

**Programa de asignatura  
Pensamiento Computacional**

**1. Identificación**

Nombre Escuela:	Ciencias aplicadas e Ingeniería
Nombre Departamento:	Ingeniería de sistemas
Nombre Programa:	Pensamiento computacional
Nombre Programa Académico:	Núcleo de Formación Institucional
Nombre Programa Académico (En inglés):	Computational thinking
Materia Prerrequisito	Ninguno
Semestre De Ubicación:	Libre
Código CINE:	XX
Código EAFIT:	ST0367
Intensidad Horaria Semanal	3 Horas
Intensidad Horaria Semestral	48 Horas
Créditos	3
Características	No suficientable

---

---

**2. Justificación**

La computación ha permeado todas las áreas de la vida moderna, no solo en cuanto a avances tecnológicos, sino también ha permitido resolver una gran cantidad de problemas globales. Esto debido a que la computación permite recolectar, filtrar, procesar y analizar gran cantidad de datos, y hasta simular sistemas complejos.

El pensamiento computacional es el proceso de pensamiento que involucra la formulación de un problema y la expresión de su solución de tal manera que una computadora (o un humano) pueda ejecutarla efectivamente (Wing, 2014).

En este curso, se busca que los estudiantes de cualquier área de estudio resuelvan problemas cuantitativos y centrados en datos, problemas que tienen un impacto social en el mundo real.

El estudiante aprenderá sobre los pilares del pensamiento computacional, tales como: descomposición de problemas, generalización y abstracción, lógica y algoritmia, reconocimiento de patrones y evaluación de soluciones. Además, el estudiante desarrollará habilidades y conocimientos básicos en programación. Lo cual le permitirá implementar programas de computador para resolver una amplia gama de problemas.

Finalmente, se buscará que el estudiante interactúe con personas de diferentes profesiones y se le motivará a reflexionar sobre cómo puede lograr un impacto social positivo a través del pensamiento computacional.

*“La ciencia, la sociedad y nuestra economía se beneficiarán de los descubrimientos e innovaciones producidos por una fuerza laboral capacitada para pensar computacionalmente”* (Wing, 2014).

### **3. Objetivo general de la asignatura**

Desarrollar habilidades de pensamiento computacional (descomposición de problemas, generalización y abstracción, lógica y algoritmia, reconocimiento de patrones y evaluación de soluciones) mediante la implementación de algoritmos para resolver problemas de la sociedad.

### **4. Competencias y resultados de aprendizaje**

Al culminar esta materia, el estudiante habrá fortalecido las siguientes competencias genéricas y habrá avanzado en los siguientes resultados de aprendizaje:

#### **4.1 Competencias genéricas:**

- PENSAMIENTO CRÍTICO: capacidad de analizar y evaluar la consistencia de los razonamientos
- PENSAMIENTO ANTICIPATORIO: capacidad para lidiar con la incertidumbre mediante la creación y evaluación de múltiples opciones futuras.
- PENSAMIENTO SISTÉMICO: capacidad para analizar sistemas complejos y pensar cómo están integrados dentro de distintos dominios y escenarios.

#### **4.2 Resultados de Aprendizaje**

- Explica los pilares del pensamiento computacional (descomposición de problemas, generalización y abstracción, lógica y algoritmia, reconocimiento de patrones y evaluación de soluciones).
- Aplica los pilares del pensamiento computacional en la resolución de problemas de la sociedad.
- Diseña algoritmos mediante diferentes técnicas para resolver problemas de la sociedad.
- Implementa algoritmos mediante la utilización de un lenguaje de programación para resolver problemas de la sociedad.

### **5. Contenidos**

**Unidad 1:** Introducción al pensamiento computacional y a la programación.

**Duración:** 2 semanas.

**Contenidos:**

- Introducción al pensamiento computacional.
- Introducción a la programación.
- Introducción a la algoritmia.
- Introducción a herramientas de programación.

**Unidad 2:** Variables, condicionales y ciclos.

**Duración:** 6 semanas.

**Contenidos:**

- Programación en Python.
- Variables, valores y tipos de datos.
- Condicionales.
- Ciclos.

**Unidad 3:** Tipos de datos y funciones.

**Duración:** 4 semanas.

**Contenidos:**

- Cadenas.
- Arreglos.
- Diccionarios.
- Funciones.

**Unidad 4:** Manipulación de datos y archivos.

**Duración:** 4 semanas.

**Contenidos:**

- Acumuladores y contadores.
- Manejo de archivos.
- Análisis de datos.
- Librerías tales como: Numpy, Matplotlib y/o Pandas.

## **6. Estrategias metodológicas y cronograma**

### **6.1 Metodología**

El desarrollo de la asignatura estará centrado en la participación activa del alumno. Para el desarrollo de las clases, el docente orientará las temáticas a tratar mediante una presentación inicial de las mismas. Con esta presentación, el alumno podrá adquirir la información necesaria para analizar y sintetizar conceptos, además de reconocer fundamentos. Posteriormente, el ejercicio de aprendizaje se trasladará al alumno para que pueda desarrollar su potencial de aprendizaje, mediante actividades de clase, ejercicios y de trabajo independiente fuera del aula que incluyen trabajo tanto grupal como individual, con el acompañamiento del docente y/o monitores.

Adicionalmente, se fomentará la participación de los alumnos en la clase mediante: a) lectura y discusión de material, b) solución de problemas relacionados con las temáticas tratadas, y c) realización de actividades prácticas en laboratorios virtuales de programación, incluyendo el análisis y la discusión de los resultados.

Lo anterior con el propósito de ofrecerle al alumno las condiciones que le permitan desempeñarse adecuadamente, enfrentando retos y generando soluciones pertinentes a las temáticas tratadas. Por otro lado, en el curso de la asignatura el alumno se enfrentará a la realización de dos proyectos enfocados en el desarrollo de unas aplicaciones informáticas que permitirán a el alumno integrar y aplicar el conocimiento adquirido, logrando por lo tanto un desempeño integral.

**6.2 Cronograma**

Mes 1		Mes 2		Mes 3		Mes 4	
Semanas 1-2	Semanas 3-4	Semanas 5-6	Semanas 7-8	Semanas 9-10	Semanas 11-12	Semanas 13-14	Semanas 15-16
<u>Unidad 1</u>	<u>Unidad 2</u>	<u>Unidad 2</u>	<u>Unidad 2</u>	<u>Unidad 3</u>	<u>Unidad 3</u>	<u>Unidad 4</u>	<u>Unidad 4</u>
Introducción al pensamiento computacional y a la programación.	Variables, condicionales y ciclos.	Variables, condicionales y ciclos.	Variables, condicionales y ciclos.	Tipos de datos y funciones.	Tipos de datos y funciones.	Manipulación de datos y archivos.	Manipulación de datos y archivos.

**7. Recursos**

**7.1 Locativos:**

El curso requiere una sala de laboratorios donde cada estudiante disponga de un computador.

**7.2 Tecnológicos:**

Los estudiantes deben tener a su disposición un computador con acceso a internet. Se utilizarán plataformas tales como: Colab y/o Moodle VPL (virtual programming lab).

**7.3 Didácticos:**

N/A.

## 8. Criterios de evaluación académica

Resultado de aprendizaje	Criterio de evaluación
1. Explica los pilares del pensamiento computacional (descomposición de problemas, generalización y abstracción, lógica y algoritmia, reconocimiento de patrones y evaluación de soluciones).	1.1. Identifica los pilares del pensamiento computacional.
	1.2 Clasifica los pilares del pensamiento computacional.
	1.3 Define los pilares del pensamiento computacional.
2. Aplica los pilares del pensamiento computacional en la resolución de problemas de la sociedad.	2.1 Identifica los elementos relevantes en problemas de la sociedad.
	2.2 Selecciona los pilares del pensamiento computacional que mejor aplican a los problemas de la sociedad presentados.
3. Diseña algoritmos mediante diferentes técnicas para resolver problemas de la sociedad.	3.1 Escribe pseudocódigo para representar soluciones a problemas de la sociedad.
	3.2 Dibuja diagramas para representar soluciones a problemas de la sociedad.
4. Implementa algoritmos mediante la utilización de un lenguaje de programación para resolver problemas de la sociedad.	4.1 Codifica programas de computador en un lenguaje de alto nivel.
	4.2 Implementa soluciones a problemas de la sociedad mediante el uso de lenguajes de programación.

## 9. Bibliografía

### Libros

- Amos, D., Bader, D., Jablonski, J. & Heisler, F. (2021). Python Basics: A Practical Introduction to Python 3. Real Python.
- Beecher, K. (2017). Computational Thinking: A beginner's guide to problem-solving and programming. BCS, The Chartered Institute for IT.
- Nelli, F. (2018). Python Data Analytics: With Pandas, NumPy, and Matplotlib. Apress.
- Sweigart, A. (2019). Automate the boring stuff with Python: practical programming for total beginners. No Starch Press.
- Theobald, O. (2021). Python for Absolute Beginners (Python for Data Science). Independently published.
- Zapotecatl López, J. L. (2018). Introducción al pensamiento computacional: conceptos básicos para todos.

## 10. Requisitos del proceso de aseguramiento de la calidad

**Versión número:** 02.  
**Fecha elaboración:** Agosto de 2022.  
**Responsable:** Daniel Correa Botero.