



**Vicerrectoría Académica**  
**Dirección Curricular y de Docencia**  
**Formato para la Elaboración de Microdiseños de Cursos**

1 Identificación del Curso			
1.1 Código	1.2 Nombre del Curso	1.3 Pre-Requisito	1.4 Co-Requisito
021715	Programación		
1.5 No. Créditos	1.6 HAD	1.7 HTI	1.8 HAD:HTI
2	34	68	1:2
1.9 Horas presenciales aula clase	1.10 Horas presenciales laboratorio/Salida campo	1.11 Horas Virtuales	1.12 Total Horas HAD
2	0	0	34
Obligatorio <input type="checkbox"/>		Optativo <input type="checkbox"/>	
Teórico <input checked="" type="checkbox"/>	Practico <input checked="" type="checkbox"/>	Libre <input type="checkbox"/>	
Teórico/Practico <input checked="" type="checkbox"/>			
1.13 Unidad Académica responsable del Curso			
Ingeniería Ambiental y Sanitaria			
1.14 Área de Formación			
Ingeniería Aplicada			
1.15 Componente			No aplica <input type="checkbox"/>

2 Justificación del Curso
<p>La apropiación del pensamiento computacional (PC) para ingenieros ambientales y sanitarios es una competencia vital para su ejercicio profesional e investigativo. Es particular, el pensamiento computacional trae consigo la abstracción, descomposición de problemas en subproblemas, el reconocimiento de patrones y la creación de algoritmos (Casali et al., 2020). De manera análoga, el pensamiento computacional involucra el proceso de formular problemas y encontrar sus soluciones de manera que puedan llevarse a cabo por un agente que procesa información, como, por ejemplo, una máquina de cómputo. Asimismo, se hace concreto cuando se aprende a programar. (Rosas et al., s. f.).</p> <p>El desarrollo del pensamiento computacional en los estudiantes le aporta al cumplimiento de dos de los resultados de aprendizaje (RA) que el programa se ha propuesto alcanzar: el primero de estos es la capacidad de identificar, formular y resolver problemas aplicando los principios de la ingeniería, ciencias y las matemáticas; el segundo es la capacidad para desarrollar y realizar la experimentación adecuada, analizar y e interpretar datos y utilizar el juicio de ingeniería para sacar conclusiones.</p> <p>Finalmente, el PC y la programación se enfocan en el procesamiento y análisis de datos ambientales para generar información para la toma de decisiones en proyectos propios de la ingeniería ambiental y sanitaria.</p>

### 3 Competencias por Desarrollar

#### 3.1 Competencias Genéricas

- Capacidad de aprender aprender.
- Capacidad de trabajar en equipo.
- Resolución de problemas y toma de decisiones.
- Capacidad de buscar datos, gestionar e integrar la información.
- Capacidad de análisis, síntesis y de visión global.
- Compromiso con la calidad.
- Capacidad de comprender y expresarse oralmente y por escrito.
- Uso apropiado de las TIC.

#### 3.2 Competencias Específicas

- Propone algoritmos y Scripts computacionales que procesan y analizan datos ambientales para resolver problemas disciplinares.

### 4 Resultados de Aprendizaje del Curso

1. Comprende las estructuras de cómputo y de datos propios de los lenguajes de programación para aplicarlas en problemas computacionales e ingenieriles.
2. Propone algoritmos y programas para resolver problemas computacionales.
3. Procesa y analiza datos ambientales a través de algoritmos y programas como apoyo a los procesos de investigación.

### 5 Programación del Curso

Unidad Temática	Semana	Contenido de Aprendizaje	Evidencias	Actividades Aprendizaje	HAD		HTI		Total Horas
					Aula Clase	Espacio Virtual	Trabajo dirigido	Trabajo Independiente	
1	1	Presentación de la Asignatura, Datos, constantes y variables, expresiones aritméticas	Interpretar o plantea algoritmos para solucionar problemas.	Ejercicios prácticos, Trabajos individuales y colaborativos	2	0	1	4	7
1	2	Operadores lógicos, sentencia asignación, Sentencias de entrada y salida, Condicionales, Condicional Switch			2	0	1	4	7
1	3	Ejercicios prácticos			2	0	1	4	7
1	4	Talleres en clase	Taller algoritmos para	talleres prácticos, Trabajos	2	0	0	4	6

### Formato para la Elaboración de Microdiseños de Cursos

			solucionar problemas computacionales	individuales y colaborativos					
1	5	Parcial	Parcial en el computador o en medio escrito	Parcial individual o Colaborativo	2	0	0	0	2
2	6	Funciones	Interpreta o plantea Script en lenguajes de programación	Ejercicios prácticos, Trabajos individuales y colaborativos	2	0	1	4	7
2	7	Ciclos			2	0	1	4	7
2	8	Ejercicios prácticos			2	0	1	4	7
2	9	Talleres en Clase	Plantea Script en lenguajes de programación	Talleres individuales y colaborativos	2	0	0	4	6
2	10	Segundo Parcial	Parcial en el computador o en medio escrito	Parcial individual o Colaborativo	2	0	0	0	2
3	11	Arreglos unidimensionales y multidimensionales	Interpreta o plantea Script en lenguajes de programación Ejercicio computacional	Ejercicios prácticos, Trabajos individuales y colaborativos	2	0	1	4	7
3	12	Archivos y Estructuras, Clases y objetos			2	0	1	4	7
3	13	Clases y objetos			2	0	1	4	7
3	14	Ejercicios prácticos			2	0	1	4	7
3	15	Talleres en Clase	Taller computacional procesamiento de datos	Talleres individuales y colaborativos	2	0	1	4	7
3	16	Examen Final o Proyecto final	Parcial en el computador o en medio escrito	Parcial individual o Colaborativo	2	0	0	0	2
3	17	Recuperaciones	Recuperación en el computador o en medio escrito	Recuperación individual	2	0	0	0	2
<b>Total</b>					34	0	11	52	97
<b>Créditos Académicos</b>					<b>2</b>				

## 6 Prácticas de campo (Laboratorios y Salida de Campo)

Unidad Temática	Fundamentación Teórica	Evidencias	Actividades Aprendizaje	Recursos	Tiempo (h)	Semana

## Formato para la Elaboración de Microdiseños de Cursos

### 7 Mecanismos de Evaluación del Aprendizaje

Resultado de Aprendizaje	Mediación de Evaluación	Mecanismos, Criterios y/o Rúbricas	Semana de Evaluación
Comprende la lógica de programación y las estructuras de cómputo de los lenguajes de programación para aplicarlas en la resolución de problemas computacionales e ingenieriles	Pseudocódigo y Scripts entregados por los estudiantes	Rúbrica y actividades acordadas con el estudiante	Semana acordada con los estudiantes
Propone algoritmos y Scripts para resolver problemas computacionales.	Algoritmos y Scripts entregados por los estudiantes	Rúbrica y actividades acordadas con el estudiante	Semana acordada con los estudiantes
Procesa y analiza datos ambientales a través de algoritmos y Scripts como apoyo a los procesos de investigación.	Scripts e informe de resultados entregados por los estudiantes	Rúbrica y actividades acordadas con el estudiante	Semana acordada con los estudiantes

### 8 Valoración de los Resultados de Aprendizaje

Valoración	Sobresaliente	Destacado	Satisfactorio	Básico	No Cumplimiento
<b>Fundamentos Cualitativos</b>					
Desarrollo de la lógica de programación para resolver problemas computacionales.	Aplica la lógica de programación en la resolución de problemas computacionales	Analiza y modifica lógica de programación en algoritmos propuestos	Aplica la lógica de programación para resolver problemas computacionales	Comprende la lógica de programación de computadores.	No comprende la lógica de la programación de computadores.
Aplicación de algoritmos y Scripts para la resolución de problemas computacionales	Crea algoritmos y Scripts que resuelven problemas computacionales	Analiza y modifica algoritmos y Scripts que resuelven problemas computacionales	Aplica algoritmos y Scripts que resuelven problemas computacionales	Comprende algoritmos y Scripts que resuelven problemas computacionales	No comprende algoritmos y Scripts que resuelven problemas computacionales
Aplica las estructuras de cómputo para procesar datos ambientales.	Propone o Crea Scripts que procesan datos ambientales para la toma de decisiones	Analiza y modifica Scripts que procesan datos ambientales para la toma de decisiones	Aplica Scripts que procesan datos ambientales para la toma de decisiones	Comprende Scripts que procesan datos ambientales para la toma de decisiones	No Comprende Scripts que procesan datos ambientales para la toma de decisiones

### 9 Recursos Educativos y Herramientas TIC

N	Nombre	Justificación	Contenido de Aprendizaje
1	Aplicaciones Ofimáticas	Documentación y Socialización de los proyectos de aula	
2	Lenguaje de Programación Python, R o Matlab	Construcción de programas para análisis de datos	
3	Equipos de cómputo y videoBeam	Redacción y socialización del anteproyecto	
4	BrightSpace y Ms Teams	Comunicación e interacción con estudiantes	

## 10 Referencias Bibliográficas

- Casali, A., Deco, C., Viale, P., Bender, C., Zanmarini, D., & Monjelat, N. (2020). Enseñanza y aprendizaje del pensamiento computacional y la programación en los distintos niveles educativos. *XXII Workshop de Investigadores en Ciencias de la Computación (WICC 2020, El Calafate, Santa Cruz)*.
- Rosas, M. V., Zúñiga, M. E., Fernández, J. M., & Guerrero, R. A. (s. f.). El Pensamiento Computacional en el Ámbito Universitario. 4.


--

**Director de Programa**

--

**Decano Facultad**