

# El método científico en la era de los ordenadores

C. León, G. Miranda, C. Rodríguez, E. Segredo

Dpto. de Ingeniería Informática y de Sistemas

Universidad de La Laguna

38271 San Cristobal de La Laguna, Santa Cruz de Tenerife

(cleon|gmiranda|crguezl|esegreso)@ull.es

Carlos Segura

Area de Ciencias de la Computación

Centro de Investigación en Matemáticas

36240 Guanajuato, México

carlos.segura@cimat.mx

## Resumen

En los estudios de grado de las ramas de Ciencias e Ingeniería se suele incluir una asignatura denominada Técnicas Experimentales en la que se introducen los mecanismos de funcionamiento de los aparatos o herramientas específicas que se van a utilizar en la especialidad. Generalmente en ella se expone el método científico adaptado a cada campo particular, se estudian las unidades de medida, las representaciones gráficas y la preparación de informes escritos y orales. En este trabajo se hace una propuesta de contenidos para una asignatura con la misma denominación, en la que la herramienta a utilizar sea una computadora y se aborde la resolución de problemas a través del pensamiento computacional permitiendo la aplicación del método científico en cualquier rama de conocimiento.

## Abstract

Science and Engineering undergraduate studies usually include an academic subject (to which we will refer from now on as Experimental Techniques) whose goal is to introduce the students in the operation of the devices and tools that they will use along their degree. The course exposes an adaptation of the scientific method to the specific field of the degree and it studies the measurement units, graphic representations and the preparation of written and oral reports. This paper presents a proposal of contents for a version of such a course in which the particular tool to use is the computer and the adapted version of the scientific method is the ability to solve problems through computational thinking with the final aim of leading the student to the application of the scientific method to any branch of knowledge.

## Palabras clave

Pensamiento computacional, resolución de problemas, elaboración de informes, presentación de informes.

## 1. Motivación y objetivo

Según el diccionario de la Real Academia Española<sup>1</sup>, la *ciencia* es el “Conjunto de conocimientos obtenidos mediante la observación y el razonamiento, sistemáticamente estructurados y de los que se deducen principios y leyes generales”. A esta cantidad de información que los humanos han ido adquiriendo sobre la Naturaleza y sobre sí mismos se le denomina *conocimiento científico*, en contraposición a otro tipo de conocimientos no científicos como pueden ser el sentido común (refranes: no dejes para mañana lo que puedas hacer hoy), la religión (origen de los seres humanos...) o la magia (hechizos...). El conocimiento científico se puede definir en relación con sus objetivos y con el modo en el que éstos se tratan de alcanzar. Se adquiere utilizando un método estandarizado: el *método científico*, cuya característica principal es la replicabilidad, lo que permite el consenso dentro de la comunidad científica. Según Feynman [2] “El método científico consiste en observación, razonamiento y experimentación”. El método científico no es un procedimiento único, es más bien una perspectiva filosófica que explora muchas cosas. Según esta filosofía, los pasos a seguir para abordar la resolución de un problema incluyen:

1. Identificar el problema mediante la observación y modelizarlo.
2. Hacer una conjetura razonable, una hipótesis, y predecir las consecuencias.
3. Experimentar, esto es, poner a prueba las predicciones.
4. Formular la regla general más simple que aglutine la hipótesis, la predicción y el resultado experimental.
5. Elaborar un informe del proceso y los resultados.

Esta filosofía de trabajo, se ha de adaptar al uso intensivo de ordenadores. El *pensamiento computacional* [8] es un proceso de solución de problemas que permite su formulación de manera que se puedan usar ordenadores para solucionarlos. Esto conlleva una organización de los datos de manera lógica para po-

<sup>1</sup><http://www.rae.es/drae/>

der analizarlos y representarlos mediante abstracciones. Además, mediante una serie de pasos ordenados (*pensamiento algorítmico*) se ha de automatizar la obtención de soluciones haciendo uso de los recursos de forma efectiva y eficiente. Este trabajo, requiere de una serie de disposiciones o actitudes tales como: la confianza en la manipulación de la complejidad, la persistencia al trabajar con problemas difíciles, la tolerancia a la ambigüedad, la habilidad para lidiar con problemas no estructurados y finalmente la habilidad para comunicarse y trabajar con otros para alcanzar una meta o solución común.

*Internet* juega un papel fundamental en la comunicación y el trabajo en equipo. De hecho, dejó de ser un sitio donde se consulta información o se intercambian mensajes, para pasar a ser un espacio abierto en el que todos pueden participar. Esto es lo que ha venido a denominarse *Web 2.0* o *Web Social* [3]. Vinculada a ella surgen las denominadas *Herramientas colaborativas*, que permiten pasar de ser un mero receptor de información a ser partícipe de esa información, bien creándola, compartiéndola o mejorándola a través de redes de colaboración. Esta nueva capacidad de participación permite facilitar el trabajo colaborativo en cualquier ámbito.

El pensamiento computacional debería ser una competencia básica en todos los ámbitos de la ciencia. En [5] se hace una propuesta para su inclusión en primaria y secundaria. Teniendo en cuenta que esta filosofía promueve el desarrollo de la habilidad para comunicarse y trabajar con otros para alcanzar una meta o solución común es fundamental que se utilicen herramientas colaborativas para sacar el mejor rendimiento posible a Internet. En este trabajo se hace una propuesta de contenidos para una asignatura denominada *Técnicas Experimentales* cuyo objetivo es el desarrollo y la consolidación del *pensamiento computacional* a través del uso de *herramientas colaborativas de software libre*. La ubicación de la asignatura en un plan de estudio dependerá de la rama de conocimiento y el nivel de los mismos.

El resto del presente trabajo se organiza tal y como sigue: en la sección 2 se describen los contenidos, las competencias y los resultados de aprendizaje esperados. A continuación, en la sección 3 se relacionan las herramientas informáticas seleccionadas así como algunos de los recursos docentes desarrollados y se profundiza en algunos a modo de ejemplo. Finalmente, se enumeran las conclusiones.

## 2. Contenidos

El objetivo principal de la asignatura *Técnicas Experimentales* es proporcionar al alumnado los medios necesarios para el desarrollo y validación del conoci-

miento científico mediante la experimentación, motivándolo e introduciéndolo en las herramientas informáticas asociadas. Por ello, durante su evolución, el profesorado ha de poner especial énfasis en dar a entender que el *aula de informática* debe ser considerada como cualquier otro tipo de *laboratorio* en el cual se llevan a cabo experimentos prácticos mediante la instrumentación presente en el mismo, es decir, mediante el hardware y el software disponibles. El carácter de los contenidos es transversal en el sentido de que los conocimientos adquiridos pueden emplearse en cualquier otra asignatura del plan de estudios en la que sea necesaria la realización de trabajos prácticos con ordenadores, así como en el ejercicio posterior de la profesión.

Para que la propuesta esté acorde con el Espacio Europeo de Educación Superior se plantea en base al trabajo de competencias. Concretamente se sugiere que se cubran las siguientes competencias específicas:

- Utilizar aplicaciones informáticas de análisis estadístico, visualización gráfica, optimización u otras para experimentar y resolver problemas.
- Desarrollar programas que resuelvan problemas utilizando para cada caso el entorno computacional adecuado.

Y las competencias transversales:

- Utilizar herramientas de búsqueda de recursos bibliográficos.
- Comunicar, tanto por escrito como de forma oral, conocimientos, procedimientos, resultados e ideas.
- Saber trabajar en equipo.

Como consecuencia de las competencias trabajadas durante la impartición de la asignatura se pretende que entre los resultados de aprendizaje por parte del alumnado se encuentren las capacidades de:

- Analizar, sintetizar, evaluar y describir información y datos científicos.
- Medir magnitudes esenciales.
- Evaluar y analizar cuantitativamente los resultados experimentales.
- Representar gráficamente resultados experimentales.
- Proponer informes científicos sintetizando los resultados de una experiencia.
- Utilizar herramientas informáticas en el contexto adecuado.
- Programar en un lenguaje de programación relevante para el cálculo científico.
- Argumentar con rigor científico conceptos, problemas y experimentos.

Se ha realizado un estudio de las asignaturas impartidas en las titulaciones de Grado a nivel nacional, bus-

cando aquellas que se ajusten a las competencias y los resultados de aprendizaje descritos en los párrafos anteriores. En la mayoría de los casos se encuentran asignaturas que enfocan sus contenidos al estudio de un lenguaje de programación concreto, al diseño y análisis de algoritmos, al uso de herramientas de cálculo numérico y/o simbólico, al estudio de paquetes gráficos, a la optimización o a la investigación operativa, entre otros, pero siempre de forma individual. La mayor carencia detectada en los contenidos de dichas asignaturas es que la competencia de comunicación oral y escrita de conocimientos, procedimientos, resultados e ideas no está contemplada – o por lo menos no se indica explícitamente – en sus contenidos. Esta competencia es muy importante porque permite difundir el conocimiento adquirido durante el desarrollo de una experiencia de manera rigurosa y debería trabajarse con especial énfasis desde los primeros cursos de cualquier Grado.

Los contenidos de la asignatura Técnicas Experimentales se organizan de la siguiente manera:

1. Realización de prácticas de observación y experimentación en el laboratorio.
2. Realización de informes.
3. Técnicas de demostración matemática.
4. Concepto de medida. Sistemas de unidades. Concepto de error y su tratamiento.
5. Representación de resultados.

En el primer tema se aborda la resolución de problemas, respondiendo a preguntas como: ¿Qué es un problema?, ¿Cuáles son las fases del desarrollo de software?, ¿Qué son los lenguajes de programación?, ¿Por qué estudiar lenguajes de programación?, ¿Qué diferencia hay entre compilación e interpretación?. A continuación se pasa a estudiar la sintaxis y la semántica de un lenguaje de programación científica revisando los conceptos de tipos de datos, su abstracción, flujo de control, subrutinas, operaciones con ficheros y creación de bibliotecas.

La experiencia demuestra que no es fácil escribir literatura científica si no se está acostumbrado a ello. En el segundo tema se aborda la realización de informes dejando claro que es cuestión de práctica el lograr una redacción clara y precisa. Se describen las secciones con las que ha de contar un informe: Un primer capítulo de motivación y objetivos, en segundo lugar se han de desarrollar los fundamentos teóricos, el tercer epígrafe se ha de dedicar al procedimiento experimental y finalmente se han de enumerar las conclusiones. También, se muestra cómo realizar búsquedas bibliográficas en los diferentes catálogos y bases de datos contrastadas disponibles para la comunidad universitaria y cómo citarlas de forma adecuada. Además, se

hace especial hincapié en el mantenimiento de una carpeta o directorio de proyecto.

Teniendo como objetivo el desarrollo de las habilidades mínimas que permitan elaborar una fundamentación teórica, en el tercer tema, se introducen algunas técnicas de demostración matemática simples: directa, indirecta, por contraposición, por reducción al absurdo, por inducción, por distinción de casos y por contraejemplo.

En el cuarto tema se estudian las unidades de información en los sistemas digitales, la conversión entre unidades, las representaciones de números enteros, números reales y caracteres, los conceptos de error, medida, exactitud, precisión y resolución.

Finalmente en el quinto tema se aborda la representación gráfica de los resultados obtenidos.

Estos contenidos se pueden ofertar como una asignatura de 6 créditos ECTS (*European Credit Transfer System*) distribuidos en 16 semanas. En cada una de ellas, el alumnado debe asistir de manera presencial al aula de teoría y/o al laboratorio de prácticas durante 3 o 4 horas, dependiendo de la semana que se trate. Además, el alumnado debe trabajar de manera autónoma durante 6 horas cada semana. Esto produce unas 9 o 10 horas de trabajo total dependiendo de la semana. Por lo tanto, la asignatura consta de 60 horas de trabajo presencial y 90 horas de trabajo autónomo, cumpliendo con las 150 horas de los 6 créditos.

Establecidos los contenidos y su organización temporal, se han de seleccionar las herramientas a utilizar para el desarrollo de las prácticas de laboratorio. Es decir, a la hora de implantar la asignatura se ha de tener en cuenta el área de conocimiento a la que va enfocada para elegir adecuadamente qué lenguaje de programación y herramientas utilizar. Una vez hecha la elección, se han de elaborar ejercicios guiados y ejercicios autónomos que permitan la adquisición de las competencias. En los párrafos siguientes se describen los recursos docentes elaborados para la implantación de la asignatura en el grado en Matemáticas.

### 3. Recursos docentes

Con los ejercicios guiados y autónomos de prácticas de laboratorio, se pretende introducir al alumnado en la metodología de desarrollo de software de código abierto, por ello, se propone utilizar: el sistema de control de versiones Git [1], los sistemas de hospedaje en la nube para el trabajo en grupo GitHub<sup>2</sup> y Bitbucket<sup>3</sup>, el lenguaje de programación de alto nivel Python [7], la biblioteca SciPy<sup>4</sup> para el cálculo científico, para la

<sup>2</sup><https://github.com/>

<sup>3</sup><https://bitbucket.org/>

<sup>4</sup><http://www.scipy.org/>

Número	Contenido	Tipo de recurso
1	Fuentes de información para los estudios universitarios	Ejercicios guiados
2	Entorno Linux y manipulación de una consola	Ejercicios guiados
3	Sistema de control de versiones Git (Github)	Ejercicios guiados
4	Compilación vs. interpretación y primeros pasos en Python	Ejercicios guiados
5	Sentencias y funciones Python	Ejercicios guiados
6	Aproximación del número $\pi$ con una máquina de cómputo	Ejercicio autónomo
7	Función Python para la aproximación del número $\pi$	Ejercicio autónomo
8	Módulo Python para la aproximación del número $\pi$	Ejercicio autónomo
9	Umbral de error en la aproximación del número $\pi$	Ejercicio autónomo
10	Procesamiento de texto con $\LaTeX$	Ejercicios guiados
11	Informe sobre $\pi$ usando $\LaTeX$	Ejercicio autónomo
12	Evaluación de la eficiencia	Ejercicio autónomo
13	Bibliotecas científica y visualización	Ejercicios guiados
14	Presentación sobre la eficiencia de la aproximación de $\pi$ usando BEAMER	Ejercicio autónomo

Cuadro 1: Prácticas de Laboratorio.

representación gráfica de datos Mathplotlib<sup>5</sup>, para la realización de informes se propone el procesamiento de texto con  $\LaTeX$  [4] y para la realización de presentaciones el paquete BEAMER [6].

Si se quiere gestionar de manera eficiente una asignatura con un alto número de alumnos matriculados y varios profesores se puede conseguir la sincronización y colaboración por parte del profesorado mediante reuniones presenciales, el entorno institucional para la docencia virtual basado en Moodle<sup>6</sup>, así como con herramientas en la nube, fundamentalmente: Gmail, Google Calendar, Google Drive y Hangouts.

El Cuadro 1 muestra una relación de prácticas de laboratorio y los contenidos tratados en cada una de ellas. En la última columna se especifica el tipo de recurso indicando si se trata de ejercicios guiados o de un ejercicio para realizar de forma autónoma.

Nótese que se introduce el uso de la herramienta para el control de versiones Git. El *control de versiones* es un sistema que registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo, de modo que se puedan recuperar versiones específicas más adelante. Cualquier tipo de archivo que se encuentre en un ordenador se puede poner bajo control de versiones. Esto permite revertir archivos a un estado anterior, revertir el proyecto entero a un estado anterior, comparar cambios a lo largo del tiempo, ver quién modificó por última vez algo que puede estar causando un problema, quién introdujo un error y cuándo. Este tipo de herramientas es fundamental para el trabajo en equipo. Los ejercicios prácticos de laboratorio se llevarán a cabo en equipos con un mínimo de dos personas. Puesto que el objetivo es el desarrollo de un módulo, es decir, una biblioteca en Python, se va a programar

código fuente y el mismo tiene que ser compartido por los miembros del equipo. Así mismo, también se han de compartir los ficheros  $\LaTeX$  y BEAMER del informe y la presentación. Haciendo uso de Git, los miembros del equipo deben ir almacenando las diferentes versiones de todo el material desarrollado para las prácticas y compartirlo a través de un repositorio remoto situado en GitHub o en Bitbucket. De esta forma todos los miembros del equipo tienen disponible la última versión del material elaborado para su descarga. Además, en el repositorio remoto hay que dar de alta como colaboradores al profesorado de la asignatura de manera que puedan acceder al mismo para realizar las labores de seguimiento y de evaluación. Se ha programado una sesión de ejercicios guiados para introducir el uso y configuración de Git y para la creación y compartición de cuentas en GitHub y Bitbucket. En la Figura 1 se muestra la primera página del enunciado.

La cuarta sesión de prácticas consiste en una batería de ejercicios guiados para dar los primeros pasos en Python. En ella se utiliza el intérprete interactivo de Python para jugar con cadenas, rangos, símbolos, listas y diccionarios. La programación imperativa es el modelo más antiguo y aunque en Python se pueda programar Orientado a Objetos es posible utilizarlo en la quinta sesión de prácticas para introducir la sintaxis y la semántica de las sentencias condicionales y las iterativas. En la quinta sesión se introducen las sentencias que modifican el flujo de control y las funciones.

En el pensamiento computacional se encuentra el *pensamiento algorítmico* que consiste en diseñar una serie de pasos ordenados que permitan automatizar la obtención de soluciones a un problema dado, haciendo uso de los recursos de forma efectiva y eficiente. Esta es la metodología que se emplea en la propuesta de asignatura, concretamente la técnica denominada

<sup>5</sup><http://matplotlib.org/>

<sup>6</sup><https://moodle.org/>

**Técnicas Experimentales**  
Práctica de laboratorio #3

1. Iniciar una sesión de trabajo en GNU-Linux.
2. Abra una terminal.
3. Muestre el árbol de directorios de su HOME (`tree`).
4. Compruebe si existe el directorio `.ssh` (`cd ~/.ssh`).
5. Si la respuesta es “No existe tal directorio” continúe por el ejercicio 8.
6. Si la respuesta es afirmativa, cree un directorio con nombre `copia` dentro del directorio `.ssh` (`mkdir ~/.ssh/copia`).
7. Mueva la pareja de clave-pública clave-privada al directorio `copia` (`mv ~/.ssh/id_rsa* ~/.ssh/copia/`).
8. Genere una nueva pareja de clave-pública clave-privada en el directorio `.ssh` (`ssh-keygen -t rsa`). Para usar las opciones por defecto, a cada pregunta responda pulsando la tecla de retorno de carro.
9. ¿Cuáles son las funcionalidades del comando `git`?
10. Configure `git` con el **nombre de usuario** de manera que pueda etiquetar de forma correcta las actualizaciones que este realice. (`git config --global user.name "Nombre Apellido"` )
11. Configure `git` con la dirección de correo electrónico para asociarla a las actualizaciones que se hagan en el repositorio `git`. (`git config --global user.email "aluXXXXXXXXX@edu.es"` )
12. `git` permite almacenar la configuración global de un usuario en el archivo `.gitconfig`. Este archivo se encuentra en el directorio HOME del usuario. `git` almacena el remitente y el nombre del autor de un cambio en cada registro en el repositorio. Esta información se puede almacenar en el fichero de configuración global de manera que no se solicite cada vez que se haga un registro. Muestre el contenido del fichero `.gitconfig` (`cat .gitconfig`)
13. Muestre las configuraciones globales de `git`. (`git config --list` )
14. Sitúese en la **Carpeta de Proyecto** de la asignatura Técnicas Experimentales esto es en el directorio `TE` (`cd TE`).
15. Muestre el contenido del directorio actual (`ls -la`).
16. Cree un nuevo directorio denominado `prct03` (`mkdir prct03`). Este será el directorio actual durante la realización de esta práctica.
17. Sitúese en el directorio `prct03` y cree la estructura de directorios que le permita tener subcarpetas para el código y los documentos, es decir:
  - un subdirectorio `src`
  - un subdirectorio `docs`
18. Guarde el fichero PDF que contiene el enunciado de esta práctica en el directorio `docs`.
19. Situado en el directorio actual, inicialícelo para que sea un repositorio `git`. (`git init` )
20. Compruebe que se crea el directorio `.git` (`ls -la`).

Figura 1: Ejercicio guiado: Sistema de control de versiones Git.

*divide y vencerás*. Este modelo de resolución de problemas consiste en dividir el problema original en subproblemas, hasta que se llegue a una definición que sea lo suficientemente simple de resolver, de manera que, la conjunción de todas las soluciones parciales obtenidas proporcione la solución al problema original. Aplicando esta técnica a la organización de los ejercicios de laboratorio, en la sexta y séptima sesión de prácticas se han de aplicar los conocimientos adquiridos a un problema simple de aproximación del número  $\pi$ . Se plantea lo siguiente:

A lo largo de la historia han sido muchas las formas utilizadas por el ser humano para calcular aproximaciones cada vez más exactas del número  $\pi$ . El número  $\pi$  es el cociente entre la longitud de una circunferencia cualquiera y el diámetro de la misma. Ludolph van Ceulen (1540-1610), matemático alemán profesor en la Universidad de Leiden en Holanda, se pasó buena parte de su vida calculando los primeros 35 decimales de  $\pi$  y su tumba lo refleja en el hecho de tener grabadas las 35 cifras: 3.14159265358979323846264338327950288.

El objetivo de esta práctica de laboratorio es implementar el código Python que permita aproximar el número  $\pi$  con una cierta precisión.  $\pi$  se puede calcular mediante integración:

$$\int_0^1 \frac{4}{1+x^2} dx = 4(\text{atan}(1) - \text{atan}(0)) = \pi$$

Esta integral se puede aproximar numéricamente con una fórmula de cuadratura. Si se utiliza la regla del punto medio se obtiene:

$$\pi \approx \frac{1}{n} \sum_{i=1}^n f(x_i), \text{ con } f(x) = \frac{4}{(1+x^2)},$$

$$x_i = \frac{i-\frac{1}{2}}{n}, \text{ para } i = 1, \dots, n$$

Escriba un programa que reciba como entrada el número de subintervalos con los que se desea abordar la aproximación de  $\pi$ . A partir de él se debe calcular y mostrar por la consola el resultado de la aproximación.

Un mayor número de intervalos hace mayor la precisión en los decimales de la aproximación de  $\pi$ . Por lo tanto, se concluye que  $n$  ha de constituir un *parámetro* de la función Python que aproxime  $\pi$ . En la Figura 2 se muestra gráficamente utilizando  $n = 4$ .

Todos los lenguajes de programación tienen soporte para las bibliotecas. Las bibliotecas son colecciones

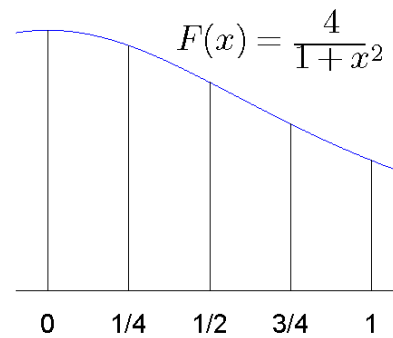


Figura 2: Aproximación de  $\pi$  numéricamente.

de código reutilizable que suelen agruparse en torno a un propósito común. Se pueden encontrar bibliotecas para casi cualquier cosa, desde el procesamiento avanzado de cadenas a funciones matemáticas de nivel superior. La integración de herramientas y plataformas se ha convertido en un estándar. La mayoría de los sistemas operativos, aplicaciones y herramientas permiten incorporar o quitar funcionalidades a partir de bibliotecas que se distribuyen a través de Internet. Un módulo Python es esencialmente un directorio con una estructura especial que contiene el código fuente, la descripción, la documentación y las pruebas de una biblioteca Python. En la octava sesión de prácticas de laboratorio se aborda la creación de un módulo partiendo del código desarrollado. Usando el módulo se realizan experimentos para establecer un umbral de error en la aproximación de  $\pi$ , así como la evaluación de la eficiencia del código desarrollado. Finalmente se ha de elaborar un informe y una presentación de los resultados obtenidos, así como una representación gráfica de los resultados. En la Figura 3 se muestra el enunciado del ejercicio autónomo número 11. Puesto que este ejercicio autónomo se ha de tener en cuenta para la evaluación la asignatura, en el enunciado del mismo, se especifica la competencia que se trabaja y se establecen los criterios con los que se evaluará el mismo.

## 4. Conclusiones

En este trabajo se hace una propuesta de contenidos para una asignatura denominada Técnicas Experimentales, en la que se aborda la resolución de problemas a través del pensamiento computacional, esto es, teniendo en cuenta que la herramienta a utilizar sea un ordenador. Se presenta un ejemplo de contenidos Matemáticos, pero se puede adaptar a cualquier rama de conocimiento porque se basa en la aplicación del método científico. Se hace especial hincapié en el tratamiento de las competencias transversales de expresión oral y escrita y trabajo en equipo.

## Informe sobre $\pi$ usando $\LaTeX$

Técnicas Experimentales  
Práctica de Laboratorio #11

20 de febrero de 2015

### Resumen

El objetivo de esta práctica es entregar un artículo escrito en  $\LaTeX$ . El artículo versará sobre el número  $\pi$ .

## 1. Motivación y Objetivos

En la asignatura *Técnicas Experimentales* se ha de desarrollar la competencia transversal: *Comunicar, tanto por escrito como de forma oral, conocimientos, procedimientos, resultados e ideas*. Con este ejercicio de laboratorio se pretende sintetizar las habilidades adquiridas en la *comunicación escrita*. Se valorará como resultado de aprendizaje la claridad, la precisión y el rigor científico en la exposición de los conceptos. El objetivo de esta práctica es entregar un artículo escrito en  $\LaTeX$ . El artículo versará sobre el número  $\pi$ .

## 2. Ejercicios propuestos

### 2.1. Creación de una carpeta de proyecto

Se ha de crear un directorio dentro de la carpeta de proyecto de la asignatura para almacenar los archivos que se generen con el mismo.

### 2.2. Creación de un informe

Se ha de elaborar un documento en formato PDF, cuyo tema principal sea el número  $\pi$ . Ha de contar con lo siguiente:

- Título, autor, fecha, resumen
- Secciones (mínimo dos)
- Subsecciones (mínimo dos)
- Pies de página<sup>1</sup>
- Gráficos (mínimo uno)
- Referencias a Gráficos (mínimo una)
- Tablas (mínimo uno)
- Referencias a tablas (mínimo una)
- Bibliografía (como mínimo dos)
- Citas a la bibliografía (mínimo dos)

### 2.3. Entregable

En la tarea habilitada para esta práctica en el Aula Virtual, se subirán dos cosas:

1. La dirección del repositorio *github* donde se ha almacenado la práctica.
2. El artículo solicitado en formato PDF

---

<sup>1</sup>mínimo uno

La acción innovadora de la propuesta se enmarca dentro de las metodologías y estrategias docentes. La principal aportación es la aproximación al método científico haciendo uso del pensamiento computacional. Así mismo, es novedoso el especial énfasis realizado en la introducción de los elementos necesarios para la aplicación de metodologías ágiles para el desarrollo de software como son: el uso de sistemas de control de versiones, el desarrollo dirigido por pruebas y la elaboración de documentación.

Se ha diseñado una batería de ejercicios prácticos de laboratorio y elaborado un material reutilizable para cubrir las competencias de la asignatura. Cada una de las prácticas de laboratorio se asocia con una o varias herramientas. Concretamente, se utilizan: Git para el control de versiones distribuido, SciPy para la implementación de funciones de cálculo, y Matplotlib para las representaciones gráficas. El material elaborado también incluye un Aula Virtual Moodle hospedada en el campus virtual institucional. A través de ella es posible gestionar de manera eficiente la comunicación con el alumnado y mejorar el proceso de coordinación entre el profesorado.

Agradecemos desde estas líneas el soporte del Vicerrectorado de Calidad Institucional e Innovación Educativa dentro del programa de Proyectos de Innovación Docente.

También al Vicerrectorado de Tecnologías de la Información y las Comunicaciones dentro del programa de Apoyo a la Docencia Presencial en la modalidad de “Apoyo a la docencia presencial de asignatura y al trabajo autónomo del alumnado”.

## Referencias

- [1] Scott Chacon and Ben Straub. Pro Git. Apress, segunda edición, 2014. disponible en <https://progit.org/>.
- [2] Richard Feynman. *¿Está usted de broma, Sr. Feynman?*. Alianza Editorial, 2003.
- [3] Antonio Fumero y Genís Roca. *Web 2.0*. Fundación Orange España, mayo de 2007.
- [4] Leslie Lamport. *TEX: A Document Preparation System*. Addison–Wesley Pub. Co., Reading, MA, segunda edición, 1994.
- [5] M. Riesco, M. Fondón, D. Álvarez, B. López, A. Cernuda, A. Juan. La Informática como materia fundamental en un sistema educativo del siglo XXI. En *Actas de las XX JENUI*. Págs. 27-32. Oviedo, 9-11 de julio 2014.
- [6] T. Tantau, J. Wright, V. Miletic. The Beamer class. User Guide for version 3.31. CTAN, 2013. disponible en <https://bitbucket.org/rivanvx/beamer/wiki/Home>.
- [7] Guido van Rossum. Python Programming Language. In *Proceedings of the 2007 USENIX Annual Technical Conference*, June 17-22, 2007, Santa Clara, CA, USA.
- [8] J. M. Wing. Computational Thinking. *Communications of the ACM, CACM*, 49(3):33–35, marzo 2006.