

## PROGRAMA DE CURSO

Nombre del curso (en castellano y en inglés)			
PROGRAMACIÓN COMPUTACIONAL			
Escuela	Carrera (s)		Código
Escuela de Ciencias Agroalimentarias, Animales y Ambientales ECA3	Ingeniería Ambiental		AMB2501
Semestre	Tipo de actividad curricular		
	OBLIGATORIA		
Prerrequisitos		Correquisitos	
Herramientas Computacionales (AMB1401)		NO TIENE	
Créditos SCT	Total horas a la semana	Horas de cátedra, seminarios, laboratorio, etc.	Horas de trabajo no presencial a la semana
5	8,3	4,5	3,8
Ámbito	Competencias a las que tributa el curso		Subcompetencias
1. Ámbito estudio y análisis multidimensional de sistemas, ambientes y territorio  3. Ámbito de Desempeño Profesional	1.3. Modelar, simular y proyectar escenarios presentes y futuros a través del uso de herramientas computacionales e ingenieriles  3.1. Investigar e incorporar, de manera continua, nuevos conocimientos, habilidades y tecnologías que le permiten profundizar, adaptar y/o generar formas distintas de abordar las situaciones propias de su profesión en el marco de un entorno de constante cambio.		No aplica
Propósito general del curso			
Este curso tiene por propósito que los estudiantes resuelvan problemas de diversa complejidad, siguiendo una ruta metodológica y generando programas capaces de dar respuestas a las distintas peticiones y finalidades de éstos. Los problemas estarán definidos en diversos dominios de aplicación. Los estudiantes podrán centrarse fundamentalmente en el desarrollo de una metodología de trabajo que los llevará a adquirir rigor procedimental para enfrentarse a la resolución de estas tareas en base al razonamiento algorítmico y lógico.			

Por ello, las clases tendrán una estructura teórico-práctica en las que se introducirán las nuevas temáticas a partir de problemas seleccionados.

**Resultados de Aprendizaje (RA)**

**RA 1:** Reconocer la arquitectura de los computadores y los lenguajes de programación.

**RA 2:** Aprende y aplica metodologías de análisis y resolución de problemas.

**RA 3:** Aprende y aplica algoritmos en la resolución de problemas de diversa índole y complejidad.

**RA 4:** Aprende y crea programas computacionales utilizando la herramienta de programación PseInt.

**RA 5:** Aprende y crea programas computacionales utilizando el lenguaje computacional Python aplicándolo al análisis de datos.

Número	RA al que contribuye la Unidad	Nombre de la Unidad	Duración en semanas
1	RA1	Arquitectura y funcionamiento de los computadores.	1
Contenidos		Indicadores de logro	
<p><b>Arquitectura y funcionamiento de los computadores.</b></p> <ul style="list-style-type: none"> <li>● Describir la estructura funcional de un computador.</li> <li>● Diferenciar los conceptos de Hardware y Software.</li> <li>● Diferenciar los conceptos de datos e información.</li> <li>● Diferenciar los tipos y niveles de lenguaje de programación.</li> </ul> <p><b>Análisis resolución de problemas:</b></p> <ul style="list-style-type: none"> <li>● Análisis de problemas y búsqueda de datos</li> <li>● Algebra de conjuntos para la resolución de problemas.</li> </ul>		<p>Conocer el funcionamiento y arquitectura de los computadores y comprender la diferencia entre hardware y software, así como la diferencia entre datos e información y como esta última genera conocimiento.</p>	

Número	RA al que contribuye la Unidad	Nombre de la Unidad	Duración en semanas
2	RA2	Elementos y estructuras de algoritmos:	2,6
Contenidos		Indicadores de logro	
<ul style="list-style-type: none"> <li>• Que es un Algoritmo y tipos de representación.</li> <li>• Describir los pasos de un algoritmo (entrada, proceso y salida) para representar procesos de negocio.</li> <li>• Identificar el diagrama de flujo que resuelve un problema.</li> <li>• Identificar estructuras de control secuencial y condicional (simple, compuesta y múltiple) para la formulación de algoritmos que resuelvan problemas planteados.</li> <li>• Identificar los ciclos (para, mientras y repetir) que permitan implementar los algoritmos que resuelvan problemas planteados.</li> <li>• Diferenciar expresiones aritméticas, relacionales y lógicas para la resolución de problemas.</li> <li>• Describir estructuras de control secuencial y condicional (simple, compuesta y múltiple) para la resolución de problemas.</li> </ul>		<p>Mediante el trabajo teórico/práctico se busca que los estudiantes identifiquen diferentes problemas planteados aplicando métodos que logren descomponerlos y estructurarlos de manera que permita abordarlos de forma secuencial, analizando: las posibles entradas del problema, el proceso que lo resuelva y sus salidas</p>	

Número	RA al que contribuye la Unidad	Nombre de la Unidad	Duración en semanas
3	RA3	Estructuras en Pseudocódigo	4,4
Contenidos		Indicadores de logro	
<ul style="list-style-type: none"> <li>• Describir las características del pseudocódigo y las herramientas a utilizar para implementar los algoritmos.</li> <li>• Describir los tipos de datos simples (numérico, alfanumérico y lógico) para implementar algoritmos.</li> <li>• Describir los identificadores, variables y constantes para programar un algoritmo.</li> <li>• Aplicar expresiones aritméticas, relacionales y lógicas para programar un algoritmo.</li> <li>• Aplicar estructuras de control secuencial y condicional (simple, compuesta y múltiple) para la formulación de algoritmos en pseudocódigo.</li> <li>• Aplicar los ciclos (para, mientras y repetir) que permitan implementar los algoritmos que resuelvan los problemas planteados.</li> <li>• Aplicar las variables de control (contadores y acumuladores), su inicialización, las condiciones de control, los incrementos y decrementos y los mínimos y máximos para programar un algoritmo.</li> <li>• Identificar y aplicar los arreglos (declaración e inicialización, tipos de datos de los elementos del arreglo, dimensión e índices) que permitan implementar algoritmos que resuelvan los problemas planteados.</li> </ul>		<p>Mediante trabajo teórico/práctico el estudiante identifica y aplica algoritmos en pseudocódigo que permitan resolver los problemas planteados, siguiendo la metodología aprendida en la unidad anterior.</p>	

Número	RA al que contribuye la Unidad	Nombre de la Unidad	Duración en semanas
4	RA4	Resolución de Problemas usando programación I:	6
Contenidos		Indicadores de logro	
<ul style="list-style-type: none"> <li>Identificar la nomenclatura del lenguaje.</li> <li>Aplicar programación Funcional con PseInt.</li> <li>Aplicar resolución de problemas con PseInt.</li> </ul>		Mediante trabajo teórico/práctico los estudiantes identificarán y aplicarán el proceso de desarrollo de programas, escribiendo y depurando algoritmos que satisfagan ciertas especificaciones, usando la herramienta de programación PseInt.	

Número	RA al que contribuye la Unidad	Nombre de la Unidad	Duración en semanas
5	RA5	Resolución de Problemas usando programación II con Python:	2
Contenidos		Indicadores de logro	
<ul style="list-style-type: none"> <li>Identificar la nomenclatura del lenguaje.</li> <li>Aplicar programación Funcional con Python.</li> <li>Aplicar programación imperativa con Python.</li> <li>Aplicar lectura de archivos y trabajo con grandes volúmenes de datos usando Python</li> </ul>		Mediante trabajo teórico/práctico los estudiantes identificarán y aplicarán el proceso de desarrollo de programas, escribiendo y depurando algoritmos que satisfagan ciertas especificaciones, usando el lenguaje de programación Python.	

Metodologías	Requisitos de Aprobación y Evaluaciones del Curso
<p><b>Teórica:</b> Sesiones a cargo del profesor de cátedra, en donde se realiza una exposición introduciendo los contenidos a ser estudiados durante el día, exponiendo un problema y realizando los pasos necesarios para resolverlo.</p> <p><b>Práctica:</b> Sesiones donde los alumnos tendrán que utilizar sus computadores para poner en práctica los contenidos y reforzar lo aprendido, bajo supervisión del docente.</p>	<p><b>Evaluaciones Formativa:</b> Se realizarán 2 evaluaciones formativas correspondientes a las unidades de “Conceptos informáticos” y “Resolución de Problemas usando programación”.</p> <p><b>Evaluaciones Sumativa:</b> Se realizarán 3 evaluaciones correspondientes a:</p> <ul style="list-style-type: none"> <li>● Evaluación 1, Conceptos informáticos y Elementos y estructuras de algoritmos 30%</li> <li>● Evaluación 2, Estructuras en Pseudocódigo 30%</li> <li>● Evaluación 3, Resolución de Problemas usando programación I y II 40% (Trabajo final práctico de programación)</li> </ul> <p><b>Evaluaciones recuperativas</b> Solo en caso de que la o el estudiante no haya rendido alguna(s) de las evaluaciones 1 o 2 y su(s) inasistencia(s) esté(n) debidamente justificada(s), el/la estudiante deberá rendir el examen, esta nota reemplazará la evaluación pendiente. La Evaluación 3 o Trabajo Final Práctico deberá ser entregado en los plazos establecidos en la planificación, de no ser entregado será considerado como inasistencia injustificada y sin opción a evaluación recuperación, salvo casos debidamente justificados de enfermedad por periodos extensos y que cubran todo el plazo de entrega.</p> <p><b>Asistencia:</b> La asistencia exigida para poder aprobar el curso, es de un 70% a clases de Cátedra, y 100% a Laboratorio/Seminarios. Las inasistencias injustificadas a una evaluación tendrán una nota mínima de 1,0. Por lo tanto, no son recuperables.</p> <p><b>Examen</b> Examen 30% de nota final (sumado al 70% de nota de presentación). Todo estudiante con nota de presentación igual o superior a 5.0, y sin notas bajo 3.95 en cualquier evaluación sumativa, será eximido de la obligación de rendir el examen final, en tal caso, la nota final corresponderá a la nota de presentación.</p>

### Bibliografía Fundamental

- Fundamentos de Programación. Luis Joyanes Aguilar, Luis Rodríguez Baena y Matilde Fernández Azuela. 2016 Editorial: McGraw-Hill.
- Fundamentos de Programación, Libro de Problemas. Luis Joyanes Aguilar, Luis Rodríguez Baena y Matilde Fernández Azuela. 2003 Editorial: McGraw-Hill.
- Ebel, Franck - Rohaut, Sébastien. Algoritmia - Técnicas fundamentales de programación, 2019. ENI.
- Python aplicaciones prácticas- Jorge Nolasco. 2016 Editorial Ra-Ma
- Algorithms. Sedgewick, Robert. 2011 (<http://libros-uoh.cl/ESCUELADEINGENIERIA/Algorithms/>)
- Python paso a paso. Ángel Pablo Hinojosa Gutiérrez. 2016. Editorial Ra-Ma
- Python Programming for the absolute beginner. Dawson, Michael. 2010 ISBN: 9781435455 (<http://libros-uoh.cl/ESCUELADEINGENIERIA/Pythonprogramming>)
- Python para todos, Raul Gonzalez Duque. libro se distribuye bajo una licencia Creative Commons Reconocimiento 2.5 España.( <http://mundogeek.net/tutorial-python>).
- Curso: Python para Principiantes. Eugenia Bahit, 2012. se distribuye bajo una Licencia Creative Commons Atribución- NoComercial-SinDerivadas 3.0 Unported. ([www.safecreative.org/work/1207302042960](http://www.safecreative.org/work/1207302042960))

### Bibliografía Complementaria

- Python software foundation, Python v3 Documentation, <http://docs.python.org/3/>.
- Ceder. The quick python book. Manning Publications Co., 2010.
- Downey, B. Think Python: How to think like a computer scientist. Green Tea Press, 2013

**Fecha última revisión:** 8 de marzo de 2024

**Programa visado por:** Escuela de Ciencias Agroalimentarias, Animales y Ambientales  
ECA3