

*Implantación de metodologías de cálculo a través del lenguaje Python para asignaturas impartidas en las titulaciones de la ETSIN*

*Implementation of Computation Methodologies through Python Programming Language for Subjects Taught in the School of Naval Engineering of the UPM*

Leo Miguel González, Jose Luis Cercós, Daniel Esteban Burgos

UNIVERSIDAD POLITÉCNICA DE MADRID

[Leo.gonzalez@upm.es](mailto:Leo.gonzalez@upm.es), [jl.cercos@upm.es](mailto:jl.cercos@upm.es), [d.estebanburgos@gmail.com](mailto:d.estebanburgos@gmail.com)

---

**Abstract**

*El objetivo de este proyecto es mejorar la eficiencia en los resultados de aprendizaje de los estudiantes, incorporando metodologías activas de aprendizaje como es la programación en el lenguaje Python. Este tipo de iniciativas se espera que favorezcan la interdisciplinariedad y la coordinación curricular de tipo horizontal y vertical. También buscamos desarrollar un repositorio de notebooks, recursos docentes, y videos online, que faciliten la experimentación así como el aprendizaje autónomo.*

*The global target of this project is to improve the efficiency of the students' learning results by incorporating active learning methodologies such as the Python programming language which will improve the interdisciplinarity and both horizontal and vertical academic coordination. We are also developing a complete repository, educational resources, and online videos in order to make the interaction and self-learning easier.*

---

**Keywords:** OpenCourseWare, Python, improve learning skills, increase in the quality of teaching skills, Innovation in the development of subjects, tutorials

**Palabras clave:** OpenCourseWare, Python, Mejora del proyecto de aprendizaje, Mejora de la calidad docente, Innovación de la calidad docente, Innovación en el desarrollo de asignaturas, guías de aprendizaje, Creación materiales

## 1 Introducción

En las escuelas de ciencias e ingeniería el ordenador es la herramienta que sin lugar a dudas ha cambiado de forma más relevante la forma de estudiar y de aprender por parte del estudiante. El hecho de que cada estudiante tenga acceso directo mientras trabaja a un número casi infinito de datos relacionados con su tarea académica a través de internet, así como la posibilidad de hacer uso de herramientas especialmente indicadas para algunas de las subtarefas que aparecen constantemente en una jornada de estudio, rompen con el método clásico de estudio que un alumno tenía hace años. Un buen número de estudiantes tiene ordenadores portátiles personales y/o constante acceso a los recursos informáticos de la facultad donde cursan sus estudios, esto permite que un adecuado software de ayuda siempre este accesible. Editores de texto, programas de CAD, asistentes matemáticos, hojas de cálculo, etc. . . son ejemplos de lo que aquí se está hablando. En algunas ocasiones el estudiante debe ser capaz de interactuar de forma no trivial con dicho software para conseguir ejecutar las tareas precisas que necesita, siendo un ejemplo claro de ello la posibilidad de programar en los distintos lenguajes que cada software requiera. Debemos aclarar en este momento que el proyecto de innovación educativa que se documenta aquí va mucho más allá de la discusión sobre cuál es el lenguaje de programación ideal. Creemos que además de las múltiples ventajas que la programación le permite al ingeniero, hay una más que no debe ser pasada por alto y es que “la programación educa”. Obligar a un alumno a hacer entender a un sistema objetivo, neutro, inflexible y ajeno a las presiones humanas en el que se desea realizar un cálculo bien definido, desarrolla tanto el ingenio como la atención y la humildad por parte del estudiante. Nuestro contexto, es claramente el de una escuela de ingeniería o el de una facultad de ciencias, donde asumimos un grado de desarrollo global en el que todos los alumnos tienen acceso a los recursos informáticos de un modo u otro. En numerosas ocasiones, para obtener resultados a la hora de superar asignaturas impartidas en titulaciones técnicas, no solo depende del buen entendimiento conceptual y teórico de las áreas a estudiar, sino de tener la capacidad de realizar cálculos matemáticos de cierta complejidad en un tiempo razonable que modelan un cierto problema tecnológico. Este trabajo se inspira en el trabajo previo que ha mostrado la profesora Lorena A. Barba de la Boston University (véase Lorena Barba), quien ha empleado una metodología docente basada en un lenguaje de programación a una asignatura concreta como es la aerodinámica. La profesora Barba muestra como con un lenguaje de programación sencillo, y unos tutoriales bien adaptados, se puede conseguir que el alumno se adentre en los problemas conceptuales de la asignatura sin que la resolución de las ecuaciones sea un problema. En concreto, se llega a mostrar cómo es posible enseñar a resolver un sistema complejo de ecuaciones diferenciales en derivadas parciales no lineal como son las ecuaciones de Navier-Stokes, en tan sólo 12 pasos.

## 2 Objetivos

Nuestro objetivo principal no es otro que dotar al alumno de una herramienta de libre acceso para realizar cálculos matemáticos que sea transversal a todas las asignaturas de la carrera y que incluso pudiera tener una cierta proyección profesional. Otros objetivos más concretos son:

1. Dotar al alumnado de una herramienta computacional en forma de lenguaje de programación que de forma sencilla y rápida permita al estudiante tener una mayor eficiencia académica.
2. Que el alumnado se encuentre mejor preparado a nivel técnico en su próximo futuro tec-

nológico.

3. Que el alumno de ciencias e ingeniería sepa distinguir entre las dificultades de modelado matemático de un problema y la resolución del mismo. Para esta segunda parte, debemos entender que siempre y cuando el problema no requiera de una dificultad excesiva y se deba recurrir a programas comerciales especializados, el alumno debe ser capaz de abordarlo mediante las herramientas aquí indicadas.
4. Facilitar a los estudiantes de *Grado y de Master de la ETSI Navales*, mediante recursos online, el conocimiento de metodologías de cálculo a través del lenguaje de programación interpretado Python, que podrán aplicar a un gran número de las asignaturas impartidas en las distintas titulaciones.
5. Motivar a los estudiantes de ciencias e ingeniería a realizar pequeños programas que permitan resolver de forma sistemática problemas de distintas asignaturas, haciendo de la cultura de la programación una forma de estudio transversal.

### 3 Desarrollo del proyecto

El desarrollo del proyecto ha tenido las siguientes etapas:

1. Creación de la página web, ver <http://canal.etsin.upm.es/iepython/> (véase González, et al.), donde se van a colgar todos los resultados de este trabajo. Ver logotipo en la Figura 1.
2. Selección del becario y el software libre adecuado, para el desarrollo de notebooks de Python on-line y para la grabación y edición de videos.
3. Elaboración de un listado de tareas y ejemplos por las que el alumno de ciencias e ingeniería debe progresar de forma secuencial para incorporar este lenguaje de programación a su método de estudio.
4. Difusión de los contenidos del proyecto.



Figura 1 – Logotipo del proyecto de innovación educativa IE-Python.

### 4 ¿Por qué Python?

Nos hemos decantado por el lenguaje de programación Python porque es una herramienta de software libre y gratuito, de uso razonablemente simple, que permite realizar diversos cálculos en

muchas ramas de las ciencias e ingeniería. El conocimiento de comandos e instrucciones básicas de dicha herramienta permitirá a los estudiantes de ciencias e ingeniería, y en concreto, de la ETSI Navales, enfrentarse a la resolución de problemas que contienen cálculos complejos. Valga como ejemplo, el cálculo de la trayectoria y velocidad de un oscilador armónico amortiguado, mostrado en la Figura 2; o la ecuación del atractor de Lorenz de la Figura 3.

Al igual que otros entornos de programación amigables como por ejemplo Matlab y Octave, Python tiene una curva de aprendizaje claramente indicada para estudiantes de grado. No debemos olvidar, que en el contexto donde nos encontramos, las herramientas computacionales aún hoy en día encuentran ciertas reticencias entre el alumnado debido a una cierta intolerancia a la frustración (véase Guzdial, 2010). Lenguajes interactivos y amigables contribuyen a una metodología y un planteamiento mucho más ordenado y sistemático, permitiendo obtener unos buenos resultados a la hora de ser aplicado a las distintas asignaturas.

Entre las ventajas que nos han hecho decantarnos por un entorno Python podemos citar:

1. Python es un lenguaje de programación versátil: se pueden analizar datos, construir páginas webs, mantener servidores, incluso hacer arte o música.
2. Los ingenieros de muchas empresas técnicas de primer nivel (por ejemplo Airbus, Navantia, etc.) emplean Python de forma habitual, por lo tanto si el estudiante domina Python aumenta claramente su potencia curricular.

```
In [2]: import matplotlib.pyplot as plt
import numpy as np
from scipy.integrate import odeint # lo escribimos así porque de scipy.integrate solo queremos odeint

In [3]: k = 4 #Definimos la constante del muelle
m = 1 #Definimos la masa
B = -1 #Definimos la constante de amortiguacion

def harmonic((x, y),t):
    return [y, -k*x/m + B/m*y]

inicial = [0.7, 0.5] #Vector de condiciones iniciales x(t=0)=0.7 y(t=0)=0.5

t_output = np.arange(0, 15, 0.01) #Dominio temporal de 0 a 15 y paso de tiempo 0.01 (nuestro eje x)

result = odeint(harmonic, inicial, t_output) #Resolvemos el sistema con odeint(sistema, condicion inicial
# , El rango donde graficaremos)

In [4]: fig, ax2 = plt.subplots(ncols = 1, figsize=(10,10))

xx, yy = result.T # extraer columnas y filas

plt.xlabel('tiempo')
plt.plot(t_output, xx,label="Posicion") #label lo utilizamos para poner una leyenda
plt.plot(t_output, yy,label="Velocidad")
plt.legend() #Con esto ponemos la leyenda
plt.show()
```

Figura 2 – Código Oscilador armónico.

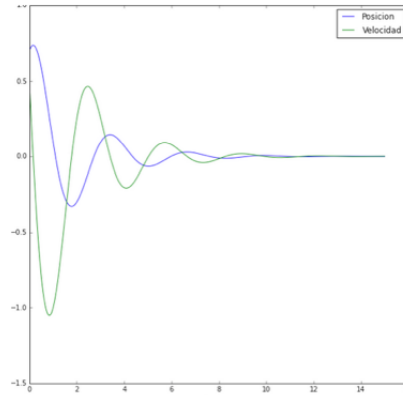


Figura 3 – Gráfica obtenida mediante la ejecución del programa del Oscilador Armónico donde se muestran las gráficas de posición y velocidad en función del tiempo para un caso amortiguado.

3. Python es un lenguaje de programación ideal para la enseñanza, un buen número de instituciones educativas, algunas tan reconocidas como el MIT, están empleando Python como lenguaje de programación.
4. Leer un código en Python es como leer en inglés por lo tanto carece de la menor complejidad sintáctica (véase Manilla et al., 2006).
5. Python tiene una eficaz curva de aprendizaje, pudiéndose realizar mucho trabajo valioso en poco tiempo.
6. A diferencia de otros lenguajes de programación, como por ejemplo Fortran, la librería Matplotlib permite realizar gráficos y animaciones en 3D (véase Matplotlib Library).
7. Se pueden escribir juegos de ordenador en Python mediante la librería PyGame.

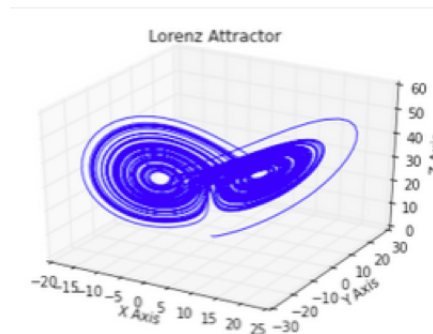


Figura 4 – Representación tridimensional de la solución proveniente del código del Atractor de Lorenz.

```

In [1]: from mpl_toolkits.mplot3d import Axes3D

In [2]: a = 10
        b = 28
        c = 8 / 3

        def lorenz(x, y, z):
            u = a * (y - x)
            v = b * x - y - x * z
            w = x * y - c * z
            return u, v, w

        dt = 0.01
        stepCnt = 10000

        xs = np.empty((stepCnt + 1,)) #Necesitamos un valor mas para el contador
        ys = np.empty((stepCnt + 1,))
        zs = np.empty((stepCnt + 1,))

        xs[0], ys[0], zs[0] = (0., 1., 1.05) #Fijamos valores iniciales de x,y,z

        #Iniciamos un contador con salto de tiempo "dt" hasta "stepCnt"
        for i in range(stepCnt) :
            # Derivadas de x, y, z.
            x_dot, y_dot, z_dot = lorenz(xs[i], ys[i], zs[i])
            xs[i + 1] = xs[i] + (x_dot * dt)
            ys[i + 1] = ys[i] + (y_dot * dt)
            zs[i + 1] = zs[i] + (z_dot * dt)

In [3]: fig = plt.figure()
        ax = fig.gca(projection='3d')

        ax.plot(xs, ys, zs)
        ax.set_xlabel("X Axis")
        ax.set_ylabel("Y Axis")
        ax.set_zlabel("Z Axis")
        ax.set_title("Lorenz Attractor")

        plt.show()

```

Figura 5 – Código Atráctor de Lorenz.

8. Python a diferencia de otros lenguajes de programación como Matlab es gratuito, esto representa una gran ventaja para el alumno ya que lo puede adquirir y utilizar sin coste y sin la necesidad de cometer delito alguno sobre la propiedad intelectual. Otra ventaja adicional derivada de esto último es que los contenidos en Python son trasladables a estudiantes y profesores de universidades de países menos desarrollados debido al nulo coste del software.
9. Python minimiza el número de errores que el alumno comete en programación y por consiguiente hace descender la frustración y la sensación de fracaso que el joven alumno de ingeniería detesta.

Nuestra sensación es que un semestre de un curso de Matlab, C++ o C no dota a los estudiantes de una herramienta accesible, útil y sencilla para resolver sus problemas académicos. En el contexto educativo actual, los planes de estudio tan solo incluyen en la mayoría de los casos un cuatrimestre de lenguajes de programación. Esto es precisamente lo que queremos abordar en este proyecto, como dotar en poco tiempo y con poco esfuerzo al alumnado de habilidades computacionales para ser más eficaces en su estudio de las distintas ramas de la ciencia y la ingeniería.

En el libro “How to think like a computer scientist” (J. Elkner, 2002), defiende que el uso de Python ha mejorado la efectividad en el aprendizaje de las asignaturas de programación en su institución educativa. Según sus datos, un mayor número de estudiantes terminan el curso con posibilidad de crear programas de gran utilidad y con una actitud altamente positiva hacia el mundo de la programación.

Históricamente, nuestro contexto académico ha ofrecido al estudiante el programa Matlab como solución a los problemas que aquí pretendemos resolver, en este sentido es interesante citar el estudio llevado a cabo por A. Souto-Iglesias (A. Souto-Iglesias, 2014) sobre el debate relativo a software libre y propietario en un contexto educativo. A continuación damos una argumentación de por qué Matlab no cubre nuestras expectativas:

- Python se encuentra actualmente a la altura de entornos de programación como Matlab y en breve los adelantará.
- Python es un verdadero lenguaje de programación.
- Python puede comunicarse fácilmente con otros lenguajes de programación.
- Con Python uno puede trabajar íntegramente en un entorno open-source con el consabido ahorro que ello implica para el alumno y la institución.
- Las licencias de Matlab son complicadas y caras.
- Todo lo que es posible hacer en Matlab se puede hacer en Python pero no al revés.

## 5 Ejemplos de aplicación

Los ejemplos van desde una introducción al entorno de trabajo Python, hasta la resolución de los problemas típicos que un alumno puede encontrar a lo largo de sus estudios de grado. Los dos primeros ejemplos, “Introducción a Python” y “Python como calculadora” son claramente introductorios y presuponen una experiencia nula en este tipo de entorno. Tras el primer ejemplo, el alumno debe entender en que entorno de programación se va a mover y se muestra la forma de trabajo on-line que se va a promover a lo largo del curso. Se busca por tanto que el alumno para poder trabajar con Python pueda elegir entre hacerlo on-line sin tener que instalar nada en su propio ordenador pero conectado a internet o bien instalar una distribución de Python sin tener que estar obligatoriamente conectado a la red. En este proyecto se ha apostado por la primera forma de trabajo, debido a la facilidad mostrada en el proceso de iniciación. A continuación, una vez el estudiante se ha acomodado en el entorno de trabajo, se le explican los procesos más básicos que se pueden realizar en Python. A esta segunda etapa la hemos bautizado “Python como calculadora”, y creemos que tiene una vital importancia para que el alumno recién iniciado pierda el miedo al entorno y vea de modo inmediato las aplicaciones prácticas de este proceso de aprendizaje.



Aquí tenemos una muestra de los ejemplos que hemos tratado en este proyecto:

Ejemplo	Conceptos tratados	Asignaturas implicadas
Introducción a Python	Básicos	Lenguajes de Programación.
Python como calculadora	Realización de cálculos simples	Física, Mecánica, Álgebra, etc...
Oscilador armónico	- Física básica - EDOs	- Vibraciones - Mecánica
Trayectoria de un tiro parabólico	- Leyes de movimiento - Sistemas de EDOs	- Mecánica - Física
Tiro parabólico de un sólido aerodinámico (pluma de Bádmiton)	- Sistemas de EDOs - Leyes de movimiento. - Resistencia y sustentación aerodinámicas.	- Mecánica - Mecánica de Fluidos
Ecuaciones de Lorenz	- Sistemas de EDOs - Sistemas complejos y caos.	- Mecánica
Diagrama de Moody, ecuación de Colebrook-White	- Sistemas de ecuaciones no lineales. Flujo turbulento incompresible en tuberías.	- Mecánica de Fluidos. - Hidráulica.
Cuerpo semi-infinito de Rankine	- Fluido ideal	- Mecánica de Fluidos

Tabla 1 – Listado de los ejemplos a los que se ha recurrido para mostrar la metodología de Python como herramienta de aprendizaje.

El resto de los ejemplos, son claras aplicaciones de distintas asignaturas típicas de ciencias e ingeniería. Comenzando por la resolución de un oscilador armónico y acabando por un típico ejemplo de fluido ideal como es la determinación del punto de máxima velocidad sobre un ovalo de Rankine. Estos ejemplos, tienen por objeto mandar un mensaje claro al alumno, “en general, se puede hacer un estudio muy completo de un problema de ingeniería sin necesidad de conocer los detalles numéricos o analíticos de la resolución del mismo”, obviamente este mensaje tiene sus peligros en ciertos casos y conviene matizarlo diciendo que:

- En ciertos casos la resolución numérica exige tratamientos específicos cuando los problemas son muy rígidos.
- Tras un modelo matemático resuelto con Python, es obligatoria una validación y/o comprobación tan exhaustiva como sea posible.
- Python tan sólo resuelve unas ecuaciones, si estas son incorrectas o están mal programadas, el error se extiende a la solución.
- Una vez se ha obtenido una primera solución aceptable del problema, es recomendable que el alumno trata de optimizar el código en términos de minimización de tiempo y memoria.








Como se comentaba anteriormente, para cada uno de estos casos descritos en la Tabla 1, se ha realizado un Notebook y un vídeo explicativo que describe paso a paso como se han elaborado dichos Notebooks con una voz que narra las tareas en castellano. Los vídeos son accesibles tanto desde la página web del proyecto, como desde la plataforma Youtube, esto último debido a la facilidad que tienen los estudiantes para trabajar con ella. Este moderno material didáctico hace que el alumno pueda siempre tener un material de apoyo a la hora de realizar sus propios ejercicios.



## 6 Conclusiones

En este proyecto de innovación educativa, se ha conseguido elaborar una serie de materiales on-line que permiten al alumno de grados de ciencias e ingeniería acometer el aprendizaje de forma rápida y progresiva del entorno de programación Python. Una vez el alumno recorra la totalidad de los tutoriales audiovisuales elaborados se debe sentir capaz de dar un salto al uso de Python para sus propios casos, y esperemos que esta herramienta vaya integrada en su vida profesional como una segunda piel.

## Referencias

- 
[Lorena Barbá.](http://lorenabarba.com/blog/announcing-aeropython/)  
<http://lorenabarba.com/blog/announcing-aeropython/>.
- 
[González, L. M., Cercós, J. L., Burgos, D. E.](#)  
*Implantación de metodologías de cálculo a través del lenguaje Python para asignaturas impartidas en las titulaciones de la ETSIN.*  
 Proyecto de Innovación Educativa de la ETSI Navales. Curso 2014–15.  
<http://canal.etsin.upm.es/iepython/>
- 
[Guzdial, M. \(2010\).](#)  
*Why is it so hard to learn to program?*  
 Editors, In Andy Oram and Greg Wilson.
- 
[Mannilla, L. et al. \(2006\).](#)  
*What about a simple language? Analyzing the difficulties in learning to program.*  
 Computer Science Education 16(3), 211–227.
- 
[Matplotlib library.](#)  
<http://matplotlib.org/>.
- 
[Elkner, J., Meyers, C. \(2002\).](#)  
*How to think like a computer scientist. Learning with Python*  
 Allen Downey, Green Tea press // Free book.
- 
[Souto-Iglesias, A. \(2014\).](#)  
*Software libre y propietario en el contexto de la Educación Superior en España: elementos para un debate.*  
 Revista Puig-Adam. N97.

Modelling in Science Education and Learning  
<http://polipapers.upv.es/index.php/MSEL>