



FL-1051 Procesamiento automatizado de textos en Filología y Lingüística

- Intensidad: 3 horas semanales
- Créditos: 3
- Modalidad: virtual
- Ciclo: II-2022
- Nivel: grado
- Profesor: Haakon S. Krohn (krohn.ucr@gmail.com)

1. Descripción general

En este curso se aborda la manipulación de grandes cantidades de texto en Filología y Lingüística como una tarea automatizable. Además, considerando las características propias del lenguaje humano, se resuelven problemas específicos del procesamiento de texto. También se establecen las bases para identificar las necesidades y soluciones en proyectos complejos que requieran tareas de programación en dichas disciplinas.

2. Objetivo general

Brindar las herramientas de programación necesarias para la resolución de tareas específicas de procesamiento automatizado de textos en Filología y Lingüística.

3. Objetivos específicos

Al final del curso, el estudiante será capaz de:

- Efectuar manipulaciones de textos por medio de un lenguaje de programación.
- Identificar tareas programables en Lingüística y Filología.
- Resolver problemas de programación básicos en Lingüística y Filología.
- Adaptar soluciones de programación de otras áreas temáticas a la Filología y la Lingüística.

4. Contenidos

1. Instalación y uso de Python y de entornos de desarrollo
2. Introducción a la programación en Python
3. Expresiones regulares
4. Aplicaciones de Python en Filología y Lingüística

5. Metodología

Este curso se impartirá de manera virtual. Se utilizará la plataforma institucional Mediación Virtual para la comunicación asincrónica entre estudiantes y el docente, así como para la asignación y la entrega de tareas y trabajos. Además, se hará uso del software Zoom para sesiones sincrónicas.

El curso tiene un componente teórico, el cual será tratado en clases expositivas por parte de la persona docente y se complementará con lecturas asignadas. El componente de mayor peso es el práctico, por lo que los estudiantes deberán participar activamente en las sesiones sincrónicas y realizar todas las asignaciones del curso. Las sesiones sincrónicas se llevarán a cabo para ver temas teóricos, discutir aspectos importantes de las asignaciones prácticas y aclarar dudas. Cada estudiante es responsable de ir avanzando en el curso dentro del cronograma planificado.

A lo largo de todo el curso, se asignarán tareas que deben entregarse dentro de plazos determinados. Asimismo, habrá un examen parcial y un trabajo final.

6. Cronograma

Fecha	Contenidos
16/08	Introducción a la programación Instalación y uso de Python y de entornos de desarrollo
23/08	Variables, expresiones y sentencias
30/08	Ejecución condicional
06/09	Funciones
13/09	Iteración
20/09	Iteración
27/09	Cadenas
04/10	Cadenas
11/10	Examen parcial
18/10	Archivos
25/10	Listas Diccionarios
01/11	Tuplas Expresiones regulares
08/11	Expresiones regulares
15/11	Importación (módulos, paquetes y bibliotecas)
22/11	Importación (módulos, paquetes y bibliotecas)
29/11	Conclusión del curso Entrega del trabajo final

7. Evaluación

Tareas: 25 %

Examen parcial: 40 %

Trabajo final: 35 %

8. Bibliografía

- Bird, Steven, Ewan Klein y Edward Loper. (2009). *Natural Language Processing with Python*. Sebastopol, California: O'Reilly.
- Brooker, Philip D. (2020). *Programming with Python for Social Scientists*. SAGE Publications.
- Fitzgerald, Michael. (2012). *Introducing regular expressions*. Sebastopol: O'Reilly.
- Führer, Claus, Jan Erik Solem y Olivier Verdier. (2016). *Scientific Computing with Python 3*. Birmingham: Packt Publishing.
- González Duque, Raúl. (2011). *Python para todos*. Bajo licencia Creative Commons Reconocimiento 2.5 España.
- Hill, Christian. (2015). *Learning Scientific Programming with Python*. Cambridge: Cambridge University Press.
- Kinsley, Harrison y Will McGugan. (2015). *Beginning Python Games Development* (2ª edición). Apress.
- Langtangen, Hans Petter. (2016). *A Primer on Scientific Programming* (5ª edición). Springer.
- Martelli, A. (2008). *Python: Guía de referencia*. Anaya Multimedia / O'Reilly. Madrid.
- Mehta, Hemant Kumar. (2015). *Mastering Python Scientific Programming*. Birmingham: Packt Publishing.
- Millstein, Frank. (2018). *Natural Language Processing with Python*. CreateSpace Independent Publishing Platform.
- Nelli, Fabio. (2018). *Python Data Analytics* (2ª edición). Apress.
- Peirce, Jonathan y Michael MacAskill. (2018). *Building experiments in PsychoPy*. SAGE Publications.
- Python Software Foundation. (2020a). *Python Module Index*. <https://docs.python.org/3/py-modindex.html>
- Python Software Foundation. (2020b). *The Python Package Index*. <https://pypi.org>
- Python Software Foundation. (2020c). *The Python Standard Library*. <https://docs.python.org/3/library/>
- Severance, Charles R. (2020). *Python para todos. Explorando la información con Python 3*.
- Sundnes, Joakim. (2020). *Introduction to Scientific Programming with Python*. Springer Open.
- Watt, A. (2005). *Beginning regular expressions*. Indianapolis: Wiley Publishing.