

# Programación Estructurada en un Curso Introductorio. Una Experiencia Explorando Python.

Edith Lovos<sup>1</sup>, Tatiana Gibelli<sup>1</sup>, Rodolfo, Bertone<sup>2</sup>

<sup>1</sup> Universidad Nacional de Río Negro, Sede Atlántica  
8500 Viedma – Río Negro Argentina  
[\\_{elovos,tgibelli}@unrn.edu.ar](mailto:{elovos,tgibelli}@unrn.edu.ar)

<sup>2</sup>Instituto de Investigación en Informática III-LIDI.  
Facultad de Informática, Universidad Nacional de La Plata  
50 y 120, 1900 La Plata, Buenos Aires, Argentina  
[pbertone@lidi.info.unlp.edu.ar](mailto:pbertone@lidi.info.unlp.edu.ar)

**Abstract.** En este artículo se presentan los resultados preliminares alcanzados en relación al uso del lenguaje Python, en el marco de la implementación de una propuesta pedagógica para la enseñanza y aprendizaje de la programación estructurada. La experiencia se desarrolló en el marco de una asignatura correspondiente al primer año de la carrera Licenciatura en Sistemas de la Sede Atlántica de la Universidad Nacional de Río Negro. A partir de la experiencia, se analizan los beneficios y debilidades que presenta el lenguaje propuesto desde la perspectiva docente.

**Keywords:** Aprendizaje, Programación, Python, Ingresantes

## 1 Introducción

La asignatura que da lugar a la experiencia que se menciona en esta comunicación se denomina Programación de Computadoras I, en adelante PCI. Esta materia forma parte del plan de estudios de la carrera Licenciatura en Sistemas de la Sede Atlántica de la Universidad Nacional de Río Negro y pertenece al área Algoritmos y Lenguajes de Programación. La misma se dicta en forma presencial en el primer cuatrimestre del primer año con un total de 96 horas.

Cómo objetivos generales, desde PCI, se busca que los alumnos puedan analizar problemas resolubles con computadora, poniendo énfasis en la modelización, la abstracción de funciones y en la descomposición funcional de los mismos, a partir de un paradigma procedural/ imperativo. Se realiza una introducción de las nociones de estructuras de datos, tipos de datos y abstracción de datos.

En relación al enfoque adoptado para la enseñanza, se persigue el aprendizaje del proceso de resolución de problemas (análisis, diseño e implementación) en el marco

de actividades que promuevan en los alumnos el desarrollo de otras habilidades, como el trabajo en equipos colaborativos, el razonamiento crítico, y la capacidad de comunicación; en un espacio que no será siempre el del salón presencial de clases; como por ejemplo el uso del aula virtual de la asignatura.

En este sentido, desde el año 2013, se vienen desarrollando conjuntamente con las actividades de teoría y práctica del curso, actividades de laboratorio, denominadas Actividades Prácticas Entregables (APE) [1], que buscan a través de la resolución de problemas, la integración de los conceptos analizados en el curso. Las consignas de trabajo que se proponen a través de las APE, incluyen la definición del problema, formas de entrega consistente en un cronograma de actividades, fechas de previstas para cada etapa del proceso de resolución y recursos TIC a utilizar para su desarrollo, información sobre pruebas (se definen o se proporcionan los datos con los que serán puestos a prueba los programas y consideraciones especiales).

Desde los inicios de la carrera en el año 2009, el lenguaje de programación utilizado ha sido Pascal. Los resultados alcanzados con el uso de este lenguaje, coinciden con otros estudios a nivel nacional, que señalan por una parte dificultades técnicas como compiladores para plataformas obsoletas, problemas de configuración [2], y también otros relacionados a la empatía que los alumnos tienen con el lenguaje. A partir de este último concepto y teniendo en cuenta que la nuestra es una Universidad novel, que comenzó su actividad académica hace apenas 7 años y dónde la matrícula de ingresantes a la Lic. En Sistemas no es voluminosa, no contemplar esta visión puede como señalan algunos autores [3, 4], restar al objetivo de sostener a los alumnos a lo largo de la cursada. Por otra, desde otras materias de la carrera que le suceden a PCI existen requerimientos para que los alumnos lleguen a las mismas conociendo otros lenguajes, como por ejemplo C.

## 2 Explorando Python

Python es un lenguaje de programación creado a principios de los años noventa por el investigador holandés Guido Van Rossum. Y en términos de edad, podríamos decir que es joven comparado con otros lenguajes utilizados en el ámbito académico, como Pascal(1970), ó C(1972) . Una característica interesante teniendo en cuenta el auge del software libre, es su popularidad en esta comunidad. Así se puede acceder al mismo a través del sitio oficial <https://www.python.org>, evitando cualquier tipo de costos asociados a licencias.

Por otra parte, Python es un lenguaje interpretado, multiplataformas, de sintaxis y semántica sencilla y consistente. Y aunque es un lenguaje orientado a objetos, puede ser utilizado para soportar el paradigma procedural y funcional. Promueve la creación de código legible, maneja tipos dinámicos, gestión dinámica de la memoria (recolector de basura) y maneja excepciones. Al ser fuertemente tipado no esta permitido tratar una variable como si fuera de un tipo diferente al que tiene, sin embargo el lenguaje permite la conversión explícita entre todos los tipos de datos. Se

caracteriza también por su modularidad y por incluir una poderosa y extensa biblioteca de clases [5].

En relación a la formas de ejecutar código en Python, esto se puede hacer en una sesión interactiva (escribiendo líneas directamente en el intérprete y obteniendo la respuesta a cada una de ellas; o es posible escribir el código de un programa en un archivo con la extensión .py y ejecutarlo. Ambas formas pueden llevarse a cabo usando el entorno IDLE (Integrated DeveLopment Environment) que viene con los paquetes de Python.

Philip Guo [6], en un análisis sobre los lenguajes de programación más utilizados en la enseñanza en 39 centros de estudios de los Estados Unidos, señala a Python como el mejor posicionado, en los cursos introductorios a la disciplina. A nivel nacional es posible mencionar las experiencias de la FCEFyN - Universidad Nacional de Córdoba [9], la UNNOBA [2], la asignatura "Algoritmos y Programación 1", de la Facultad de Ingeniería de la UBA [7] y de la carrera de ingeniería informática de la Universidad Nacional de Avellaneda [10]. Asimismo algunas de estas experiencias han dado como resultados recursos bibliográficos y herramientas que se presentan como valiosos materiales de apoyo al uso de Python como lenguajes para la enseñanza y aprendizaje de la programación [8,11,12,13,16].

Por lo aquí expuesto, las características y experiencias de uso en docencia, ubican a Python como una opción posible para el lenguaje de programación que permitirá la implementación de los algoritmos en el marco del curso PCI, de una manera que resulte sencilla y amena para los alumnos que recién se inician en la actividad de programar.

### **3 Desarrollo de la Experiencia**

El programa de la asignatura PCI, consta de seis unidades didácticas, cada una con su correspondiente trabajo práctico y tres Actividades Prácticas Entregables (APE) integradoras de los conceptos tratados en la asignatura. Estas APE serán entregadas y evaluadas y las fechas de publicación de las mismas, están establecidas en el cronograma de actividades de la materia. Así mismo, en el programa de la materia se incluye un apartado especial que indica cómo será el desarrollo y evaluación de las mismas.

La asignatura prevé tres exámenes parciales que permiten evaluar los contenidos de las unidades trabajadas y una instancia global de recuperación al final de la cursada. Respecto al dictado, las actividades presenciales de la materia se desarrollan en dos clases semanales de 3 horas reloj cada una. Cada clase tiene una parte teórica y otra práctica. En la primera, se desarrollan los conceptos teóricos previstos en el plan de estudio haciendo uso de ejemplos prácticos que permitan la aplicación de los conceptos analizados. La parte práctica, tienen como objetivo la aplicación de los conceptos trabajados en la teoría, en la resolución de problemas computacionales, a través del diseño algoritmos. En un paso siguiente estas soluciones serán implementadas usando el lenguaje Python.

El énfasis de la asignatura está puesto en la parte práctica, en este sentido, para desarrollar la habilidad de resolver problemas usando algoritmos es fundamental el entrenamiento que se realiza a través de las situaciones que se plantean en los trabajos prácticos, mediante las cuales los alumnos se enfrentan con situaciones en las que tienen que decidir sobre la naturaleza del problema, seleccionar una representación que ayude a resolverlo (modelo) y, monitorear sus propios pensamientos (metacognición) y estrategias de solución. [14]

Durante la primera unidad del programa, resolución de problemas, se trabaja en papel usando pseudocódigo. Luego, a partir de la unidad II, una de las dos clases semanales, se desarrolla en el laboratorio de Sistemas, de manera que los alumnos puedan realizar la parte práctica en máquina y con la asistencia presencial de los docentes de la cátedra.

Para esta experiencia, el lenguaje de programación propuesto para la implementación de las soluciones algorítmicas, fue Python en su versión 3.4 y el alumno estaba habilitado para utilizar en la edición y ejecución de los programas el entorno IDLE (Interactive DeveLopment Environment), incluido con Python y/o el laboratorio virtual VPL que se encuentra dentro del aula virtual de la asignatura. Para el desarrollo de los ejercicios de la parte práctica de la materia, los alumnos usaron IDLE, en tanto que VPL se utilizó para el desarrollo de las Actividades Prácticas Entregables (APE), posibilitando que los grupos y el tutor compartieran el mismo espacio de edición de la implementación.

## 4 Resultados

En relación al pasaje del pseudocódigo a la implementación de los algoritmos usando Python, no se observaron mayores dificultades, entendemos que esto es así, debido a que la sintaxis del pseudocódigo se presenta en español, y como puede verse en el ejemplo de la tabla 1, y no difiere de la Python salvo por el idioma y los caracteres especiales (:) que se presentan como fin de las estructuras de control.

**Tabla 1.** Comparación Pseudocódigo - Python

Pseudocódigo - Algoritmo contarDig	Python – Programa contarDig.py
leer(N)	N = int(input('Ingrese un nro: '))
Num = N	Num = N
Cd = 0	Cd = 0
Mientras (Num > 0)	while (Num > 0):
Cd = Cd + 1	Cd = Cd + 1
Num = Num / 10	Num = Num // 10
mostrar(N, " tiene ", Cd, " digitos")	print(N, ' tiene ', Cd, ' digitos')

En este pseudocódigo, y al igual que en Python, no se usan delimitadores específicos de bloques, como Begin - End de Pascal, los bloques se delimitan por la

indentación, y cada línea termina sin ningún carácter de corte especial (a diferencia del símbolo ; de Pascal). Sin embargo uno de los primeros problemas a los que se enfrentaron los alumnos al implementar sus algoritmos en Python estuvo asociado a los nombres de las variables, dónde a diferencia de Pascal, estos resultan sensibles al tipo de letra (mayúsculas/minúsculas) y a las conversiones necesarias al momento de realizar una operación de entrada. Otra situación conflictiva tanto para principiantes como para quienes conocen otros lenguajes tipo Pascal, estuvo vinculado a la idea de variable como contenedor de información, ya que en Python las variables son referencias a posiciones de memoria. Sobre este tema, resulta fundamental el desarrollo de actividades prácticas que le permitan al alumno visualizar el concepto de variable e interpretar sus diferentes usos.

Desde la perspectiva del docente, una de las ventajas de utilizar Python está asociada a la facilidad de instalación del aplicativo, a diferencia de lo que sucedía, en experiencias anteriores utilizando Pascal con Lazarus o Geany. Desde el inicio de uso del lenguaje, las consultas de los alumnos se relacionaron con el problema y la expresión de la solución al mismo (sintaxis y semántica), pero en menor medida con cuestiones asociadas a la configuración del lenguaje de programación o del entorno. Así mismo, quienes llegan al aula con las netbooks del Programa Conectar Igualdad, ya tienen instalado Python, solo fue necesario verificar la versión disponible. Sobre esta cuestión, y coincidiendo con el expresado por Marzal[15], es deseable que el lenguaje de programación tenga en el curso un valor instrumental, interfiriendo lo menos posible en la implementación de los algoritmos.

PCI es un curso, que para la mayoría de los alumnos se presenta como su primer contacto con la actividad de programación, por este motivo es común que durante el aprendizaje, los mismos utilicen el método de prueba y error. Así, en el caso de la experiencia usando Python, el entorno IDLE facilitó la edición y ejecución y la detección de errores, gracias a características como el resaltado de errores, el formato del código y el resaltado de la sintaxis. Aún cuando el programa presente errores, es posible ejecutarlo y obtener algunos resultados parciales, permitiendo que los alumnos puedan descubrir no solo errores sintácticos sino también de semántica.

En Pascal, esta forma de trabajo implica primero eliminar todos los errores sintácticos y luego enfocarse en la semántica.

En cuanto al manejo de las estructuras de datos que se ven en el curso (arreglos, matrices, registros), el tipo lista de Python, resultó de utilidad para trabajar el concepto de secuencia y los algoritmos que permiten manejar estructuras de este tipo. Sobre este punto, en PCI, primero se presentó el concepto de arreglo como estructura de datos homogénea, de acceso directo y luego se utilizaron las listas de Python para implementar los mismos. Aquí se pudo observar que aunque en la práctica se realizaron las implementaciones de las operaciones asociadas a la estructura como por ejemplo buscar, pertenece, existe; luego en el examen parcial usaban directamente las instrucciones propias de Python. Ejemplo, el operador *in* para determinar si un elemento está o no dentro de una secuencia.

Un tema que resulto difícil desde la enseñanza, está asociado al concepto de registro, el cual no existe como tal en Python. Aquí se adoptó la línea presentada en el libro “Introducción a la programación con *Python 3*” [11] y se introdujo la idea de “

Tipos de Datos a Medida” a través del uso de clases, pero sin necesidad de entrar en detalles propios de la programación orientada a objetos.

Históricamente uno de los temas incluidos en el programa de la asignatura PCI, que resulta más difíciles de apropiar y significar para los estudiantes, es el de modularización. Aún cuando logren realizar un diseño modularizado de la solución al problema, luego será necesario utilizar los recursos del lenguaje de programación para implementarlo. En esta experiencia usando Python, el concepto de modularización, se comenzó a trabajar cuando solo se conocían los tipos de datos simples y así los parámetros a utilizar en los módulos, sean estos funciones o procedimientos, pertenecían solo a los tipos simples. Python no dispone de una forma explícita de definir funciones o procedimientos (Procedures y Function de Pascal), sin embargo cuando un subprograma se comporta como una función y retorna un resultado, se utiliza la palabra reservada return como instrucción final del módulo. Así la diferenciación entre funciones y procedimientos no resultó problemática desde el punto de vista de la implementación para el proceso de enseñanza y aprendizaje. Respecto a el pasaje de parámetros con estos tipos, denominados inmutables para Python, se trabajó de forma equivalente al pasaje de parámetros por valor que utiliza Pascal.

Las dificultades en relación al pasaje de parámetros, surgieron al comenzar a trabajar con los tipos compuestos (listas y listas enlazadas), ya que el manejo es similar al pasaje de parámetros por referencia de Pascal y la ausencia de un mecanismo explícito (var de Pascal) para diferenciar entre estos comportamientos, dificultó asimilar el concepto e implicó necesariamente tratar cuestiones propias de Python como son la mutabilidad e inmutabilidad de los objetos del lenguaje.[5,11]

Respecto a si la utilización de Python ha permitido sostener a los alumnos a lo largo de la cursada, el gráfico 1, muestra la distribución de los alumnos para los años 2013, y 2015, dónde se siguió la misma metodología de trabajo, solo que en 2013 se utilizó Pascal y en 2015 Python.



Fig. 1. Distribución de los alumnos durante los cursos 2013 y 2015.

En relación al abandono, aunque en ambos casos los resultados finales son similares, se observa que en la experiencia del año 2015 se ha logrado mantener el nivel de alumnos hasta la mitad del curso. Así mismo, el desarrollo de la primer APE para la cursada 2015, se pudo iniciar luego de transcurrida la semana 5 del curso (en 2013 se inició luego de la semana 6), y entendemos que esta situación se produjo porque resultó más fácil el uso y apropiación del lenguaje por parte de los alumnos.

En este sentido, resulta interesante mencionar que durante la experiencia con Python algunos alumnos utilizaron sus dispositivos móviles (celulares principalmente) para la actividad de programación, posibilitando esta actividad fuera del laboratorio físico y en un medio que les resulta familiar.

## 4 Conclusiones

Frente a Python, Pascal tiene la ventaja de haber sido diseñado para trabajar las técnicas de la Programación Estructurada. Sin embargo, como el objetivo de PCI no es aprender un lenguaje de programación sino comprender y adquirir destrezas para resolver problemas usando un enfoque procedural; puede decirse entonces que Python ha logrado cumplir ese objetivo. Así, a partir de una sintaxis y semántica sencilla se hace más simple el proceso de implementación de las soluciones algorítmicas. Logrando disminuir la sensación de frustración en los alumnos (principalmente para quienes la programación es una actividad novedosa) en relación a la vinculación con el lenguaje, el entorno de desarrollo, y el proceso de resolución de problemas. De esta forma, el lenguaje de programación se posiciona dentro del curso, como el recurso que permite expresar la solución algorítmica en términos de una sintaxis entendible por un autómatas y no como un obstáculo. Todo esto, acompañado por un entorno de desarrollo sencillo (IDLE), con características como el resaltado de sintaxis, la tabulación automática y el resaltado de errores en otras, que facilitan el proceso de implementación.

En relación a los resultados académicos del curso, no puede decirse que Python haya modificado notablemente los mismos, sin embargo y asociado nuevamente a la simplicidad del mismo, permite poner el foco en cuestiones complejas de la resolución de problemas algorítmicos siguiendo el paradigma procedural, como son en esta instancia de la carrera, la modularización, y el manejo de estructuras de datos.

## Referencias

1. Lovos, Edith. El uso de estrategias colaborativas mediadas por tecnología. La enseñanza de la programación en el primer año de la Licenciatura en Sistemas de la UNRN. Tesis de Maestría. Facultad de Informática, UNLP. (2014). Disponible: <http://hdl.handle.net/10915/38038>
2. Osella Massa, G.L., Russo, C. C., Sarobe, M., Pompei, S.: Análisis de nuevos lenguajes para la enseñanza de programación imperativa en los primeros años de las

- carreras de Informática de la Universidad Nacional del Noroeste de la Provincia de Buenos Aires. En Actas del XVIII Congreso Argentino de Ciencias de la Computación (2012). <http://hdl.handle.net/10915/23860>
3. Ramirez, A. O.: Python como primer lenguaje de programación. En: Publicación interna del Tecnológico de Monterrey, Campus Estado de México. (2010)
  4. Radenski, A.: Python First: A Lab-Based Digital Introduction to Computer Science. En ItiCSE'06: Proceedings of the 11th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education. pp.197–201 (2006)
  5. Peña Rosalia: Paseo por la programación estructurada y modular con Python. En Revista de Investigación en Docencia Universitaria de la Informática. Vol.8 Nro1.SSN 1989-1199 España. (2015)
  6. Philip Guo. Python is Now the Most Popular Introductory Teaching Language at Top U.S. Universities. (2014), <http://cacm.acm.org/blogs/blog-cacm/176450-python-is-now-the-most-popular-introductory-teaching-language-at-top-us-universities/fulltext>
  7. Wachenchauser, R.: Cátedra Algoritmos y Programación 1., <https://sites.google.com/site/fiuba7540rw/>
  8. Wachenchauser,R., Manterola, M., Curia,M.,Medrano, M., Paez, N. (2010) Algoritmos y Programación I con lenguaje Python. Disponible: [https://librosweb.es/libro/algoritmos\\_python/](https://librosweb.es/libro/algoritmos_python/)
  9. <http://python.org.ar/wiki/Proyectos/UsoDePythonEnLaUniversidad/>
  10. Wachenchauser, R. Programación con Python. En Portal Educar. <http://www.educ.ar/sitios/educar/recursos/ver?id=121388>
  11. Marzal A., Gracia Luengo V.I., García,P. (). *Introducción a la programación con Python 3. Publicacions de la Universitat Jaume I. Servei de Comunicació i Publicacions Campus del Riu Sec. Edifici Rectorat i Serveis Centrals. 12071 Castelló de la Plana. ISBN: 978-84-697-1178-1. (2014)*
  12. <http://www.pythontutor.com/>
  13. Marzal A., Gracia Luengo, I. : *Introducción a la programación con Python y C. Volumen I: Python.* Disponible:<http://ftp.tku.edu.tw/Linux/Gentoo/distfiles/python.pdf>
  14. Gary Stager: “En pro de los computadores”. Portal Eduteka (2003). Disponible: <http://www.eduteka.org/ProComputadores2.php>
  15. A Marzal, D. Llorens, I. Gracia : *Aprender a programar con Python: una experiencia docente.* Univ. Jaume I, <ftp://metalab.unc.edu/pub/Linux/docs/LuCaS/Presentaciones/200309hispalinux/15/15.pdf>
  16. Raúl González Duque. *Python para todos.* [http://www.ceibal.edu.uy/contenidos/areas\\_conocimiento/aportes/python\\_para\\_todos.](http://www.ceibal.edu.uy/contenidos/areas_conocimiento/aportes/python_para_todos)