

JUPYTER NOTEBOOKS COMO HERRAMIENTA VERSÁTIL PARA DOCENCIA EN CURSOS STEM

Sebastián Flores
U-Planner, Universidad Técnica Federico Santa María
sebastian.flores@u-planner.com

Christopher Cooper
Universidad Técnica Federico Santa María
christopher.cooper@usm.cl

RESUMEN

Recientes estudios sobre la enseñanza de cursos tipo STEM indican que es necesario un cambio en lo que ocurre en el aula, donde la clase está centrada en el profesor, y poner el foco en el estudiante. Esto ha obligado la comunidad educativa a buscar herramientas metodológicas que permitan motivar la participación activa del estudiante en el aula, y donde la utilización de notebooks de jupyter surge como una estrategia viable para mejorar y complementar las metodologías docentes. Este trabajo tiene como objetivo dar a conocer la experiencia de algunos profesores de la Universidad Técnica Federico Santa María con esta herramienta y realizar una invitación a usar y colaborar con esta iniciativa. Para ello repasa las diferentes formas de uso de los jupyter notebooks en el aula, presenta material desarrollado y disponible online y realiza recomendaciones de uso y buenas prácticas. Jupyter notebooks constituye una herramienta que facilita el aprendizaje de manera virtual o descentralizada, para la cual existen activas comunidades de enseñanza-aprendizaje para una amplia gama de contenidos relacionados con matemática, ciencia e ingeniería.

PALABRAS CLAVES: Metodologías Docentes, Herramientas de enseñanza-aprendizaje, python, enseñanza de programación.

INTRODUCCIÓN

El proyecto jupyter constituye la evolución de un proyecto durante 15 años, y es un excelente representante de la cultura open source. En el año 2001, el estudiante de la Universidad de Colorado Boulder, Fernando Perez, decidió proporcionar al lenguaje de python un ambiente de programación interactivo, como los que ya existían para Maple y Mathematica, puesto que ello facilitaba la programación. El proyecto nació bajo el nombre de IPython (Interactive Python), aprovechando código y experiencias de otros 2 proyectos con el mismo objetivo: LazyPython e IPP. Posteriormente se le agregó el sistema de notebooks, que permite tener contenedores editables con contenido de tipo y contenedores de código, editables y ejecutables. Una de las principales características es que los archivos generados, de extensión ipynb (ipython notebooks), son archivos de texto plano que pueden ser fácilmente almacenados bajo sistemas de control de versiones e intercambiados entre usuarios puesto que son independientes del sistema operativo. El modelo de edición y colaboración de los notebooks del proyecto IPython, estable, maduro y en funcionamiento desde 2011, fue tan exitoso que resultó natural extenderlo a otros lenguajes. En efecto, el sistema de notebooks no requiere utilizar de manera específica python, y puede aplicarse naturalmente a cualquier lenguaje. Buscando esta generalidad nace el proyecto jupyter, nombre tomado por los lenguajes Julia, Python y R, que resultan candidatos naturales para el mismo formato, al ser lenguajes de alto nivel, tipado dinámico y que no requieren compilación. Actualmente el proyecto jupyter soporta extensiones a más de 40

lenguajes, y admite la creación de extensiones para usos específicos. Tres ejemplos particulares y de gran contingencia a la educación lo constituyen los proyectos jupyter hub, jupyter nbgrader y jupyter RISE. El primero, jupyter hub, permite proveer una plataforma con copias paralelas de un notebook, situación particularmente conveniente para dictar laboratorios o tareas, puesto que cada estudiante puede autenticarse y completar el notebook con sus respuestas. El segundo, nbgrader, permite manejar y evaluar las respuestas realizadas por los estudiantes de manera automática. El tercero, jupyter RISE, permite presentar el contenido de un notebook en modo de presentación a través de slides, por lo que puede utilizarse para realizar clases completamente interactivas, puesto que el código es editable y ejecutarse durante la presentación sin necesidad de salir de ésta. Existen dos características esenciales que permiten justificar su uso en el largo plazo para la educación: el proyecto jupyter se encuentra suficientemente maduro y con financiamiento para garantizar su permanencia en los próximos años, y la comunidad detrás de jupyter desarrolla de manera activa mejoras y extensiones para su utilización en aula de clases.

Una de las principales ventajas de trabajar con jupyter notebooks consiste en la fácil integración de contenidos. Gracias al sistema de celdas de contenido (cells), es posible alternar de manera natural texto y código. En las celdas de tipo texto, es posible incluir de manera sencilla texto plano, markdown, latex y html, además de imágenes y videos. Esto posibilita que un jupyter notebook pueda servir como plataforma de enseñanza para una gran variedad de contenidos, y que pueda utilizarse para un gran número de metodologías de enseñanza-aprendizaje. En la figura 1 es posible apreciar el contenido que debe escribirse en una celda de texto, para obtener el resultado mostrado en la figura 2. En las celdas de tipo código es posible incluir código en python, y mediante la instalación de librerías adicionales, de julia, R y otros lenguajes de programación. En la figura 3 se muestra un código sencillo en python, que es ejecutado y cuyo resultado se muestra en la figura 4. Resulta interesante destacar que la ejecución del código se realiza en directo sobre la celda y regresa eventualmente el mismo tipo de errores que se obtiene en el lenguaje, permitiendo así encapsular los errores sin causar efectos colaterales en la ejecución de otras celdas, como se muestra en la figuras 5 y 6. Destaca que los modos de edición y de visión de las celdas de texto son completamente opuestos, mientras que los modos de edición y ejecución de celdas de código son idénticas y sólo varía el output que se obtiene al ejecutarlas.

Otra ventaja del formato ipynb desarrollado en IPython y jupyter es la versatilidad que permite para la exportación de formatos. En efecto, un archivo ipynb puede exportarse como página web (html), latex, markdown y pdf, además de permitir la exportación a formatos especificados por el usuario. Esta misma versatilidad permite que el uso de los notebooks pueda adaptarse a distintos tipos de metodologías educativas: es posible utilizar notebooks de manera pasiva en una clase tradicional, simplemente mostrando el texto y código a los estudiantes, de manera interactiva al ejecutar el código durante la clase y solicitando a los estudiantes completar o analizar el código presentado, o bien como parte de tareas y laboratorios en las cuales los estudiantes deben completar código y texto en determinadas secciones. El hecho de tener texto y código autocontenido ha sido aprovechado por algunas plataformas de cursos on-line, puesto que permite que sus alumnos aprendan de manera autodidacta, a través de la experimentación numérica, ya que no requiere un profesor guiando la experiencia educativa. En efecto los foros permiten responder las dudas y el profesor tiene el rol de crear el contenido para el auto aprendizaje.

Slide Type Slide

Ejemplo de celda de texto

****Jupyter notebooks**** es una excelente herramienta para el aula puesto que permite utilizar *diversos* tipos de contenidos, como Markdown y Latex. Se permiten ecuaciones en línea $f(x) = \alpha \cos^2(x)$ y multilínea:

$$\begin{array}{l} f(x) = \alpha \cos^2(x) \\ \leq \alpha \end{array}$$

Además es posible incluir imágenes utilizando código html:

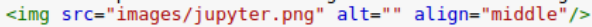


Figura N° 1. Celda de texto - modo edición.

Slide Type Slide

Ejemplo de celda de texto

Jupyter notebooks es una excelente herramienta para el aula puesto que permite utilizar *diversos* tipos de contenidos, como Markdown y Latex. Se permiten ecuaciones en línea $f(x) = \alpha \cos^2(x)$ y multilínea:

$$\begin{array}{l} f(x) = \alpha \cos^2(x) \\ \leq \alpha \end{array}$$

Además es posible incluir imágenes utilizando código html:



Figura N° 2. Celda de texto - modo visión.

Slide Type Slide

```
# Ejemplo de celda de código
digitos = set(range(10))
cuadrados = set([x**2 for x in range(10)])
digitos_cuadrados = digitos & cuadrados
for n in digitos_cuadrados:
    print(n)
```

Figura N° 3. Ejemplo de celda de código (python) - modo edición.

Slide Type Slide

```
# Ejemplo de celda de código
digitos = set(range(10))
cuadrados = set([x**2 for x in range(10)])
digitos_cuadrados = digitos & cuadrados
for n in digitos_cuadrados:
    print(n)
```

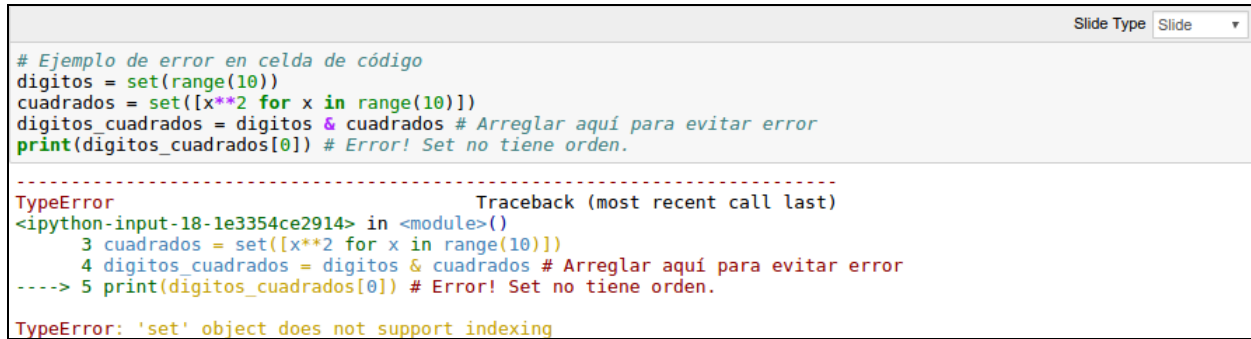
0
1
4
9

Figura N° 4. Ejemplo de celda de código (python) - modo ejecución.

Slide Type Slide

```
# Ejemplo de error en celda de código
digitos = set(range(10))
cuadrados = set([x**2 for x in range(10)])
digitos_cuadrados = digitos & cuadrados # Arreglar aquí para evitar error
print(digitos_cuadrados[0]) # Error! Set no tiene orden.
```

Figura N° 5. Ejemplo de error de ejecución en celda de código (python) - modo edición.



```
# Ejemplo de error en celda de código
digitos = set(range(10))
cuadrados = set([x**2 for x in range(10)])
digitos_cuadrados = digitos & cuadrados # Arreglar aquí para evitar error
print(digitos_cuadrados[0]) # Error! Set no tiene orden.

-----
TypeError                                 Traceback (most recent call last)
<ipython-input-18-1e3354ce2914> in <module>()
      3 cuadrados = set([x**2 for x in range(10)])
      4 digitos_cuadrados = digitos & cuadrados # Arreglar aquí para evitar error
----> 5 print(digitos_cuadrados[0]) # Error! Set no tiene orden.

TypeError: 'set' object does not support indexing
```

Figura N° 6. Ejemplo de error de ejecución en celda de código (python) - modo ejecución.

INICIATIVAS DE USO

Aplicaciones en Educación

Los notebooks de Jupyter están siendo activamente utilizados en universidades de todo el mundo. En general, la utilización se limita a la metodología de un profesor en un curso en particular, sin embargo, podemos encontrar esfuerzos de un nivel institucional, donde el caso más emblemático ocurre en University of California, Berkeley. De hecho, los principales desarrollos del proyecto Jupyter ocurren en el Berkeley Institute of Data Science (BIDS), alojado en dicha universidad, impulsando un uso intensivo de los notebooks en diferentes cursos de, por ejemplo, ciencia de datos y procesamiento de señales. En conjunto con UC Berkeley, California Polytechnic State University es otra institución impulsora del proyecto jupyter, y utilizan los notebooks en del Departamento de Física para cursos de computación científica.

Con respecto a profesores que individualmente utilizan notebooks de jupyter en el aula, podemos encontrar cursos de temas muy variados. Algunos ejemplos son

- Data Science:
 - Columbia University
 - New York University
- Computación científica:
 - University of Washington
 - King Abdullah University of Technology
- Bioinformática:
 - Vanderbilt University
 - Northern Arizona University
- Neurociencia:
 - Massachusetts Institute of Technology
- Aerodinámica:
 - The George Washington University

Incluso, esta herramienta ha sido usada para introducir a escolares a la programación en Alaskan High School.

Un caso que toca de cerca la realidad nacional, es el curso online masivo abierto “Practical Numerical Methods in Python”. Este curso nació en 2014 como una colaboración liderada por Lorena Barba en The George Washington University, con profesores de University of Southampton y la Pontificia Universidad Católica de Chile, aunque las instituciones involucradas han ido cambiando en ediciones más recientes. La idea era coordinar el curso de métodos numéricos que se dictaba en cada una de estas universidades a través de este MOOC, y poner

todo el material en internet. De esta forma, el curso estaba abierto a todo público, mientras que aportaba con créditos a quienes estuviesen registrados en alguna de las universidades. El material era compartido en el formato de los jupyter notebooks.

La forma de uso de los notebooks es también muy variada. Sin embargo, lo más común es encontrarlos como cápsulas autocontenidas con la explicación de algún tema en el texto, junto con ejemplos donde el estudiante puede interactuar con el código. Este formato es muy útil en el contexto de aprendizaje activo, ya que permite al estudiante interactuar con el material mientras el profesor actúa como un tutor. Cabe destacar que en la mayoría de los casos, los profesores utilizan la plataforma de Github (<https://github.com>) para presentar el material, ya que es abierto, y permite visualizar (aunque no ejecutar) el contenido del notebook online, sin tener que descargarlo.

Aplicaciones en la Industria

El sistema de notebooks de jupyter está siendo integrado de manera progresiva en distintos ámbitos de la industria. Una de las primeras empresas en integrar jupyter notebooks fue Github, que provee alojamiento de proyectos utilizando el sistema de control de versiones git, y que desde mayo del 2015 permite visualizar automáticamente y de manera estática los archivos con extensión ipynb. Las plataformas de Microsoft, Azure ML Studio y Azure HDInsights, permiten generar notebooks para documentar y ejemplificar el código generado en la plataforma. Similares servicios son permitidos por IBM con sus plataformas Bluemix Data and Analytics y Data Scientist Workbench, además de Continuum Analytics Anaconda Enterprise y SageMathCloud. Otras empresas como Quantopian y Kaggle utilizan notebooks para la investigación en inversiones y competencias de Data Science. En general, cualquier empresa puede generar y mantener de manera sencilla notebooks con explicaciones sobre la utilización, o con análisis dinámicos de datos como sucede en equipos de data science.

Por último, resulta destacable la iniciativa de artículos reproducibles, en la cual artículos científicos presentan el código y los datos utilizados en el artículo, permitiendo su libre reproducción por terceros. En esta línea, la editorial O'Reilly anunció en Mayo del 2015 que comenzará a mantener una plataforma con jupyter notebooks de manera complementaria a sus libros, de conocido y marcado contenido técnico. De esta manera, quien haya adquirido un libro contará con "ejemplos autocontenidos" que facilitarán la experimentación y asimilación del material.

INICIATIVAS EN LA UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

La Universidad Técnica Federico Santa María posee una cuenta github, en la dirección <https://github.com/usantamaria>, donde se encuentran diversos cursos que utilizan jupyter notebooks, permitiendo integrar y coordinar los esfuerzos de distintos profesores. En los contenidos de los notebooks de jupyter citados a continuación, los textos cuentan con licencia CC-BY 4.0 y los códigos con código MIT.

Curso de programación

Durante el segundo semestre del 2015, se desarrolló material pedagógico en el formato jupyter notebooks para el curso IWI131, Programación de Computadores, que corresponde a un curso común de primer año de ingeniería, donde se enseña pensamiento algorítmico y programación en python. El contenido de dicho curso se encuentra en la dirección <https://github.com/usantamaria/iwi131>. Algunas de las principales características del material generado son la utilización de la extensión RISE para presentación en formato de slides del contenido, permitiendo un alto grado de interacción durante las clases. Fue posible hacer

participar a los alumnos en la generación de código, mostrar y solucionar errores comunes, e integrar de manera sencilla el contenido con la aplicación del mismo en vivo durante las clases.

Curso de matemática

Durante el segundo semestre del 2015, se utilizó el formato jupyter notebooks, tanto para clases como laboratorios numéricos, en la elaboración del contenido del curso MAT281, Aplicaciones de la Matemática en la Ingeniería. El contenido se encuentra disponible en <https://github.com/usantamaria/mat281>. Este curso se dicta para alumnos de cuarto año de la carrera de Ingeniería Civil Matemática, y en él se aplican diversas técnicas de modelamiento matemático y el uso de algunos algoritmos clásicos de Machine Learning. Algunos puntos principales son la aplicación de jupyter notebook tanto para el contenido de las clases, con la extensión RISE, como para uso en los laboratorios. La utilización de jupyter notebooks en clase y laboratorios fue una experiencia muy positiva y ampliamente valorada por los alumnos, para los cuales constituía una valiosa instancia de aprendizaje guiado previo a las tareas que se realizaron en formato tradicional.

Curso de mecánica

El Departamento de Ingeniería Mecánica de la UTFSM tiene una gran inclinación hacia la mecánica computacional, por lo que imparte varios cursos afines a esta disciplina a nivel de pregrado y postgrado. Uno de estos cursos es Fundamentos de la Dinámica de Fluidos Computacional (MEC223), dirigido a alumnos en el quinto año de Ingeniería Civil Mecánica, donde el principal enfoque es la implementación de soluciones numéricas de ecuaciones diferenciales parciales, hasta llegar a resolver la ecuación de Navier-Stokes. En este curso se pasa más de la mitad del tiempo en el laboratorio, y utiliza jupyter notebooks para el desarrollo de éstos. El contenido del curso está disponible en el enlace https://github.com/usantamaria/mec223_laboratorios.

La utilización de jupyter notebooks se centró principalmente en el desarrollo de guías de laboratorio interactivas y en la utilización de ejemplos en clases expositivas.

Repositorio Centralizado

Bajo el proyecto MECESUP FSM1307 “Implementar un Plan de Integración de las Tecnologías para articular experiencias de innovación pedagógica y explorar iniciativas de formación en D-learning en las carreras de la UTFSM” se generó material transversal a cursos de carácter numérico. El material se encuentra disponible bajo licencia MIT en la dirección https://github.com/usantamaria/ipynb_para_docencia. La idea central es que gran parte del material de los 3 cursos anteriormente mencionados es común, y consiste en instrucciones sobre la utilización del material, python básico, algoritmos básicos, y uso de la librería numpy, por lo que resulta natural la existencia de un sólo lugar donde el material debe mantenerse actualizado, y que permite que los alumnos puedan aprender el material incluso sin la necesidad de estar realizando un curso, a través de las explicaciones proporcionadas y el desarrollo de desafíos que permiten al estudiante medir su grado de avance y conocimiento del material en estudio.

PROPUESTA METODOLÓGICA

Clases

La utilización de notebooks de jupyter durante las clases resulta natural y deseable para cursos en los cuales se enseña programación en alguno de los lenguajes de python, R o Julia. Jupyter realiza un buen trabajo de indentación y coloración del código, lo cual permite presentar

ejemplos de código que son fácilmente asimilables. El código presentado en los notebooks puede ejecutarse, con lo cual es posible comprobar empíricamente durante la clase la ejecución del código y donde es posible interactuar con los estudiantes mostrándoles el resultado de cambios en los algoritmos o la aparición de errores en la ejecución. Como se mencionó anteriormente, existe una extensión a jupyter notebook llamada RISE, que permite presentar las celdas de texto y código como transparencias (slides) de manera muy similar a lo que se realiza tradicionalmente con PowerPoint, pero con la ventaja de poder editar tanto texto como código durante la presentación, y de poder ejecutar el código presentado.

Laboratorios

El uso de los notebooks en ejercicios numéricos de laboratorio resulta natural. En un notebook, hay espacio para discutir la teoría y el contexto detrás del ejercicio de laboratorio en un texto para luego dar espacio al estudiante para que realice los ejercicios. A partir de una guía de laboratorio bien diseñada en un notebook, el profesor pasa a ser un tutor, ya que las principales dudas teóricas son resueltas por la guía. Una gran ventaja de usar un notebook de jupyter como guía de laboratorio, es que se hace fácil entregar al alumno un código prehecho, explicando en el texto donde se espera que el alumno modifique o interactúe con el notebook. De esta forma, el alumno puede focalizar su aprendizaje en lo pertinente al laboratorio, sin distraerse con otras cosas. Por ejemplo, si el objetivo del laboratorio es aprender un algoritmo, en el notebook puede venir previamente configurado el código necesario para realizar la visualización de los resultados, y así el alumno no se tiene que preocupar de esto. Además, al presentar un código de buena calidad al alumno se le están inculcando buenas prácticas de programación.

Tareas

Similar al caso de los laboratorios, los notebooks de jupyter resultan muy útiles para diseñar tareas. Puesto que es posible incluir fórmulas matemáticas usando Latex en el texto para darle contexto y discutir la teoría detrás del ejercicio, el documento de la tarea es completamente autocontenido. Además, el profesor puede incluir código prehecho, focalizando el aprendizaje en lo que es pertinente a la tarea, e inculcando buenas prácticas de programación.

El estudiante comúnmente tiene más tiempo para desarrollar una tarea que un laboratorio, por lo que el volumen de trabajo e información puede ser mayor. Es más, un notebook permite agregar enlaces, figuras y videos para información adicional, facilitando así el trabajo de preparación del documento, y enriqueciendo la experiencia del estudiante. Además las respuestas pueden escribirse en un lenguaje que contiene expresiones de Latex, pero que no requiere compilación y que resulta más fácil de utilizar para los estudiantes.

Ya que los estudiantes cuentan con más tiempo para una tarea que un laboratorio, los ejercicios son de una complejidad mayor, y normalmente hay una alta probabilidad de cometer errores en el desarrollo. Una ventaja de la estructura de los notebooks, basado en celdas independientes con porciones pequeñas de código ejecutable, es que facilita al estudiante el proceso de ensayo y error. Si en un notebook existen celdas ejecutadas, es posible operar sobre las variables de esas celdas en nuevas celdas, lo que permite al usuario interactuar con el código, facilitando la solución de posibles problemas ("bugs").

BUENAS PRÁCTICAS

Control de versiones

Una práctica muy común en proyectos donde se usan los notebooks de jupyter es combinarlo con Github (<https://github.com>). Github es un repositorio web basado en el programa de control de versiones git, que es ampliamente utilizado por desarrolladores de software, principalmente los que apoyan iniciativas de código abierto. El uso de Github es atractivo en el contexto de los

jupyter notebooks porque permite visualizar el contenido del notebook online, sin tener que descargarlo. Además, al estar basado en git, obliga al desarrollador a un flujo de trabajo muy ordenado y seguro, con acceso todas las versiones anteriores, y la posibilidad de hacer desarrollos en diferentes ramas en paralelo. El hecho de que por defecto la información en Github sea abierta, presiona al desarrollador a seguir buenas prácticas en su trabajo. Para que lo que se encuentra en la cuenta de Github sea entendible por alguien externo, es necesario mantener el material y foros actualizados, tener una versión estable que solamente se actualice por otra versión igual de estable, y realizar comentarios con información clara en cada actualización. Estas prácticas no ayudan solamente a quien accede al proyecto desde afuera, sino que además al mismo desarrollador y posibles futuros colaboradores al proyecto (por ejemplo, otros profesores que utilicen el material).

Resultados esperados

Puesto que tanto el texto como el código de un notebook de jupyter es editable, resulta pedagógicamente importante explicitar cuando se espera una interacción o respuesta específica del estudiante. Así, por ejemplo, si el código lanzara un error al ejecutar, esto debería ser señalado apropiadamente mediante un comentario que indique el tipo de error (y eventualmente una explicación de porqué ocurre el error). De manera similar, si en un notebook de laboratorio o de tarea es esperable que el estudiante edite una celda de texto o de código para introducir su respuesta, es necesario que el texto contenga una indicación para el estudiante.

Resulta aconsejable indicar al principio del notebook el contenido y/o la finalidad pedagógica de éste. Así, el estudiante se focaliza en el objetivo desde el principio, y realiza la lectura de la parte teórica ya en el contexto de lo que se le pedirá, haciendo más eficiente el proceso de aprendizaje. Un punto interesante de los notebooks de jupyter es que puede ser exportado a formato PDF. Así, el estudiante puede realizar su informe en el mismo documento donde trabaja el código.

Material autocontenido

Los notebooks de jupyter son una buena herramienta para generar material autocontenido, puesto que se puede enlazar texto en Latex, imágenes y videos para la explicar teoría, y el código ejecutable permite realizar ejemplos concretos, todo en el mismo documento. Las imágenes o videos pueden ser incrustadas a partir de una url, pero en general es conveniente guardar una copia local y así evitar que la referencia pueda desaparecer con la página web y que el notebook pueda funcionar sin acceso a internet. Sin embargo es importante no saturar el notebook con información en el afán de que sea autocontenido. En nuestra experiencia, conviene diseñar el notebook para que su desarrollo no tarde más de dos horas (incluyendo la lectura de la parte teórica), de manera que pueda realizarse de manera lineal y sin interrupciones.

Utilización de librerías auxiliares

A veces es conveniente encapsular el funcionamiento de algunas rutinas más avanzadas o cuyo funcionamiento no queremos revelar, como por ejemplo el código solución o rutinas gráficas. Para ello, es posible generar un archivo adicional al jupyter notebook, y que es posible importar de manera natural. Así, por ejemplo, si estuviéramos trabajando con el lenguaje python, sería necesario crear un archivo con extensión .py, por ejemplo, `mi_libreria.py` donde definimos una función `mi_funcion`. Desde jupyter notebooks podríamos utilizar la función anterior simplemente ejecutando:

```
import mi_libreria
mi_libreria.mi_funcion()
```

CONCLUSIONES

A pesar que los notebooks de jupyter son el resultado de varios años de desarrollo, es solamente en los últimos años que su utilización ha tomado singular fuerza. De manera específica a la academia, existe una importante comunidad de profesores de las universidades más importantes del mundo que lo utilizan en el aula, tocando temas que van desde neurociencia hasta aerodinámica. La principal ventaja es que es una herramienta muy versátil, ya que puede ser utilizado en varios contextos docentes (clases expositivas tradicionales, clases activas, laboratorios, tareas, etc.), y reafirma metodologías de aprendizaje activo y aprendizaje por pares. Además, sirve como plataforma para inculcar buenas prácticas de programación en los estudiantes.

En este trabajo hacemos una revisión del uso de los notebooks de jupyter como herramienta docente, y presentamos iniciativas de diferentes profesores de la Universidad Santa María donde se usa esta herramienta, centralizados en la cuenta usantamaria de Github. Este material es de libre acceso y queremos dejar abierta la invitación a quienes quieran usarlo y/o contribuir a él. Si bien la aplicación de la herramienta es muy reciente y no se cuentan actualmente con comparaciones que permitan justificar cuantitativamente el efecto de su aplicación en el aula, si ha sido posible observar un cambio significativo del ambiente de aprendizaje en el aula y de manera transversal se han obtenido comentarios muy positivos de los estudiantes respecto al material entregado. Para incentivar el uso de jupyter notebooks se ha generado material introductorio a los notebooks y que resulta transversal a un gran número de cursos, y que los autores esperan sirva para que nuevos profesores introduzcan esta herramienta en su aula.

AGRADECIMIENTOS

Los autores desean agradecer al proyecto MECESUP FSM1307 y particularmente a Cecilia Ritchie por su constante apoyo en el uso de nuevas tecnologías para mejorar el aprendizaje de los estudiantes de la UTFSM.

REFERENCIAS

Rossant C. (2013). Learning IPython for Interactive Computing and Data Visualization. Packt Publishing, ISBN 1782169938.

Neuman M. (2013). Computational Physics with Python, CreateSpace Independent Publishing Platform, ISBN 1480145513

Langtangen H.P. (2012). A Primer on Scientific Programming with Python. Springer, 2012, 3642302920

Perez F. et al (2007). IPython: A System for Interactive Scientific Computing, Computing in Science and Engineering, 9 (3), 21-29.

Light G. et al (2009). Learning and Teaching in Higher Education: The Reflective Professional. Sage.

ENLACES

Ejemplo de artículo reproducible, publicado en Nature, <http://www.nature.com/news/ipython-interactive-demo-7.21492>

Entrada del blog personal de Fernando Perez con detalles de la creación de ipython, <http://blog.fperez.org/2012/01/ipython-notebook-historical.html>

Entrada en el blog de IBM, revisión del ecosistema jupyter, <http://blog.ibmjstart.net/2016/03/21/powered-by-jupyter/>

Entrada del blog de jupyter notebooks, detallando el uso de los notebooks para colaborar en data science, <http://blog.jupyter.org/2015/07/07/project-jupyter-computational-narratives-as-the-engine-of-collaborative-data-science/>

Lista de correo de proyectos educativos con jupyter notebooks, <https://groups.google.com/forum/#!topic/jupyter-education/HprQgDcDbo4>

Listado de lenguajes admitidos en jupyter notebooks mediante extensiones, <https://github.com/ipython/ipython/wiki/IPython-kernels-for-other-languages>

Noticia en Nature, sobre los notebooks interactivos que permiten compartir el código, <http://www.nature.com/news/interactive-notebooks-sharing-the-code-1.16261>

Noticia en editorial O'Reilly sobre utilización de jupyter notebooks, <https://www.oreilly.com/ideas/jupyter-at-oreilly>

Página web principal del proyecto jupyter notebook, <https://jupyter.org/>

Repositorio de código de Data Science, Columbia University, <https://github.com/ledeprogram/courses/tree/master/algorithms>

Repositorio de código de Data Science, New York University, <http://jattenberg.github.io/PDS-Fall-2013/>.

Repositorio de código de Computación científica, University of Washington, <http://www.astro.washington.edu/users/vanderplas/Astr599/>),

Repositorio de código de Computación científica, King Abdullah University of Technology (<https://github.com/ketch/HyperPython>)

Repositorio de código de Bioinformática, Vanderbilt University, <https://plot.ly/ipython-notebooks/computational-bayesian-analysis/>.

Repositorio de código de Bioinformática, Northern Arizona University, <http://readiab.org/>

Repositorio de código de Neurociencia, Massachusetts Institute of Technology, <http://gablab.mit.edu/index.php/gablab/people/research-scientists/satra-ghosh>

Repositorio de código de Aerodinámica, The George Washington University, <https://github.com/barbagroup/AeroPython>

Repositorio de código de Introducción a Programación, Alaskan High School, <http://introtopython.org/>