particular, aprendimos a distinguir entre el *experto* y el *usuario* de un saber.

Exactas Programa nos enseñó a ser más flexibles, o al menos, a intentarlo. Ver sufrir a los expertos en computación ante el uso que los simples usuarios hacemos de la máquina nos obligó a repensar profundamente el rol docente. La importancia de sacrificar el rigor en favor de la claridad — como sugiere Greg Wilson en Teaching Tech Together — resultó más evidente que nunca.

Cuesta entender que otros pueden ser apenas usuarios, y que nos necesitan justamente para poder desempeñar bien ese rol. Los expertos en una disciplina formamos una cofradía que, una vez dentro, olvida el arduo camino recorrido para pertenecer. Naturalizamos lo que antes nos costó tanto y pasamos, sin darnos cuenta, de víctimas a victimarios.

Todo esto, en el mejor de los casos. Porque este es el recorrido de quienes sobrevivimos y logramos pertenecer. En el camino, muchos quedan atrás.

Cerrado este espacio catártico, volvamos a lo nuestro. Intentamos consensuar. Satisfacer a todos los que estaban interesados en participar. En lo personal, no solo no lo conseguí. Tomé distancia prudencial. Al menos Perla y sus maravillosos desarrollos, como [este](#), ya formaba parte de todo esto. Y en eso, Ignacio y Esteban, con un tremendo equipo docente que veníamos cultivando, hicieron magia. El primer cuatrimestre de 2024 se largó Pensamiento Computacional.

Ignacio Ojea: Pensamiento Computacional

La Facultad y el CBC (son unidades académicas distintas) habían aprobado la materia. Exactas hacía un aporte significativo: la mitad de los cargos docentes y el espacio físico (los laboratorios). Sin embargo, tratándose de una materia del Ciclo Básico Común era necesario que hubiera un cuerpo docente dentro del CBC que la llevara adelante. Ingresé como docente al CBC en 2006, e hice todo el camino desde ayudante de segunda a profesor dentro del área de Matemática, históricamente asociada a Exactas. Además, era también profe en Exactas y había participado de Exactas Programa: tenía todos los números.

Empezamos a pensar la materia con Esteban Mocskos, socio fundador de Exactas Programa y Christian Cossio, ambos con cargo en el Departamento de Computación de Exactas, cuya carga docente cubren en este curso. Para el armado general (redacción de consignas, confección de un cronograma, pensar la modalidad de evaluación, etc.) contamos con un equipo de preparación modestamente rentado formado por docentes que venían de Exactas Programa.

Hacer una materia a partir de un curso optativo presenta varios desafíos. Pensamiento Computacional tiene una clase *teórica* (de dos horas, en un aula común) y una clase

de cuatro horas de laboratorio[†]. La idea es que las clases teóricas anticipen algunas de las nociones básicas para el próximo laboratorio y consoliden lo visto en el anterior, aumentando así el tiempo de trabajo neto en la máquina. En el laboratorio la dinámica sigue el modelo de Exactas Programa, pero ya no tenemos 6 auxiliares cada 50 alumnos, sino solo 2.

Tuvimos que pensar cómo evaluar. Para mantener el foco en la resolución de problemas, decidimos tomar un trabajo práctico (en parejas) que consiste en resolver una consigna similar a las de las clases, pero de manera más autónoma. Es importante, sin embargo, contar con una nota individual. Optamos, por lo tanto, por tomar también un parcial escrito. consignas,

Pensamiento Computacional aún está en período de prueba y estamos haciendo cambios en función de la experiencia del último año y medio. Por el momento, tiene solo cuatro comisiones de 50 estudiantes cada una. Eso es menos del diez por ciento de la población de potenciales estudiantes (los inscriptos al CBC de Computación, Matemática y Ciencia de Datos son algo más de tres mil). Para que la materia crezca deberán sortearse muchas dificultades: limitaciones de infraestructura, cargos docentes, etc. Pero al menos, por ahora, contamos con algunos recursos invaluable: el enfoque y, sobre todo, los docentes salidos de Exactas Programa.

Exactas Programa es el corazón de Pensamiento Computacional. Sin Exactas Programa, Pensamiento Computacional no sería posible y si la hubiesen inventado, sería completamente distinta.

Muchas consignas de Exactas Programas se trasladaron casi intactas a la nueva materia y otras las escribimos siguiendo el mismo enfoque. Los objetivos centrales siguen siendo esencialmente los mismos: enseñar a resolver problemas usando la computadora. Además, todo el equipo inicial de auxiliares de Pensamiento Computacional provino de Exactas Programa. Eso le da a la materia, de manera natural, una consistencia que usualmente es muy difícil de lograr. También resuelve por sí sólo el problema del experto mencionado más arriba: los auxiliares vienen de distintas disciplinas científicas (la mayoría, de Biología).

Nuestros docentes mantienen un entusiasmo poco común. Permanentemente están buscando cómo mejorar la cursada, observando qué trabajos prácticos andan mejor y cuáles no tanto, ajustando las consignas, trayendo nuevas propuestas, etc. Un par de veces nos juntamos a cenar y discutimos durante horas cómo hacer que la materia sea mejor. Este compromiso excepcional se observa en docentes que son estudiantes, graduados e incluso doctores, que participan con un modestísimo cargo de ayudante de primera simple, cuyo salario actual es realmente vergonzoso. Además, permanece aún cuando haya recambios: las nuevas incorporaciones se insertan en un equipo de gente muy comprometida y que funciona muy bien. El empuje de Mariela perdura. Ojalá logremos que Pensamiento Computacional siga formando docentes tan bien como lo hizo Exactas Programa y que iniciativas como estas puedan propagarse a nuevos espacios.

 [Ir al índice general](#)

[†] Este formato requirió de importantes esfuerzos por parte del personal del CBC encargado de distribuir a los estudiantes en las comisiones, dado que difiere de la distribución horaria del resto de las materias y no es fácil de insertar en el organigrama preexistente.