

© **RIDHyC**, Nº 2.:159-170 (Diciembre, 2015) · ISSN (2386-7795)
www.didacticahumanidadesyciencias.com

NUEVAS HERRAMIENTAS DOCENTES: LOS CUADERNOS IPYTHON

New teaching tools: iPython

ANDRÉS SUÁREZ-GARCÍA
Profesor Doctor
Centro Universitario de Defensa, Universidade de Vigo
asurez@tud.uvigo.es

MARÍA ELENA ARCE
Profesora Doctora
Centro Universitario de Defensa, Universidade de Vigo
elena.arce@tud.uvigo.es

MIGUEL ÁNGEL ÁLVAREZ-FEIJOO
Profesor Doctor
Centro Universitario de Defensa, Universidade de Vigo
alvarezfeijoo@tud.uvigo.es

RAQUEL FERNÁNDEZ-GONZÁLEZ
Profesora Doctora
Centro Universitario de Defensa, Universidade de Vigo
raquelf@tud.uvigo.es

Resumen

El grado de Ingeniería Mecánica combina tanto conocimientos como habilidades en las áreas de Ciencias, Tecnología, Ingeniería y Matemáticas (STEM, por sus siglas en inglés). En la enseñanza de las asignaturas STEM se debe enfatizar la asociación de los conceptos físico-matemáticos con el mundo real. Para facilitar este paso, se ha de utilizar todos los materiales disponibles, dotando a las tareas de cierta transversalidad. Dentro de este campo, una de las herramientas más innovadoras son los cuadernos de trabajo iPython. Dichos cuadernos permiten introducir texto, vídeo o incluso gráficos interactivos con el fin de transmitir o revisar nuevos conceptos e interactuar con ellos. Este trabajo presenta los resultados de la experiencia de trabajar con cuadernos iPython para realizar prácticas en el marco de la asignatura Ingeniería de Fabricación y Calidad Dimensional, del cuarto curso del grado impartido en el Centro Universitario de la Defensa de la Escuela Naval Militar de la Armada Española. Se explicará el proceso de instalación, creación y uso de los cuadernos iPython, así como la opinión de los estudiantes que utilizaron esta herramienta.

Palabras clave: Educación Universitaria, Didáctica de la Ingeniería, Nuevas herramientas, Cuadernos iPython

Abstract

Mechanical Engineering degree combines both knowledge and skills in the areas of Science, Technology, Engineering and Mathematics (STEM acronym in English). In the teaching of STEM subjects should emphasize the association of physical-mathematical concepts to the real world. To facilitate this step, you must use all available materials, providing certain tasks mainstreaming. Within this field, one of the most innovative tools are ipython notebooks work. These notebooks let you enter text, video or even interactive graphics in order to transmit or review new concepts and interact with them. This paper presents the results of experience of working with notebooks ipython for internships in the framework of the subject Manufacturing Engineering and Dimensional Quality fourth degree course taught at the University Center for the Defense of the Naval Academy Navy Spanish. the installation process, creation and use of ipython notebooks, as well as the opinion of the students who used this tool will be explained.

Keywords: University Education, Teaching Engineering, New Tools, Notebooks iPython

1. INTRODUCCIÓN

Con la creación del EEES, la innovación docente ha adquirido una gran repercusión en el sistema universitario. Es por ello que los centros de enseñanza superior están llamados a desarrollar un plan de innovación orientado hacia la convergencia. La concreción de propuestas en esta línea requiere partir un modelo educativo cuya prioridad máxima sea el aprendizaje y asumir que la función del profesor universitario consiste en posibilitar, facilitar y guiar al alumnado para que éste pueda acceder intelectualmente a los contenidos y prácticas profesionales, de modo que sea capaz de construir el conocimiento, así como de interpretar la realidad que le rodea. Esta asunción lleva implícita la aplicación de métodos y estrategias didácticas, de tipo constructivista, que permitan al alumno aprender a aprender, generando un ambiente que propicie el incremento de la autonomía personal de los estudiantes, al tiempo que fomente el pensamiento crítico y la reflexión sobre su proceso de aprendizaje. A su vez, debe posibilitar el aprendizaje cooperativo y, por extensión, el trabajo en equipo. Para hacer efectiva esta metodología se ha de considerar distintos tipos de agrupamiento del alumnado. Según De Miguel (2006), la multiplicidad de métodos aparece como la vía más fructífera para emprender la renovación didáctica de la enseñanza universitaria.

En este contexto, las Tecnologías de la Información y Comunicación (en lo sucesivo TIC) cobran una especial relevancia, bien sea por el efecto facilitador que tiene en el aprendizaje (Alcántara, 2009), o bien por la aplicabilidad que presenta en la vida cotidiana (Winocor, 2007). La incorporación de este recurso al proceso de formación supone un desafío significativo, no sólo porque la presencia de las TIC en la sociedad y en la enseñanza está evolucionando continuamente (Selwyn y Gouseti, 2009), sino también porque los alumnos van a encontrarse con una serie de

dificultades para asimilar estas nuevas metodologías. No obstante, las TIC constituyen un poderoso medio didáctico, en tanto que permiten individualizar la enseñanza (Pensosi, 2010) y facilitan el procesamiento de la información por diferentes canales (Alcántara, 2009).

Una de las herramientas más prometedoras en el ámbito TIC son los cuadernos de trabajo iPython. La finalidad de estos cuadernos es convertirse en un método ágil para la divulgación científica y el desarrollo de nuevas ideas. Sin ir más lejos, *Nature*, una de las revistas científicas más prestigiosas a nivel mundial (Python Development Team, 2015), ha comenzado a recomendar su uso basándose en su extensa comunidad de usuarios (Macmillan Publishers, 2015) y la transparencia en la reproducibilidad que conlleva su uso (Perkel, 2015). Entre sus beneficios, destaca la reproducibilidad de cualquier estudio realizado, puesto que los datos y resultados quedan almacenados o referenciados en el cuaderno iPython. Para hacerse una idea de la potencialidad de un cuaderno iPython como herramienta docente, se pueden visitar distintas galerías de cuadernos existentes en la web (Shen, 2014; Nbviewer, 2015; Wittenauer, 2015). Además del encapsulamiento de contenido multimedia, los cuadernos iPython ofrecen la creación de experimentos virtuales (e-experimentos) mediante el famoso lenguaje de programación Python (Notebook Gallery, 2015). Su utilidad se ha extendido hasta en las mejores instituciones, puesto que incluso para ellas existen experimentos cuya realización es inalcanzable debido al alto coste de los equipamientos necesarios o la imposibilidad de llevarla a cabo. En estos casos, la mejor alternativa es la creación de e-experimentos en los cuales el estudiante tenga la libertad de experimentar, permitiendo una experiencia didáctica más rica y creativa debido a la posibilidad de “tocar” el problema modificando simulaciones prescritas, sin miedo a dañar equipos costosos. Finalmente, destacar que un cuaderno iPython, una vez programado, puede ser utilizado y disfrutado sin necesidad de reprogramar una sola línea de su código.

Tomando estas máximas, en el presente trabajo, se propone el uso de cuadernos de trabajo iPython para el desarrollo de contenidos aplicados. Más concretamente, se utilizará un cuaderno iPython en sesiones prácticas para resolver problemas de optimización relacionados con procesos industriales.

1. MATERIALES Y MÉTODOS

1.1. Cuadernos de notas iPython

iPython es un entorno de trabajo basado en tecnologías web que se puede utilizar como cuaderno de notas (Figura 1). Esta tecnología permite su visualización en cualquier navegador. La principal ventaja de los cuadernos iPython (iPython Notebook) para los desarrolladores es su alta interactividad, motivada tanto por la posibilidad de incluir contenido multimedia, como de poder ejecutar código escrito. En lo que respecta a los usuarios, para poder visualizar e interactuar con los cuadernos iPython no es necesario que tengan conocimientos de programación. Esta

polivalencia ha hecho incrementar el interés de la comunidad por esta plataforma, tal y como reflejan las continuas donaciones que está recibiendo el proyecto para su continuo desarrollo por parte de gigantes de la industria del software (IEEE Spectrum, 2015).

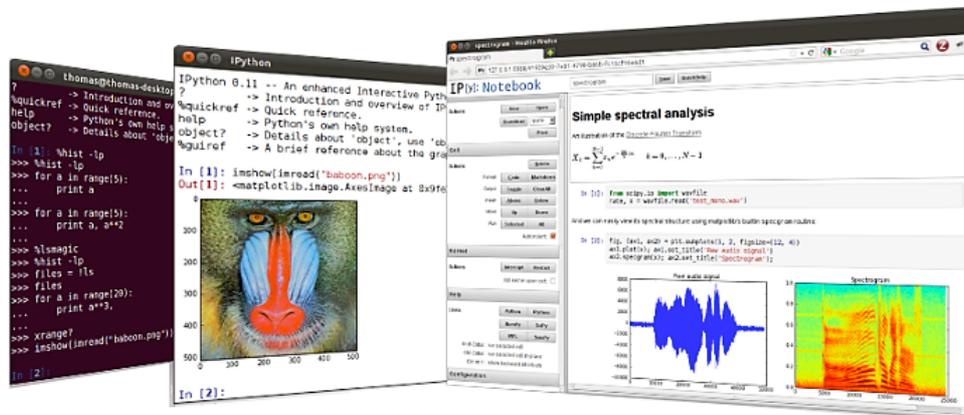


Figura 1 Evolución del entorno iPython (Recuperado de <http://ipython.org/>)

1.2. Instalación y uso de iPython

Para facilitar la instalación se recomienda descargar e instalar software científico basado en Python, pues iPython Notebook es una interfaz de iPython. Actualmente, las distribuciones más conocidas son Anaconda (Python Development Team, 2015) y Canopy Continuum Analytics, 2015). Ambas gratuitas y con una amplia comunidad de usuarios. En el trabajo aquí presentado, se usará la distribución Anaconda.

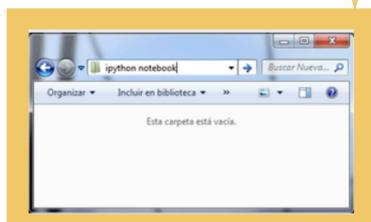
1.2.1. Primeros pasos

La instalación del entorno iPython es un proceso sencillo en cualquier plataforma seleccionada para su uso. iPython es un entorno multiplataforma y la instalación y uso en Linux y OSX es igual de sencilla que en Windows, detallada esta última en la Figura 2. Una vez instalado el software, el trabajo con los cuadernos iPython se realiza a través de un navegador web. Para iniciarlo, hay que situarse en el directorio de trabajo y ejecutar el comando indicado en el segundo paso 2.

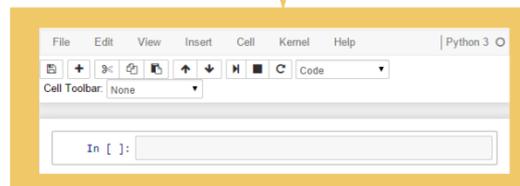
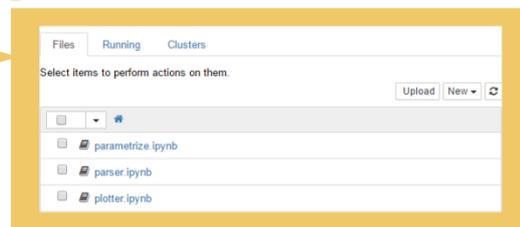
1. DESCARGAR E INSTALAR



2. LANZAR NOTEBOOKS



3. CREAR O SELECCIONAR NOTEBOOK



4. PROGRAMAR NOTEBOOK

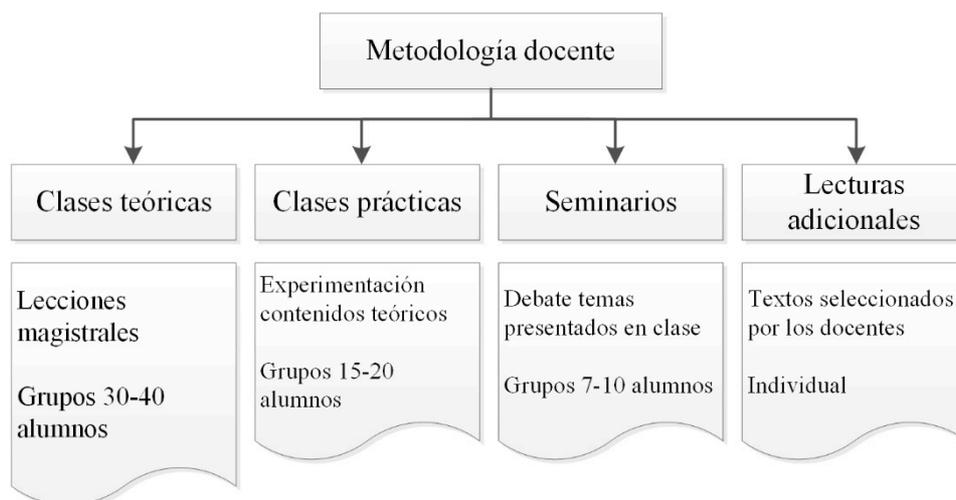
Figura 2 Flujograma instalación de software para cuadernos iPython

Una vez arranque la sesión de trabajo, pulsando el botón "New Notebook" se creará un documento dividido en una serie de celdas (paso 4 de la Figura 2). Las celdas constituyen las unidades básicas del cuaderno de trabajo en las cuales introducir texto, objetos multimedia y código de programación. Una vez introducido el programa, si queremos que se ejecute habría que darle al botón con forma de "play" en la barra superior o pulsar la combinación de teclas "Shift+Intro".

1.3. Planteamiento del problema

La asignatura Ingeniería de Fabricación y Calidad Dimensional tiene como objetivo desarrollar los conocimientos, habilidades y capacidades para clasificar y seleccionar los distintos procesos de fabricación que mejor se adecuen a cada sector productivo. Esto implica que el alumno deberá adquirir capacidades para identificar y planificar las diferentes etapas del proceso productivo a partir de las especificaciones de diseño de producto, seleccionando las distintas fases, máquinas, equipos, utillajes, herramientas y técnicas de verificación más convenientes. En este sentido, uno de los aspectos más importantes a tener en cuenta en la ingeniería de fabricación es la maximización de los beneficios.

En lo que respecta al diseño didáctico, se ha propuesto la utilización de cuatro métodos. Aun cuando cada uno de ellos tiene sus propias ventajas e inconvenientes por separado, su utilización está, en buena medida, determinado por la complementariedad de los mismos. En la Figura 3 se resume de forma esquematizada las metodologías docentes aplicadas a la materia que nos ocupa. Con el fin de poder aplicar esta herramienta de una forma más individualizada, los cuadernos iPython se utilizaron en las sesiones de seminarios y prácticas.

**Figura 3:** Metodologías docentes

1.3.1. Ejemplo práctico

El problema propuesto se encuentra descrito en la **iError! No se encuentra el origen de la referencia.**4. Se trata de una empresa que tiene como objetivo maximizar sus ganancias combinando óptimamente el número de unidades producidas en función de diferentes restricciones

(e.g. costes de materia prima, fabricación, etc.). El problema se formula matemáticamente como un caso de optimización lineal, cuya función a maximizar está descrita en la ecuación **iError! No se encuentra el origen de la referencia.** y las restricciones de coste, horas de fabricación y unidades mínimas en los bloques de Ecuaciones **iError! No se encuentra el origen de la referencia.** y (3). Las soluciones han de ser números enteros mayores que cero.

1.3.1.1. Enunciado del problema

"Usted forma parte del laboratorio farmacológico de las Fuerzas Armadas del Ejército Español. Su labor, a día de hoy, consiste en realizar la comida de superviviencia al mínimo coste posible. Resultará indispensable que el producto final tenga todos los nutrientes mínimos recomendados por la Organización Mundial de la Salud. Los ingredientes utilizados serán carne de pollo y ternera, además de arroz y trigo.

En la fabricación de las raciones se utilizará carne de pollo y ternera cuyos precios son 0.013 € y 0.08 € respectivamente; mientras que el precio del arroz y del trigo son 0.002 € y 0.005 €, respectivamente.

Cada ingrediente contribuye al contenido total de proteína, grasa, fibra y sal en el producto final. Las contribuciones, en gramos, de los ingredientes se detallan en la siguiente tabla."

Alimento	Proteína	Grasa	Fibra	Sal
Pollo	0.100	0.080	0.001	0.002
Ternera	0.200	0.100	0.005	0.005
Arroz	0.000	0.010	0.100	0.002
Trigo	0.040	0.010	0.150	0.008

Tabla 1. Datos del problema

Para una mejor comprensión de la tarea por parte del alumnado, se realizó un flujograma donde se mostraba el planteamiento del problema (Figura 4).

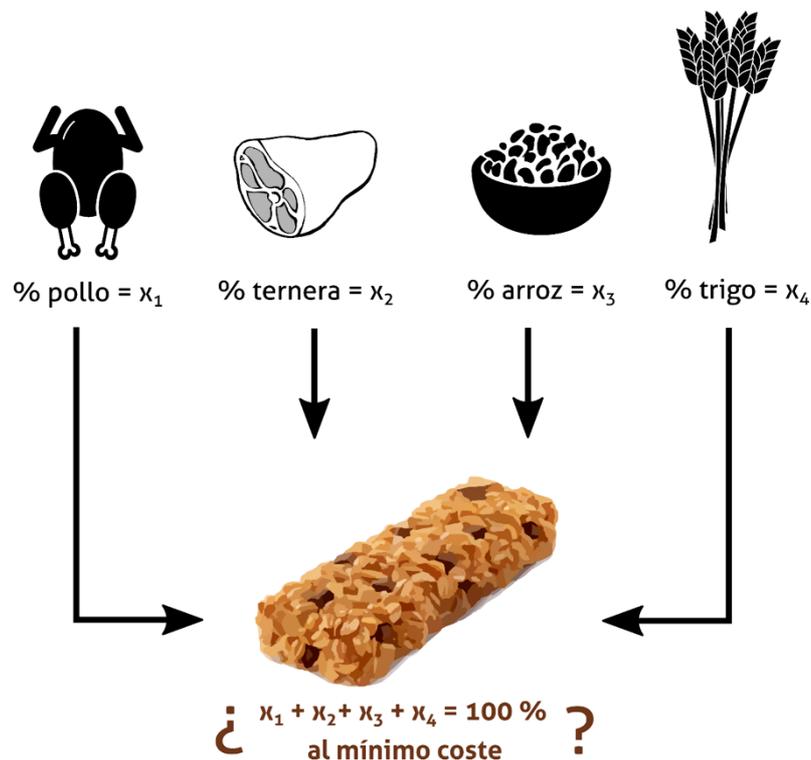


Figura 4: Descripción gráfica del problema

1.3.1.2. Ecuaciones

Los porcentajes han de variar de 0 a 100. Si la ración fuese de 100 g, el porcentaje coincidiría con la cantidad de cada ingrediente en la ración.

- x_1 = porcentaje de carne de pollo en una ración
- x_2 = porcentaje de carne de ternera en una ración
- x_3 = porcentaje de arroz en una ración
- x_4 = porcentaje de trigo en una ración

Función objetivo

El objetivo es minimizar el coste de los ingredientes por ración. Conocido el coste del ingrediente por gramo, para calcular el coste total individual, hay que dividir dicha cantidad por cien y multiplicarla por los gramos en la ración de comida. El resultado de esta operación nos indica los gramos de cada ingrediente que, a su vez, habría que multiplicar por el precio. En el caso de ser la ración de 100 g:

$\min(0.013x_1 + 0.08x_2 + 0.002x_3 + 0.005x_4)$	(1)
--	-----

Restricciones

La suma de todos los porcentajes ha de ser 100.

$x_1 + x_2 + x_3 + x_4 = 100$	(2)
-------------------------------	-----

Para cumplir los análisis nutricionales, se necesita tener al menos 8 g de proteínas, 6 g de grasa, pero no más de 2 g de fibra y 0.4 g de sal. Teniendo en cuenta el porte de cada ingrediente, se obtiene que:

$0.100x_1 + 0.200x_2 + 0.000x_3 + 0.040x_4 \geq 8.0$ $0.080x_1 + 0.100x_2 + 0.010x_3 + 0.010x_4 \geq 6.0$ $0.001x_1 + 0.005x_2 + 0.100x_3 + 0.150x_4 \leq 2.0$ $0.002x_1 + 0.005x_2 + 0.002x_3 + 0.008x_4 \leq 4.0$	(3)
---	-----

1.3.1.3. Entorno iPython

Aunque en Python existen múltiples librerías para resolver problemas de optimización (e.g. CVXOPT (Boyd y Vandenberghe, 2004), PuLP (Mitchell, O'Sullivan, y Dunning, 2011) y Pyomo Hart, Watson, y Woodruff, 2011), con el fin de llevar a cabo estas prácticas se optó por la librería PULP. Para poder utilizar esta librería es necesario instalarla, utilizando, para ello, el comando que aparece en la Figura 3. Este comando se ejecuta introduciéndolo en el propio cuaderno iPython.

```
pip install pulp
```

Figura 3 Comando de instalación de la librería "cvxopt"

En

la

```
In [4]: # problema a resolver creado en librería de optimización PULP
problema = LpProblem("Comida Supervivencia", LpMinimize)

# variables utilizadas por la librería PULP
ingredientes_pulp = LpVariable.dicts("Ingr", ingredientes, 0)

# función objetivo
problema += lpSum([costes[i]*ingredientes_pulp[i] for i in ingredientes]), "costes de los ingredientes por ración"

# restricciones
problema += lpSum([ingredientes_pulp[i] for i in ingredientes]) == 100, "suma porcentajes"
problema += lpSum([porcentaje_proteina[i] * ingredientes_pulp[i] for i in ingredientes]) >= 8.0, "requisitos proteínas"
problema += lpSum([porcentaje_grasa[i] * ingredientes_pulp[i] for i in ingredientes]) >= 6.0, "requisitos grasas"
problema += lpSum([porcentaje_fibra[i] * ingredientes_pulp[i] for i in ingredientes]) <= 2.0, "requisitos fibras"
problema += lpSum([porcentaje_sal[i] * ingredientes_pulp[i] for i in ingredientes]) <= 0.4, "requisitos sal"

# escribir fichero con ecuaciones del problema
problema.writeLP("WhiskasModel.lp")

# resolver problema
problema.solve()

# mostrar variables con valor óptimo
for v in problema.variables():
    print(v.name, "=", v.varValue)

Ingr_ARROZ = 16.498316
Ingr_POLLO = 16.835017
Ingr_TRIGO = 0.0
Ingr_VACA = 66.666667
```

Figura 4 se encuentra el código desarrollado para resolver el problema previamente planteado. En las primera celdas, después de llamar las librerías necesarias (In [1]), se han declarado las variables (In [2]), las restricciones y la función objetivo del problema (In [3]), para en último lugar proceder a su resolución. La enumeración de valores de forma sucesiva es un registro creado por el algoritmo de resolución, en el que no interviene el alumno, cuya finalización, en este caso, se produce después de 6 iteraciones. En la segunda celda (In [4]), se indica la combinación óptima de valores de pollo, ternera, arroz y trigo para minimizar los costes. A partir de este punto, sería labor del docente el pedirle al alumno que experimentase con nuevas restricciones o nuevos valores para las restricciones ya existentes, buscando la familiarización del estudiante con

el proceso de planteamiento y solución de los problemas relacionados con la optimización.

```
In [2]: from pulp import *
```

```
In [3]: # lista de ingredientes disponibles
ingredientes = ['POLLO', 'VACA', 'ARROZ', 'TRIGO']

# costes de los ingredientes
costes = {'POLLO': 0.013,
          'VACA': 0.008,
          'ARROZ': 0.002,
          'TRIGO': 0.005}

# porcentaje de proteína en los ingredientes
porcentaje_proteina = {'POLLO': 0.100,
                       'VACA': 0.200,
                       'ARROZ': 0.000,
                       'TRIGO': 0.040}

# porcentaje de grasa en los ingredientes
porcentaje_grasa = {'POLLO': 0.080,
                    'VACA': 0.100,
                    'ARROZ': 0.010,
                    'TRIGO': 0.010}

# porcentaje de fibra
porcentaje_fibra = {'POLLO': 0.001,
                    'VACA': 0.005,
                    'ARROZ': 0.100,
                    'TRIGO': 0.150}

# porcentaje de sal
porcentaje_sal = {'POLLO': 0.002,
                  'VACA': 0.005,
                  'ARROZ': 0.002,
                  'TRIGO': 0.008}
```

```
In [4]: # problema a resolver creado en librería de optimización PULP
problema = LpProblem("Comida Supervivencia", LpMinimize)

# variables utilizadas por la librería PULP
ingredientes_pulp = LpVariable.dicts("Ingr", ingredientes, 0)

# función objetivo
problema += lpSum((costes[i]*ingredientes_pulp[i] for i in ingredientes)), "costes de los ingredientes por ración"

# restricciones
problema += lpSum([ingredientes_pulp[i] for i in ingredientes]) == 100, "suma porcentajes"
problema += lpSum([porcentaje_proteina[i] * ingredientes_pulp[i] for i in ingredientes]) >= 8.0, "requisitos proteínas"
problema += lpSum([porcentaje_grasa[i] * ingredientes_pulp[i] for i in ingredientes]) >= 6.0, "requisitos grasas"
problema += lpSum([porcentaje_fibra[i] * ingredientes_pulp[i] for i in ingredientes]) <= 2.0, "requisitos fibras"
problema += lpSum([porcentaje_sal[i] * ingredientes_pulp[i] for i in ingredientes]) <= 0.4, "requisitos sal"

# escribir fichero con ecuaciones del problema
problema.writeLP("WhiskasModel.lp")

# resolver problema
problema.solve()

# mostrar variables con valor óptimo
for v in problema.variables():
    print(v.name, "=", v.varValue)

Ingr_ARROZ = 16.498316
Ingr_POLLO = 16.835017
Ingr_TRIGO = 0.0
Ingr_VACA = 66.666667
```

Figura 4 Cuaderno iPython – Planteamiento y resolución del problema de optimización

2. Opinión del alumnado

2.1. Método

2.1.1. Participantes

Participó un grupo de 21 estudiantes de cuarto curso del grado de Ingeniería Mecánica del Centro Universitario de la Defensa de la Escuela Naval Militar.

2.1.2. Procedimiento e instrumento

Los participantes cumplimentaron de forma individual un cuestionario ad-hoc de opinión sobre el uso de nuevas herramientas en el aula

desarrollado utilizando la herramienta formulario de Google. Este cuestionario estaba compuesto por dos preguntas. Las respuestas sobre opinión fueron recogidas a través de una escala de Likert de 5 puntos, donde 1 significaba totalmente en desacuerdo y 5 totalmente de acuerdo.

2.2. Resultados

El análisis de los datos se realizó por medio de una prueba t. En concreto, se ejecutó una comparación de medias con un valor dado (valor de prueba 3: ni de acuerdo, ni en desacuerdo. Valor central de la escala tipo Likert de 5 niveles).

Los resultados del estudio aportaron evidencia de la buena aceptación de esta nueva metodología de trabajo en el aula (Tabla 2), como resultado, se rechazaron las hipótesis nulas: 1) Los estudiantes no se muestran ni de acuerdo ni en desacuerdo en que las herramientas empleadas en la asignatura tienen un alto potencial didáctico y 2) los estudiantes no se muestran ni de acuerdo ni en desacuerdo en que los problemas realizados en la asignatura creo que me son útiles. Esto se dedujo de analizar el coeficiente de significatividad asintótica bilateral, que, en ambos casos, es menor de .05 y el valor medio de respuesta en ambas preguntas; superior en ambos casos al valor central, lo que indica que los estudiantes están de acuerdo.

Ítems	M	DT	t	IC 95%
Los problemas realizados en la asignatura creo que me son útiles	3.52	.00	2.75*	[1.09, 1.00]
La herramientas empleadas en la asignatura tienen un alto potencial didáctico	3.38	.00	2.02*	[-.01, .00]
Nota. *p< .05 gl(20)				

Tabla 2. Comparación de medias con un valor dado (valor de prueba: 3, ni de acuerdo, ni en desacuerdo).

3. CONCLUSIONES

Las nuevas corrientes en materia educativa introducen cambios en el papel desempeñado tradicionalmente por el profesor. En lo sucesivo, el docente se convierte en un orientador dentro del proceso de aprendizaje de los discentes. Asimismo, la exigencia en el desarrollo de competencias para los estudiantes se traduce, también, en nuevas competencias para los docentes. En el contexto actual, la elaboración colaborativa de recursos didácticos se convierte en un instrumento de gran potencial para el desarrollo de nuevas técnicas docentes. La innovación debe ser entendida como la herramienta clave para la mejora de las prácticas educativas, construyendo proyectos a través del uso de modelos y técnicas acordes con el contexto. Los cuadernos iPython, por su carácter

flexible, pueden ser considerados como un medio adecuado para el desarrollo de estrategias innovadoras en el ámbito del desarrollo curricular. Por otra parte, esta herramienta tiene la gran ventaja de que resulta accesible fuera del aula y, como consecuencia, permite interactuar a los usuarios en su tiempo libre, además de facilitar el aprendizaje autónomo.

El uso de cuadernos iPython tiene una leve curvatura de aprendizaje si lo único que se desea es introducir contenido multimedia. Sin embargo, para aprovechar plenamente su potencial, utilizándolos como laboratorios virtuales e implementar e-experimentos, se requieren nociones de Python. Esto último puede parecer un escollo para quienes no estén familiarizados con la programación. No obstante, este obstáculo puede ser superado de dos maneras: aprendiendo a programar gracias a la enorme cantidad de recursos disponible en la Web y la comunidad de usuarios deseosos de ayudar, o reutilizando los cuadernos ya existentes con licencia de dominio público. En esta línea, la progresiva formación del profesorado en este tipo de tecnologías para la docencia representa un aspecto clave, si queremos obtener el máximo rendimiento. Este último aspecto se encuentra interrelacionado con el tiempo extra que los docentes tendrán que dedicar a preparar los diferentes materiales didácticos. Además, de la opinión de los estudiantes se desprende que el uso de nuevas herramientas resulta de gran utilidad.

4. BIBLIOGRAFÍA

- Alcántara, M. D. (2009). Importancia de las TIC para la educación. *Innovación y experiencias educativas*, 15, 200-220.
- Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.
- Continuum Analytics (2015). Download Anaconda Python Distribution. [Online]. Available: <http://continuum.io/downloads>. [Último acceso: 27-Nov-2015].
- de Miguel, M. (2006). *Metodologías de enseñanza y aprendizaje para el desarrollo de competencias: orientaciones para el profesorado universitario ante el Espacio Europeo de Educación Superior*. Madrid: Alianza editorial.
- Entought Scientific Computing Solutions (2015). Python Distribution and Integrated Analysis Environment | Enthought Canopy. [Online]. Available: <https://www.enthought.com/products/canopy/>. [Último acceso: 27-Nov-2015].
- Hart, W. E., Watson, J. P., & Woodruff, D. L. (2011). Pyomo: modeling and solving mathematical programs in Python. *Mathematical Programming Computation*, 3(3), 219-260.
- IEEE Spectrum (2015), The 2015 Top Ten Programming Languages. [Online]. Available: <http://spectrum.ieee.org/computing/software/the-2015-top-ten-programming-languages/>. [Último acceso: 24-Nov-2015].
- Macmillan Publishers (2015). Impact Factors for journals published by Nature Publishing Group. [Online]. Available:

- http://www.nature.com/npg_/company_info/impact_factors.html. [Último acceso: 24-Nov-2015].
- Mitchell, S., O'Sullivan, M., & Dunning, I. (2011). PuLP: a linear programming toolkit for python. Sep-2011.
- Nbviewer (2015) - A simple way to share Jupyter Notebooks. [Online]. Available: <http://nbviewer.ipython.org/>. [Último acceso: 26-Jul-2015].
- Notebook Gallery (2015). [Online]. Available: <http://nb.bianp.net/sort/views/>. [Último acceso: 26-Jul-2015].
- Pensosi, V. (2010). Comunicación personalizada: iniciativas de la Fundación Orange en comunicación aumentativa y tecnología. En M.I. Gómez, y M. Gómez, (Coords.), *Puentes para la comunicación interpersonal* (pp. 7-20). Murcia: Consejería de Educación, Formación y Empleo de la Región de Murcia.
- Perkel, J. M. (2015). Programming: pick up python. *Nature*, 518(7537), 125-126.
- Python Development Team (2015), The Jupyter Notebook Format — IPython 3.2.0 documentation. [Online]. Available: <https://ipython.org/ipython-doc/3/notebook/nbformat.html>. [Último acceso: 26-Nov-2015].
- Python Development Team (2015). Sponsors — IPython. [Online]. Available: <http://ipython.org/sponsors.html>. [Último acceso: 26-Nov-2015].
- Selwyn, N. y Gouseti, A. (2009). School & Web 2.0: A Critical Perspective. *Educatio Siglo XXI*, 27 (2), 147-165.
- Shen, H. (2014). Interactive notebooks: Sharing the code. *Nature*, 515(7525), 151-152.
- Winocor, R. (2007). Nuevas tecnología y usuarios. La apropiación de las TIC en la vida cotidiana. *Telos. Cuadernos de Comunicación e Innovación*, 73, 109-117.
- Wittenauer, J. (2015). A gallery of interesting IPython Notebooks. [Online]. Available: <https://github.com/ipython/ipython/wiki/A-gallery-of-interesting-IPython-Notebooks>. [Último acceso: 26-Jul-2015].