

UNIVERSIDAD DE CÓRDOBA
Facultad de Ciencias

Grado en Bioquímica

TRABAJO FIN DE GRADO

Desarrollo de un tutorial de programación en Python orientado al estudiante de Bioquímica

Código del TFG: BQ-21-38-IAN

Tipología: Trabajo docente

Autor: Víctor Jesús FERNÁNDEZ RAMÍREZ



24 de mayo de 2022

Índice general

1. Contexto docente	1
2. Justificación	3
3. Objetivos	5
4. Desarrollo del trabajo docente	7
4.1. Tecnologías usadas en la realización del tutorial	7
4.1.1. PyCharm	7
4.1.2. GitHub	7
4.1.3. Markdown	9
4.1.4. Mkdocs	9
4.1.5. Google Colab	10
4.1.6. LaTeX	10
4.2. Desarrollo: organización y visualización	10
4.2.1. Inicio	10
4.2.2. Introducción a <i>Python</i>	10
4.2.3. PyCharm	10
4.2.4. Variables y operadores	11
4.2.5. Estructuras de datos	11
4.2.6. Estructuras de control	11
4.2.7. Funciones	11
4.2.8. Archivos y datos	11
4.2.9. Graficación	11
4.2.10. Ejercicios complementarios	11
4.2.11. Biopython	11
4.2.12. Acerca de	11
4.3. Material de apoyo	12
4.3.1. Ejemplos prácticos	12
4.3.2. Ejercicios	12
4.3.3. Enlaces de interés	12
5. Conclusiones y trabajo futuro	13

A. Material digital de apoyo	19
A.1. Python script	19
A.2. Google Colab	19
A.3. Ficheros de datos	19
B. Fichero mkdocs.yml	21
C. Fichero Markdown	25

Índice de figuras

Resumen del TFG

Resumen

El estudiante del Grado de Bioquímica en la Universidad de Córdoba adquiere una gran cantidad de competencias, entre las que se incluyen competencias bioinformáticas. Entre estas, destacan los conceptos aprendidos sobre el empleo del sistema operativo *Linux*, y el uso del lenguaje de programación *R*.

En este Trabajo de Fin de Grado se presenta un tutorial aplicado al estudiante de Bioquímica sobre un lenguaje de programación denominado *Python*. *Python* se trata de uno de los lenguajes más utilizados en la actualidad por empresas de todo el mundo, destacando por su potencia y facilidad de aprendizaje. Debido a esto, en el mundo de la Biología Computacional se ha incrementado su uso, siendo una competencia directa en el uso de *R* y su librería *Bioconductor*, gracias a la aparición de librerías como *Biopython*.

A lo largo del presente tutorial el estudiante aprenderá *Python* desde cero, partiendo desde su descarga e instalación, hasta llegar a conceptos más avanzados como el uso de estructuras de control, diseño de gráficas y uso de librerías específicas. Para ello se le irán proponiendo una serie de ejercicios complementarios, para que pueda ir poniendo en práctica los conocimientos adquiridos.

Sumado a esto, el estudiante también aprenderá conceptos básicos sobre una serie de herramientas muy empleadas en el campo de la Bioinformática, como son \LaTeX , *Markdown*, *MkDocs* y *GitHub*.

Palabras clave: Tutorial, Python, PyCharm, Markdown, Bioquímica, Bioinformática.

Abstract

The student of the Degree in Biochemistry at the University of Córdoba acquires a large number of skills, including bioinformatics skills. Among these, the concepts learned about the use of the operating system *Linux*, and the use of the programming language *R* stand out.

This Final Degree Project presents a tutorial applied to the Biochemistry student on a programming language called *Python*. *Python* is one of the languages most used today by companies around the world, standing out for its power and ease of learning. Due to this, its use has increased in the world of Computational Biology, being a direct competition in the use of *R* and its library *Bioconductor*, thanks to the appearance of libraries such as *Biopython*.

Throughout this tutorial the student will learn *Python* from scratch, starting from its download and installation, to more advanced concepts such as the use of control structures, graphic design and the use of specific libraries. For this, a series of complementary exercises will be proposed, so that you can put into practice the knowledge acquired.

In addition to this, the student will also learn basic concepts about a series of tools widely used in the field of Bioinformatics, such as \LaTeX , *Markdown*, *MkDocs* and *GitHub*.

Keywords: Tutorial, Python, PyCharm, Markdown, Biochemistry, Bioinformatics.

Capítulo 1

Contexto docente

Este tutorial de programación en Python orientado al estudiante de Bioquímica se enmarca dentro del ámbito de la materia *Informática aplicada a la Bioquímica*. Esta asignatura se imparte durante el primer cuatrimestre del segundo curso del Grado de Bioquímica de la Universidad de Córdoba [10]. En esta materia se familiariza al alumnado de Bioquímica con un conjunto de herramientas informáticas que le pueden ser de utilidad en su carrera profesional. Por ello, abarca temas como el funcionamiento básico de un ordenador, manipulación de base de datos científicas, programación computacional en *R* y su entorno *Rstudio* para optimizar el tratamiento de datos biológicos, manejo de *Linux* y edición de ficheros en \LaTeX . No obstante, aunque esta ha sido la principal asignatura en la que se han aprendido estos aspectos bioinformáticos, presenta competencias cruzadas con otras asignaturas del Grado de Bioquímica como *Biología Molecular de Sistemas*, *Biofísica* y *Estructura de Macromoléculas*.

El Grado de Bioquímica en la Universidad de Córdoba se basa en el estudio de los constituyentes químicos de los seres vivos y sus transformaciones, así como de las estructuras e interacciones de las macromoléculas que permiten la vida de estos organismos. Este concepto deriva cada vez más hacia el término de Bioquímica y Biología Molecular, ya que la complejidad de estructuras, organizaciones y funciones de los seres vivos a nivel molecular es fundamental en la comprensión de la vida [7]. La investigación tanto básica como aplicada en Bioquímica y Biología Molecular ha avanzado de una forma muy significativa en el siglo XX, continuando su avance exponencial en el siglo XXI.

Las Biociencias Moleculares y sus aplicaciones están consideradas una de las grandes olas de expansión de la economía basada en el conocimiento. De hecho, la Unión Europea ha apostado claramente por convertirse en uno de las principales líderes mundiales en la economía basada en el conocimiento. La generación de conocimiento es fundamental en el abordaje de necesidades globales relacionadas con la salud, alimentación, medio ambiente y sostenibilidad. *La Agenda 2030 para el Desarrollo Sostenible* plantea un marco ambicioso de objetivos y metas universales e indivisibles para hacer frente a una variedad de desafíos societa-rios mundiales. La

diversidad biológica y los ecosistemas figuran en forma destacada en muchos de los *Objetivos de Desarrollo Sostenible (ODS)* y metas asociadas, en las que la Bioquímica y Biología Molecular tiene un papel crítico [17].

La importancia de la Bioquímica y Biología Molecular también ha dejado huella a nivel local en Córdoba, donde encontramos el Parque Tecnológico Rabanales XXI con industrias biotecnológicas y biomédicas, y el Instituto de Biomedicina Maimónides de la Universidad de Córdoba que incorpora profesionales bioquímicos en investigación biomédica avanzada. En el ámbito agroalimentario resalta el Campus de Excelencia Agroalimentario, liderado por la Universidad de Córdoba y que agrupa a las Universidades de Cádiz, Huelva, Jaén, y Almería, y que requiere para su desarrollo graduados formados en estas tecnologías agroalimentarias, y muchas de las destrezas a nivel molecular que posee el graduado de Bioquímica.

Capítulo 2

Justificación

La Bioinformática es una subdisciplina de la Biología y de la Informática que se ocupa de la adquisición, almacenamiento, análisis y difusión de datos biológicos. En la actualidad, la parte más importante de la Bioinformática es el análisis e interpretación de los datos de las moléculas biológicas, un proceso denominado Biología Computacional [13]. Por tanto, la Bioinformática es importante por su gran aporte al desarrollo de la Biología en la investigación básica, haciendo posible el tratamiento de la extensa información que se puede obtener a partir del estudio de organismos: genomas, proteomas, transcriptomas y metabolomas.

En base a la importancia actual y futura de la Bioinformática, la impartición de la asignatura *Informática aplicada a la Bioquímica* es fundamental en el Grado de Bioquímica. A lo largo del Grado hay otras asignaturas en las que se tratan herramientas bioinformáticas, destacando especialmente *Biología Molecular de Sistemas y Estructura de Macromoléculas*. No obstante, teniendo en cuenta el total de 38 asignaturas impartidas, el porcentaje de competencias bioinformáticas es relativamente bajo.

En la asignatura *Informática aplicada a la Bioquímica* el alumnado de Bioquímica adquiere conocimientos básicos acerca del lenguaje informático *R* [10]. Este lenguaje, uno de los más utilizados en Bioinformática, no deja de ser una buena opción para ser utilizado por un bioquímico. Sin embargo, no destaca por su simpleza, y si el bioquímico conoce únicamente *R* puede verse limitado a la hora de utilizar otra serie de herramientas, programas o librerías.

Por este motivo en el presente trabajo se propone un tutorial de *Python* orientado al estudiante de bioquímica. Uno de los objetivos por los que fue concebido *Python* en sus principios fue para ser un lenguaje fácil de aprender. La elegante sintaxis de *Python* y su tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para el desarrollo de programas y aplicaciones en diversas áreas y sobre la mayoría de las plataformas. De hecho, para confirmar el peso de *Python* a día de hoy, puede comprobar que grandes gigantes como Instagram, Google y Spotify lo eligen para el backend de sus páginas web [18].

Esta tendencia al uso de *Python* también ha afectado al campo de las Biocien-

cias Moleculares. Esto se ha debido principalmente al incremento en disponibilidad de librerías de visualización y procesamiento de datos como *NumPy*, *SciPy*, *Pandas* o *Matplotlib*. Este incremento ha conllevado a que actualmente un bioquímico pueda realizar las mismas funcionalidades en *Python* que con *R*, sumado a la aparición de librerías específicas como *Biopython* que se asemejan a *Bioconductor* de *R* [3]. Aun así, hay que tener en cuenta que ambas librerías al igual que contienen programas en común, no compartirán muchos otros. Este es el factor diferencial que debe tener en cuenta: es fundamental que adquiera competencias cruzadas, ya que en situaciones problemáticas en las que un lenguaje no le pueda aportar una solución, otro quizás sí lo haga.

Además, ha de saber que el aprendizaje del lenguaje de programación *Python* le abrirá como bioquímico muchas puertas en su futuro como profesional, ya que cada vez son más los laboratorios de investigación que requieren de bioinformáticos para tareas como análisis de archivos de secuenciación, modelado de proteínas y alineamiento de secuencias en este lenguaje de programación [5].

Capítulo 3

Objetivos

El objetivo principal del presente Trabajo de Fin de Grado es el desarrollo de un tutorial de *Python* desde cero, para que un estudiante de Bioquímica sin ningún conocimiento en este lenguaje de programación adquiera los suficientes conceptos básicos para manejarse en el mismo. De hecho, este tutorial puede ser utilizado por alumnado externo al Grado de Bioquímica, presentando la única limitación de que la mayoría de ejemplos y ejercicios propuestos están enfocados en el campo de la Bioquímica y Biología Molecular.

El desarrollo del tutorial está orientado de tal forma que el alumnado de Bioquímica no va a adquirir conocimientos de programación que le puedan ser tediosos y extensos. En todo momento se ha tenido en cuenta en su diseño que un alumno de Bioquímica no es un informático, y no se puede pretender que lo sea ya que esta tarea le resultará muy compleja en la mayoría de las ocasiones. Por este motivo, el tutorial está dividido en una serie de apartados o secciones, partiendo de conceptos muy básicos y terminando con ejercicios con mayor complejidad, todo desde un punto de vista muy práctico que le supondrá una vía fácil de aprendizaje.

Sumado a esto, una de las finalidades de este trabajo es que el alumnado siga adquiriendo y desarrollando una serie de competencias definidas en el Grado de Bioquímica: capacidad de razonamiento crítico, aplicación de los principios básicos del método científico, capacidad de aprendizaje autónomo, y empleo de herramientas básicas para la comunicación, búsqueda de información y tratamiento de datos [8].

Asimismo, todo el material desarrollado en el tutorial, desde el código fuente del servidor web hasta los scripts de *Python*, están incluidos en un repositorio de GitHub accesibles para toda la comunidad educativa de forma totalmente gratuita, buscando llegar al máximo porcentaje posible del sector educativo externo a la Universidad de Córdoba.

Capítulo 4

Desarrollo del trabajo docente

4.1. Tecnologías usadas en la realización del tutorial

4.1.1. PyCharm

Para escribir, leer y editar código de la manera más eficiente posible, se necesita un editor de texto o un entorno de desarrollo integrado (*IDE, Integrated development environment*). Un IDE no deja de ser un editor de texto, pero que incluye otra serie de herramientas que le permiten examinar el código mientras se introduce y aprender sobre el mismo [14].

PyCharm es un IDE muy popular desarrollado por la compañía *Jetbrains*, construido específicamente para programar en el lenguaje Python. La versión gratuita se denomina *PyCharm Community Edition*, y contiene una gran cantidad de herramientas y opciones disponibles [15].

En este tutorial, PyCharm ha sido el IDE seleccionado para escribir, leer y editar scripts de Python; así como para escribir los ficheros Markdown que componen las distintas secciones del tutorial. De hecho, en la sección 4.2.3 del tutorial se trata exclusivamente la manipulación de PyCharm. Aunque este IDE haya sido el seleccionado, existen muchos otros IDE y editores de texto empleados por la comunidad científica, destacando Sublime Text y Visual Studio Code (VSC).

4.1.2. GitHub

GitHub es un servicio basado en la nube que aloja un sistema de control de versiones (VCS) llamado Git, siendo por tanto, un servicio de alojamiento en la Web de repositorios remotos Git [14].

GitHub permite a los desarrolladores colaborar y realizar cambios en proyectos compartidos, a la vez que mantienen un seguimiento detallado de su progreso. Por estos motivos, GitHub es una herramienta muy utilizada por la comunidad bioinformática, existiendo miles de repositorios disponibles de programas que pueden resultar de

utilidad, totalmente gratuitos.

GitHub ha sido empleado en el desarrollo de este tutorial con dos objetivos: incluir todo el material desarrollado en un repositorio accesible por la comunidad estudiantil, y facilitar la comunicación y feed-back entre el autor y el tutor del trabajo en todo momento.

PyCharm es un IDE totalmente adaptado al uso de GitHub. Como aparece en la sección 4.2.3 del tutorial, la interfaz de PyCharm permite administrar proyectos de Git alojados en GitHub de forma muy simple, pudiendo crear un repositorio propio en la opción *Create git repository*, compartir el proyecto en GitHub en la opción *Share project on GitHub*, sincronizar los archivos desde el ordenador local al repositorio con la opción *Push*, y sincronizarlos desde el repositorio al ordenador local con la opción *Pull*.

No obstante, aunque utilizando PyCharm la forma anterior es la más simple, también se pueden realizar estas funciones desde la terminal del sistema [16]:

1. Entrar en el directorio contenedor del proyecto.
2. Creación de un subdirectorio *.git* con todos los archivos necesarios para la conexión entre repositorio y plataforma.

```
# git init
```

3. Indicación y confirmación de los archivos a subir al repositorio.

```
# git add [file]  
# git commit
```

4. Actualización de la versión de los archivos del ordenador local al repositorio.

```
# git push origin master
```

5. Actualización de la versión de los archivos del repositorio al ordenador local.

```
# git pull origin master
```

Por último, es fundamental destacar que GitHub permite publicar en un sitio web el contenido del repositorio de forma completamente gratuita, utilizándose esta opción para la creación del dominio web donde se aloja el tutorial [6]:

https://victorffdez.github.io/Python_Biochemistry_Tutorial/

4.1.3. Markdown

Markdown es un lenguaje de marcaje ligero creado por John Gruber y Aaron Swartz en 2004, que permite la escritura rápida de texto. Fue diseñado con el objetivo de ser tan fácil de leer y escribir como fuera posible, presentando la gran ventaja de poder convertir el documento de texto a formato HTML [12].

Todas las secciones del tutorial han sido desarrolladas utilizando el lenguaje Markdown, siendo cada uno de los apartados un documento de texto en formato *.md*, característico de Markdown. En el Apéndice C se muestra de ejemplo uno de estos ficheros Markdown creados. En este fichero, se puede comprobar que este lenguaje presenta una sintaxis particular en la que el asterisco (*) y el guión bajo (_) se utilizan para indicar cursiva y/o negrita, la almohadilla (#) para determinar los encabezados y el acento grave (`) para identificar código.

4.1.4. Mkdocs

Mkdocs es un software que permite generar sitios estáticos, orientado a la creación de plataformas de documentación [4]. Esta herramienta escrita en Python es bastante simple de configurar, presentando como principal requisito que los ficheros estén en formato *.md* de Markdown.

Mkdocs ha sido utilizado en el desarrollo de este tutorial con el objetivo de poder enriquecer el documento utilizando las extensiones que posibilita, como incluir notas, mejorar la visualización del bloque de código o insertar diagramas. Sumado a esto, al generar sitios estáticos, esta herramienta permite visualizar en todo momento las modificaciones realizadas en el texto.

Al estar en código Python, la instalación de Mkdocs es muy simple y se puede realizar desde la opción *Python Packages* de PyCharm. Desde aquí, ha de instalarse el paquete *Mkdocs*, así como el paquete del tema que se desea emplear, en este caso *Mkdocs-material* [9]. Por último y de forma complementaria, también se instala el paquete *Mkdocs-material-extension* que contiene algunas funciones adicionales.

- Visualización del sitio web desde el directorio de trabajo.

```
# mkdocs serve
```

- Construcción de la documentación del sitio web al finalizar el proyecto.

```
# mkdocs build
```

Por tanto, el aspecto visual del sitio web del tutorial depende principalmente de la sintaxis de los ficheros *.md*, así como de la configuración del tema de Mkdocs seleccionado, albergada en un fichero *mkdocs.yml* mostrado en el Apéndice B.

4.1.5. Google Colab

La forma tradicional de escribir y ejecutar un programa en Python es crear un archivo `.py` que almacene su código fuente. No obstante, aunque en la mayor parte del tutorial sea este el método utilizado, en el apartado *Ejercicios* de la sección *Graficación* se introduce al alumno en el empleo de los cuadernos de Google Colab.

Los cuadernos en lenguaje de programación (normalmente conocidos como *notebook*) permiten combinar texto y código, siendo muy útiles para documentar el trabajo a la vez que se prueba código fácilmente. En este caso, Google Colaboratory es un producto de Google Research que permite a cualquier usuario escribir y ejecutar código arbitrario de Python en el navegador [2]. En el Apéndice A.2 se muestra parte del código de este cuaderno.

4.1.6. LaTeX

\LaTeX [1] es un sistema de composición de textos de alta calidad, cuyo uso está altamente implementado en la producción de documentación técnica y científica. Este sistema utiliza comandos \TeX de código abierto, especialmente útil para la escritura científica de artículos de revistas, informes técnicos, libros y presentaciones de diapositivas.

Esta memoria ha sido desarrollada completamente en \LaTeX , aunque en lugar de utilizar el software, se ha utilizado la herramienta Overleaf [11], un editor online de LaTeX fácil de usar, que no necesita instalación ni actualizaciones, y permite trabajar en el proyecto desde cualquier dispositivo. Además, Overleaf incluye la opción *Auto Compile*, produciendo una salida en interfaz totalmente compilada a medida que se escribe, por lo que la visualización del resultado es inmediata.

4.2. Desarrollo: organización y visualización

4.2.1. Inicio

TEXTO EJEMPLO

4.2.2. Introducción a *Python*

TEXTO EJEMPLO

4.2.3. PyCharm

TEXTO EJEMPLO

4.2.4. Variables y operadores

TEXTO EJEMPLO

4.2.5. Estructuras de datos

TEXTO EJEMPLO

4.2.6. Estructuras de control

TEXTO EJEMPLO

4.2.7. Funciones

TEXTO EJEMPLO

4.2.8. Archivos y datos

TEXTO EJEMPLO

4.2.9. Graficación

TEXTO EJEMPLO

4.2.10. Ejercicios complementarios

TEXTO EJEMPLO

4.2.11. Biopython

TEXTO EJEMPLO

4.2.12. Acerca de

TEXTO EJEMPLO

4.3. Material de apoyo

4.3.1. Ejemplos prácticos

4.3.2. Ejercicios

4.3.3. Enlaces de interés

Capítulo 5

Conclusiones y trabajo futuro

Conclusions and future work

Bibliografía

- [1] *Latex: A Document Preparation System*, 2/E. Pearson Education, 1994.
- [2] Ekaba Bisong. *Google Colaboratory*, pages 59–64. 09 2019.
- [3] I Challenger-Pérez, Y Díaz-Ricardo, and RA Becerra-García. El lenguaje de programación python. *ciencias holguín*, 20 (2), 1-13, 2014.
- [4] Tom Christie. *Mkdocs. project documentation with markdown*, 2014.
- [5] Nello Cristianini and Matthew W Hahn. *Introduction to computational genomics: a case studies approach*. Cambridge University Press, 2006.
- [6] J. Cristóbal. *Crea páginas estáticas con GitHub: De manera ilimitada y gratuita*. Javier Cristóbal, 2015.
- [7] Universidad de Córdoba. Grado en bioquímica. <https://www.uco.es/ciencias/es/grado-bioquimica>. Acceso: 2022/05/10.
- [8] Universidad de Córdoba. Grado en bioquímica, competencias básicas. <http://www.uco.es/organiza/centros/ciencias/es/competencias-bioquimica>. Acceso: 2022/05/11.
- [9] Martin Donath. Material for mkdocs. <https://squidfunk.github.io/mkdocs-material/>. Acceso: 2022/05/02.
- [10] Manuel Jesús Marín Jiménez. *Guía Docente Informática aplicada a la Bioquímica*. Universidad de Córdoba, Curso 2020-2021.
- [11] John Hammersley & John Lees-Miller. Overleaf, editor de latex online. <https://www.overleaf.com>. Acceso: 2022/05/24.
- [12] The Daring Fireball Company LLC. Markdown: Syntax. <https://daringfireball.net/projects/markdown/syntax>. Acceso: 2022/05/24.
- [13] Nicholas M Luscombe, Dov Greenbaum, and Mark Gerstein. What is bioinformatics? an introduction and overview. *Yearbook of medical informatics*, 10(01):83–100, 2001.

- [14] E. Matthes and B.P. González. *Curso intensivo de Python, 2a edición: introducción práctica a la programación basada en proyectos*. Anaya Multimedia, 2020.
- [15] CA Michael Kennedy. Pycharm, python ide for professional developers, 2017.
- [16] Pablo Orviz Fernández. Introduction to git & github. 2022.
- [17] Tyrrell T.D. Ebenhard T. Schultz, M. La agenda 2030 y los ecosistemas - un documento para la discusión acerca de los vínculos existentes entre las metas de aichi para la diversidad biológica y los objetivos de desarrollo sostenible. *Swed-Bio, Stockholm Resilience Centre*, 2016.
- [18] Jonathan Yates. Python: Practical python programming for beginners and experts (beginner guide), 2016.

Apéndice A

Material digital de apoyo

A.1. Python script

A.2. Google Colab

A.3. Ficheros de datos

Apéndice B

Fichero mkdocs.yml

```
1 site_name: TUTORIAL PYTHON PARA BIOQUIMICOS
2 site_author: Victor J. Fernandez
3 site_url: https://victorffdez.github.io/
   Python_Biochemistry_Tutorial
4 repo_name: victorffdez/GitHub
5 repo_url: https://github.com/Victorffdez/
   Python_Biochemistry_Tutorial
6 edit_uri: ""
7
8 nav:
9   - Inicio: index.md
10  - Introduccion a Python: 2_Introduccion_python.md
11  - PyCharm: 3_Pycharm.md
12  - Variables y operadores: 4_Variables.md
13  - Estructuras de datos: 5_EstructurasDatos.md
14  - Estructuras de control: 6_EstructurasControl.md
15  - Funciones: 7_Funciones.md
16  - Archivos y datos: 8_ArchivosyDatos.md
17  - Graficacion: 9_Graficacion.md
18  - Ejercicios complementarios: 10_Ej_complementarios.md
19  - BioPython: 11_Biopython.md
20  - Acerca de: AcercaDe.md
21
22 theme:
23   name: "material"
24   logo: https://upload.wikimedia.org/wikipedia/commons/thumb/c/c3
       /Python-logo-notext.svg/1200px-Python-logo-notext.svg.png
25   favicon: https://upload.wikimedia.org/wikipedia/commons/thumb/c
       /c3/Python-logo-notext.svg/1200px-Python-logo-notext.svg.png
26   icon:
27     repo: fontawesome/brands/github
```

```

28     palette:
29         - scheme: default
30           toggle:
31               icon: material/toggle-switch
32               name: Switch to dark mode
33         - scheme: slate
34           primary: red
35           accent: indigo
36           toggle:
37               icon: material/toggle-switch-off-outline
38               name: Switch to light mode
39     language: es
40     features:
41         - navigation.instant
42         - navigation.top
43         - search.suggest
44         - search.highlight
45         - search.share
46         -
47     plugins:
48         - search:
49             lang:
50                 - es
51                 - en
52     extra:
53         search:
54             language: es, en
55         social:
56             - icon: material/twitter
57               link: https://twitter.com/Fac_CienciasUC0?ref_src=twsrc%5E
                    Egoogle%7Ctwcamp%5Eserp%7Ctwgr%5Eauthor
58             - icon: material/school
59               link: https://uco.es/
60             - icon: fontawesome/brands/linkedin
61               link: https://www.linkedin.com/school/facultadcienciasuco/?
                    originalSubdomain=es
62             - icon: fontawesome/brands/python
63               link: https://es.python.org/
64     generator: false #Elimina spam de mkdocs al final del documento
65
66     copyright: Copyright 2021-2022 Victor J. Fernandez. Todos los
67               derechos reservados.
68     markdown_extensions:

```

```
68 - attr_list
69 - md_in_html
70 - admonition
71 - pymdownx.details
72 - pymdownx.superfences
73 - pymdownx.emoji:
74     emoji_index: !!python/name:materialx.emoji.twemoji
75     emoji_generator: !!python/name:materialx.emoji.to_svg
76 - pymdownx.critic
77 - pymdownx.caret
78 - pymdownx.keys
79 - pymdownx.mark
80 - pymdownx.tilde
81 - pymdownx.highlight:
82     anchor_linenums: true
83 - pymdownx.inlinehilite
84 - pymdownx.snippets
85 - pymdownx.superfences
86 - tables
```


Apéndice C

Fichero Markdown

A continuación se muestra el apartado *Extracción y escritura de datos* del fichero Markdown correspondiente a la sección *Biopython* del tutorial. En este archivo, se puede observar algunos aspectos característicos de la sintaxis en Markdown comentados en el apartado 4.1.3.

```
1  ### ***Extraccion y escritura de datos***
2  El archivo FASTA utilizado de ejemplo empieza de la siguiente forma
3
4  '''
5  >HM590365.1 Viola tricolor voucher personal collection:I. Hiiesalu
6      74 tRNA-Leu (trnL) gene, partial sequence; chloroplast
7  GACTTAATTGGATTGAGCCTTGGTATGGAACTTACTAAGTGGATAA....
8  ....
9  '''
10 Como puede comprobar, el identificador es el primer argumento en la
11     descripcion, informacion que podemos utilizar para obtener una
12     lista con todos los IDs:
13
14 ''' py linenums="1"
15 from Bio import SeqIO
16 especies = []
17 for seq_record in SeqIO.parse("viola_tricolor.fasta", "fasta"):
18     especies.append(seq_record.description.split()[0])
19 print(especies)
20 '''
21
22 Esto te devuelve por pantalla la siguiente lista:
23
24 '''
```

```
23 ['HM590365.1', 'JZ084087.1', 'JZ084086.1', 'JZ084085.1', 'JZ084084.1',
24     ', ...]
25
26 !!! note "Lectura de otros formatos de archivo"
27
28     Le recomendamos que pruebe este metodo con archivos de EMBL o
29     GenBank.
30
31 La modificacion de datos en el archivo es muy simple. Veamos como
32     modificar por ejemplo el ID de la primera entrada:
33
34     ''' py linenums="1"
35     from Bio import SeqIO
36     entradas = SeqIO.parse("viola_tricolor.fasta", "fasta")
37     primera_entrada = next(entradas)
38     print(primera_entrada)
39
40     primera_entrada.id = "ID_NUEVO"
41     print(primera_entrada)
42     '''
```