Lecturas de Cátedra

Pensamiento lógico y programación en Python

Martín Goin y Edith Lovos

EDITORIAL UNRN

Pensamiento lógico y programación en Python

Lecturas de Cátedra

Pensamiento lógico y programación en Python

Martín Goin y Edith Lovos

EDITORIAL UNRN

Índice

Agradecimientos 9
Prólogo 11

Capítulo 1. Introducción a la programación
1. 1. Hitos de la programación 15
1. 2. ¿Qué es programar? 16
1. 3. ¿Por qué aprender a programar? 17
1. 4. Algoritmo. Un primer acercamiento 18
1. 5. Cómo diseñar algoritmos 19
1. 6. Del algoritmo al programa 19
1. 7. Python: historia y características 20

Capítulo 2. Diagramación lógica y pseudocódigo
2. 1. Diagramación lógica 25
2. 2. ¿Qué son las variables? 27
2. 3. ¿Qué son las constantes? 30
2. 4. Estructuras de control 30
2. 5. Operadores relacionales 36
2. 6. Operadores lógicos 36
2. 7. Otros operadores matemáticos especiales 40
2. 8. Pseudocódigo 43
2. 9. Guía rápida del diagrama lógico a pseudocódigo 48
2. 10. Problemas: estructura de control de decisión 49

Capítulo 3. Estructuras de control de repetición e iteración
3. 1. Estructuras de control de repetición e iteración 53
3. 2. Estructura de control de repetición: sentencia «mientras» 55
3. 3. Variables contadoras y acumuladoras 57
3. 4. Problemas: estructura de control de repetición «mientras» 60
3. 5. Estructura de control de repetición e iteración: sentencia «para» 61
3. 6. Traza o prueba escritorio 66
3. 7. Máximos y mínimos 69
3. 8. Validación de datos de entrada 72
3. 9. Problemas: estructura de control de iteración «para» 73

Capítulo 4. Codificación
4. 1. Código Python 77
4. 2. Tipos de errores: sintácticos y semánticos 80
4. 3. Pasaje de pseudocódigo a código 81
4. 4. Tipos de datos numéricos 82

4. 5. Comentarios en Python 83
4. 6. Estructura de control de decisión «if», «else» y «elif» 83
4. 7. Problemas: estructura de control de decisión con Python 87
4. 8. Estructuras de control de repetición «for» 88
4. 9. Librerías 90
4. 10. Máximos y mínimos 92
4. 11. Estructura de control de repetición «while» 92
4. 12. Números aleatorios 96
4. 13. Problemas: estructura de control «for» y «while» 99

Capítulo 5. Arreglos: vectores y matrices
5. 1. Arreglos 103
5. 2. Vectores 103
5. 3. Problemas: arreglos y vectores 118
5. 4. Matrices 120
5. 5. Asignaciones condicionales 133
5. 6. Atributos de un arreglo 134
5. 7. Gráficas: Matplotlib 134
5. 8. Problemas: arreglos y matrices 139

Capítulo 6. Modularización: funciones y procedimientos
6. 1. Modularización 141
6. 2. Funciones 142
6. 3. Procedimientos 150
6. 4. Variables locales y globales 153
6. 5. Palabras reservadas 156
6. 6. Problemas 157

Autorías y colaboraciones 160

Agradecimientos

El libro no habría sido posible sin el apoyo de nuestra querida familia que acompaña en todo momento con paciencia en este largo camino.

Agradecemos a las y los docentes con quienes compartimos la tarea y el deseo de superarnos día a día, no solo en la enseñanza, sino también en la constante búsqueda de nuevas estrategias que nos permitan enriquecer nuestras prácticas didácticas.

También agradecemos a las y los estudiantes que nos recuerdan que enseñar también es aprender.

Prólogo

Martín Goin y Edith Lovos

En un mundo cada vez más digital, en el que la tecnología está en constante crecimiento y transformación, las habilidades de programación se han convertido en una práctica esencial. Muchas de las tecnologías digitales (productos y servicios) con las que interactuamos están naturalizadas o, en algunos casos, son imprescindibles en nuestra vida cotidiana.

El aprendizaje de la programación es un proceso en el que se ponen en juego habilidades técnicas y, también, otras consideradas *blandas*, como el trabajo en equipo, el autoaprendizaje, la capacidad de aprender del error, entre otras. A esto se suma que aprender a programar es una forma de desarrollar el pensamiento creativo que involucra la resolución de problemas usando algoritmos.

Este libro no está dirigido solo a quienes aspiran a convertirse en expertos desarrolladores, sino también para cualquier persona que desea entender cómo las aplicaciones de *software*, que utilizamos todos los días, cobran vida. Está diseñado como una guía que, paso a paso y progresivamente, permite comprender los conceptos fundamentales de la programación a través de Python, un lenguaje de uso muy extendido en el mundo profesional del desarrollo de *software*. Este surge como uno de los favoritos para la enseñanza y el aprendizaje, principalmente, porque es fácil de entender, versátil y, al mismo tiempo, poderoso.

Avanzar en este libro no solo permitirá el aprendizaje de la sintaxis y la semántica de Python a través de los diferentes ejemplos que se presentan, sino también el desarrollo del pensamiento lógico y creativo con la intención de resolver problemas. De este modo, se derrumba la idea de que la programación es compleja de aprender y de usar para la realización de ideas poniendo en evidencia que es accesible a cualquier persona, independientemente de su nivel de experiencia.

El libro funciona como un tutorial destinado a los primeros cursos de grado de carreras como licenciatura en Sistemas, licenciatura en Informática, Ingeniería Ambiental u otras que incluyen asignaturas vinculadas a la programación de computadoras. Está pensado para ayudar a desarrollar las habilidades necesarias para la apropiación del lenguaje Python como herramienta para codificar algoritmos. Para esto, están disponibles 65 ejemplos prácticos resueltos y 91 problemas estratégicamente pensados como desafíos que fortalecen la capacidad para resolverlos. Cada capítulo está diseñado con el fin de contribuir con el

anterior, de esta manera se asegura una comprensión sólida y accesible debido a las explicaciones detalladas y una batería de problemas que ayudan a aplicar lo aprendido.

El libro se divide en seis capítulos. Salvo el primero que expone los conceptos básicos de la programación y el lenguaje, el resto atiende y pondera aspectos prácticos para desarrollar algoritmos. Entonces, el capítulo 1, «Introducción a la programación», expone los hitos de la programación, define el concepto y la necesidad de su aprendizaje. Además, ofrece un acercamiento a los algoritmos, su diseño, el pasaje del algoritmo al programa, e incluye la historia y las características de Python.

Luego, el capítulo 2, que se titula «Diagramación lógica y pseudocódigo», profundiza en los siguientes temas: la diagramación lógica, las variables y constantes, el intercambio de variables, las estructuras de control (decisión simple, doble, anidadas, independientes), los operadores lógicos y relacionales, así como los operadores matemáticos especiales. Por último, se explica el pasaje del diagrama lógico al pseudocódigo.

En tanto, el capítulo 3, «Estructuras de control de repetición e iteración», se enfoca en las estructuras de repetición «mientras» y «para». También se abordan las variables contadoras y sumadoras (acumuladoras), la traza o prueba escritorio, los máximos y mínimos, las estructuras de decisión combinadas con bucles de repetición y la validación de datos de entrada.

La instalación y entorno Python se expone en el capítulo 4, que se titula «Codificación». Se explican temas como el modo *prompt* y la creación de archivo fuente .py, además de los errores sintácticos y semánticos que se pueden presentar. Asimismo, se enseña a convertir pseudocódigo a código, a utilizar las estructuras de decisión «if», «else», «elif», así como las de repetición «for», «while» y a encontrar los máximos y mínimos. Se detallan también los tipos de datos que Python soporta, además, cómo se incorporan los comentarios, las librerías y la generación de números aleatorios.

El capítulo 5, «Arreglos: vectores y matrices», aborda los arreglos unidimensionales y bidimensionales, los vectores y matrices, así como el uso de la librería NumPy, el ordenamiento y la obtención de los máximos y mínimos. También se explican los vectores paralelos y las operaciones aritméticas, las asignaciones condicionales, la copia y concatenación de arreglos, los filtros y las gráficas de funciones.

Por último, el capítulo 6, que se titula «Modularización: funciones y procedimientos», cubre temas como la modularización, los parámetros, las funciones y los procedimientos. También explica la combinación de funciones y procedimientos, desarrolla las variables locales y globales, y presenta las palabras reservadas.

Cabe aclarar que el libro no incluye todos los temas de un curso de programación, pues la propuesta es brindar los conocimientos básicos bajo el paradigma de la programación estructurada utilizando un lenguaje de alto nivel.

Como docentes de asignaturas sobre programación tanto en carreras de grado vinculadas a las ciencias informáticas como a otras, intentamos producir textos didácticos que acompañen a los estudiantes en sus primeros pasos. Con ese fin, recuperamos experiencias de la enseñanza y revisiones de otras obras que hemos producido como *Caminando junto al lenguaje C* y *Problemas y algoritmos. Un enfoque práctico*, ambas de la colección Lecturas de Cátedra de la Editorial UNRN de la Universidad Nacional de Río Negro. Estos materiales están disponibles en forma gratuita a través del sitio de la editorial: https://editorial.unrn.edu.ar/.

Para finalizar, invitamos a transitar el fascinante mundo de la programación y, sobre todo, a disfrutar cada momento del proceso en este recorrido con la certeza de que se generan conocimientos y desarrollan habilidades no solo para la resolución de problemas usando algoritmos, sino también para la aplicación de esta forma de pensamiento en otros ámbitos o situaciones de la vida cotidiana.

Autorías y filiaciones institucionales

Martín Mariano Julio Goin
Universidad Nacional de Río Negro, Centro Interdisciplinario de Estudios sobre Derechos, Inclusión y Sociedad Argentina. Río Negro, Argentina.

Edith Noemí Lovos
Universidad Nacional de Río Negro, Centro Interdisciplinario de Estudios sobre Derechos, Inclusión y Sociedad Argentina. Río Negro, Argentina.

Capítulo 1

Introducción a la programación

La mayor recompensa de mi trabajo no es lo que obtengo de él,
sino lo que me convierto mientras lo hago.
Margaret Hamilton

Python está diseñado para ser fácil de aprender y usar, lo
que permite a los principiantes comenzar rápidamente
sin verse abrumados por complejidades innecesarias.
Guido van Rossum

Temas:

Hitos de la programación. ¿Qué es programar? ¿Por qué aprender a programar? Algoritmo. Un primer acercamiento. Cómo diseñar algoritmos. Del algoritmo al programa. Python: historia y características.

1. 1. Hitos de la programación

Margaret Hamilton es científica, licenciada en Matemática y diplomada en Filosofía. Fue la directora de la división de ingeniería de la NASA que desarrolló el programa de la computadora a bordo del Apolo 11 (julio de 1969). Su aporte en el diseño de *software* fue extremadamente avanzado para la época, ya que se centró en la robustez y tolerancia de fallos. Podía detectar errores y corregirlos en el momento lo más rápido posible, priorizando la seguridad y la supervivencia de la tripulación del módulo lunar.

Hay que tener en cuenta que en esa década las computadoras todavía se componían de válvulas de vacío, es decir, que los ordenadores eran del tamaño de un aula. Por lo tanto, el desafío consistió en la reducción de los componentes para poder llevar adelante el programa aeroespacial. Los ingenieros de *hardware* lograron innovar, en 1966, una de las primeras computadoras de nombre Block II que usaba circuitos integrados (chips). El Block II tenía una velocidad de 2 MHz y una memoria de 125 kB, comparado a un teléfono celular es 16.000 veces más lento y con dos millones menos de capacidad de memoria.

Sin duda el Apolo Guidance Computer (AGC) fue una maravilla de la ingeniería y el *software* de Hamilton un éxito fundamental, sin ambos la misión habría sido imposible.

El trabajo de Hamilton en el programa Apolo 11 valió el título de ingeniera de *software* (carrera inexistente hasta el momento) y facilitó las bases para la industria moderna de desarrollo de este campo. Su innovación revolucionaria contribuyó a establecer a la carrera de Ingeniería de Software como una disciplina de estudio legítima.

Figura 1. 1. Hamilton con el código del programa de navegación del Apolo impreso en doce libros



Fuente: Draper Laboratory, 1969.

1. 2. ¿Qué es programar?

La acción de programar puede sintetizarse en saber usar un lenguaje de programación (sintaxis y semántica) para escribir (codificar) en el orden