

**PROGRAMA DE PENSAMIENTO COMPUTACIONAL**

<b>Facultad:</b>	Ingeniería	<b>Departamento:</b>	Gestión de Proyectos y Sistemas
<b>Código:</b>	FBTSP04	<b>Asignatura:</b>	Pensamiento Computacional
<b>Créditos:</b>	3	<b>Tipo:</b>	X_Obligatoria Electiva
<b>Carrera (s) a la que pertenece:</b>	Ingeniería Eléctrica, Ingeniería Civil, Ingeniería Mecánica, Ingeniería Química, Ingeniería de Producción, Ingeniería de Sistemas, Matemáticas Industriales, Psicología, Educación, Idiomas Modernos, Comunicación Social y Empresarial, Turismo Sostenible, Estudios Liberales, Derecho, Estudios Internacionales.	<b>Trimestres:</b>	I
<b>Prerrequisito</b>	Ninguno	<b>Modalidad:</b>	Semi – Presencial
<b>Número de horas semanales</b>			
<b>En aula</b>	<b>Prácticas supervisadas</b>	<b>Laboratorio</b>	<b>Aprendizaje Autónomo</b>
2			4
<b>Coordinador:</b>	Christian Guillén Drija	<b>Fecha de actualización</b>	octubre 2025
<b>Revisado por:</b>	Margarita De Abreu		

1. **Justificación:** El desarrollo del pensamiento computacional ha adquirido una relevancia creciente en el ámbito educativo, debido a que el computador se ha constituido en una herramienta indispensable en todas las disciplinas del saber. Más aún, muchos investigadores sostienen que el desarrollo del pensamiento computacional coadyuva al reforzamiento de aspectos ligados al proceso de razonamiento, tales como pensamiento lógico, la resolución de problemas, el pensamiento divergente, el diseño y comprensión de sistemas, la heurística, la capacidad de abstracción, la identificación de relaciones causa-efecto a través de aproximaciones sucesivas y la sinéctica como base del trabajo en equipo; aspectos que deben moldear el perfil de los egresados de todas las carreras de la Universidad Metropolitana.

En este sentido, la asignatura de Pensamiento Computacional es un componente esencial en el currículum de las distintas carreras de la Universidad Metropolitana, el cual pretende que el estudiante desarrolle habilidades de resolución de problemas y trabajo colaborativo a través del aprendizaje significativo de programación. Se busca fortalecer competencias que son críticas en el ámbito laboral mediante métodos pedagógicos que implican gamificación y aprendizaje constructivista. Este curso prepara a los estudiantes para enfrentarse a situaciones complejas mediante la lógica, la descomposición de problemas, el reconocimiento de patrones, la abstracción y el diseño de algoritmos.

2. **Propósito:** Desarrollar en el estudiante habilidades fundamentales para resolver problemas mediante procesos de abstracción, algoritmos y pensamiento lógico. El curso busca que el alumno identifique y analice situaciones complejas, descomponiéndolas en pasos ejecutables para diseñar soluciones eficientes. A través del aprendizaje de estructuras de control y programación básica, se fomenta una base sólida que permite aplicar la computación como un aliado estratégico en su formación profesional integral.

3. **Objetivos**

**Objetivo General:** Reconocer el desarrollo del pensamiento computacional como una estrategia para reforzar los aspectos ligados al proceso de razonamiento lógico y resolución de problemas.

**Objetivos Específicos**

- Comprender los principios que conforman el pensamiento computacional.
- Caracterizar los elementos constitutivos que definen los algoritmos.
- Interpretar la dinámica de la programación orientada a objetos como una visión paradigmática.
- Comprender la programación modular como una estrategia para afrontar la complejidad de los sistemas de software.
- Analizar los requerimientos estratégicos para el desarrollo de proyectos de software.

4. **Resultados de Aprendizaje**

**RA1: Gestión y trabajo de equipo:** Capacidad para gestionar y trabajar de manera eficaz individualmente y en equipo, asumiendo roles de liderazgo, creando un entorno colaborativo e inclusivo, estableciendo metas, planificando tareas y cumpliendo objetivos (N1)

**RA5: Manejo de las TICS:** Aplicar herramientas y recursos tecnológicos para la generación y aplicación de soluciones innovadoras a problemas técnicos. (N1)

5. **Contenidos**

Tema	Contenido	Herramientas técnicas y actividades	Horas dedicadas
I: Pensamiento Computacional.	Pensamiento computacional. Aspectos del pensamiento computacional: Resolución de problemas, Descomposición de problemas complejos, Reconocimiento de patrones, Diseño de algoritmos y abstracción.	Ejercicios teóricos, casos de estudio en clase.	6
II: Definición de Algoritmo y elementos constitutivos.	Interfaz gráfica de usuarios. Concepto de algoritmo. Variables, datos y operadores. Estructuras de control. Eventos.	Clases por computadoras, laboratorio, trabajos.	14
III: Aproximación al paradigma de POO	Objetos. Reutilización de código. Interacción entre objetos. Abstracción.	Clases por computadoras, laboratorio, trabajos y proyectos.	14
IV: La programación modular.	Programación modular. Paralelismo, Listas. Funciones. La programación Modular como estrategia para afrontar la complejidad en la solución de problemas.	Clases por computadoras, laboratorio, trabajos y proyectos.	6
V: Desarrollo de proyectos de software.	Análisis de requerimientos. Desarrollo de un proyecto complejo de software. Programación en parejas. Revisión de proyectos de forma colaborativa.	Clases por computadoras, laboratorio, trabajos y proyectos.	8

6. **Métodos de aprendizaje: Clases magistrales interactivas:** Se combinan las clases tradicionales con actividades participativas como preguntas y respuestas, debates y resolución de ejercicios en clase. Lo anterior con el fin de mantener a los estudiantes activos y comprometidos con el aprendizaje. **Aprendizaje Basado en Problemas (ABP):** Los estudiantes trabajan en equipos para resolver escenarios y problemas aplicando los conocimientos teóricos a situaciones prácticas, desarrollar habilidades de pensamiento crítico y resolución de problemas, y trabajar en equipo. Presentan los resultados. **Aprendizaje Colaborativo:** Se fomenta el trabajo en equipo a través de actividades como debates, proyectos grupales y resolución de problemas en conjunto, promoviendo la comunicación, el intercambio de ideas y la construcción conjunta del conocimiento. **Uso de Herramientas Tecnológicas:** Se utilizan software y herramientas online para el análisis de datos, la creación de modelos de decisión y la simulación de escenarios.

#### 7. **Métodos de evaluación**

Aprendizaje en contacto con el docente ( 15%)	Aprendizaje práctico experimental (45%)	Aprendizaje autónomo (40 %)
Exposiciones, Participación en clases, Debates, Exámenes escritos u orales, Talleres, Defensa de proyectos, entre otros.	Resolución de problemas prácticos, Prácticas de laboratorio, Salidas de campo o visitas técnicas, Manejo de software especializado, Prototipado técnico, Estudios de caso técnicos, entre otros.	Elaboración de informes, Resolución de problemas y ejercicios, Ensayos de investigación, Creación de mapas conceptuales, Participación en foros, entre otros.

#### 8. **Referencias**

##### **Obligatoria:**

- Basogain, X., Olabe, M., & Olabe, J. (2015). *Pensamiento computacional a través de la programación: Paradigma de aprendizaje*. RED. *Revista de Educación a Distancia*, (46), 1-21. <https://doi.org/10.6018/red/46/6>
- Bordignon, F. R. A., & Iglesias, A. A. (2020). *Introducción al pensamiento computacional*. UNIPE: Editorial Universitaria; CLACSO. <https://biblioteca-repositorio.clacso.edu.ar/bitstream/CLACSO/2379/1/introduccion-pensamiento-computacional.pdf>
- Wing, J. M. (2006). *Computational thinking*. *Communications of the ACM*, 49(3), 33–35. <https://doi.org/10.1145/1118178.1118215>
- Zapata-Ros, M. (2015). *Pensamiento computacional: Una nueva alfabetización digital*. RED. *Revista de Educación a Distancia*, (46). <https://doi.org/10.6018/red/46/4>

##### **Adicional:**

- College Board. (2022). *AP Computer Science A: Course and exam description*. <https://apcentral.collegeboard.org/pdf/ap-computer-science-a-course-and-exam-description.pdf?course=ap-computer-science-a>