

**PROGRAMA DE ACTIVIDAD CURRICULAR PARA CURSOS DE FORMACIÓN GENERAL**

1) IDENTIFICACIÓN DE LA ACTIVIDAD CURRICULAR			
ACTIVIDAD CURRICULAR:	Introducción a la programación: De la teoría a la práctica.		
NOMBRE DOCENTE	Felipe Gómez		
CAMPUS (día y jornada am/pm)	Campus Rancagua. Martes 6:00 PM – 7:30 PM (Online)		
LINEA	Formación Transversal		
CÓDIGO	CFG1230	TIPO DE ACTIVIDAD	Electiva
CRÉDITOS SCT–Chile	3	SEMANAS	15
TIEMPO DE DEDICACIÓN SEMANAL			
TIEMPO DE DEDICACIÓN TOTAL	TIEMPO DE DOCENCIA DIRECTA	TIEMPO DE TRABAJO AUTÓNOMO	
4,5 hrs.	1,5 hrs.	3 hrs.	

2) DESCRIPCIÓN DE LA ACTIVIDAD CURRICULAR
<p>El objetivo de este curso es que los estudiantes obtengan una comprensión básica de la programación utilizando Python. Durante el curso, los estudiantes desarrollarán habilidades útiles para la automatización de tareas mediante la programación.</p> <p>Se incluirán ejemplos y ejercicios prácticos para ayudar a los estudiantes a comprender y aplicar los principios de programación. Los estudiantes comprenderán la importancia del pensamiento secuencial en la programación, identificarán y resolverán problemas lógicos y desarrollarán habilidades de para el desarrollo de programas (scripts) en Python.</p> <p>El curso fomenta un entorno de aprendizaje interactivo y práctico, asegurando que los estudiantes puedan aplicar los conocimientos adquiridos de manera efectiva.</p>

3) COMPETENCIA GÉNÉRICA Y RESULTADOS DE APRENDIZAJE	
COMPETENCIA GÉNÉRICA	RESULTADOS DE APRENDIZAJE
Pensamiento crítico	<ol style="list-style-type: none"> <li>1. Identificar los componentes que permiten el diseño de un algoritmo para la resolución de problemas.</li> <li>2. Comprender la sintaxis básica de Python, así como sus tipos de datos y estructuras de control para la ejecución de algoritmos.</li> <li>3. Programar scripts en Python que permiten la automatización de tareas básicas.</li> </ol>

4) UNIDADES DE APRENDIZAJE Y CONTENIDOS				
UNIDAD	SEMANAS	RESULTADOS DE APRENDIZAJE	INDICADORES DE LOGRO	CONTENIDOS
1. Algoritmos y flujogramas.	3	1. Identificar los componentes que permiten el diseño de un algoritmo para la resolución de problemas	Comprende la definición de algoritmo y cómo representarlos mediante diagramas de flujo.	<ul style="list-style-type: none"> <li>- Conceptos básicos de la programación y lenguajes de programación.</li> <li>- Definición de algoritmo.</li> <li>- Estructura y componentes de un diagrama de flujo.</li> </ul>
2. Introducción a la lógica.	2	1. Identificar los componentes que permiten el diseño de un algoritmo para la resolución de problemas	Comprende el uso de la lógica como representación de	<ul style="list-style-type: none"> <li>- Definición y uso de proposiciones lógicas.</li> <li>- Definición y uso de operaciones lógicas (negación, "y" y "o").</li> </ul>
3. Introducción a la programación con Python	5	2. Comprender la sintaxis básica de Python, así como sus tipos de datos y estructuras de control para la ejecución de algoritmos.	Elabora scripts básicos para solucionar problemas cotidianos en Python, logrando identificar los tipos de datos y estructuras de control pertinentes.	<ul style="list-style-type: none"> <li>- Tipos de datos simples ("Int", "float", "string" y "bool").</li> <li>- Estructuras de control ("if", "for" y "while").</li> <li>- Tipos de datos compuestos ("list", "tuple", "dict" y "set").</li> <li>- Definición de funciones.</li> </ul>
4. Programación avanzada con Python	2	3. Programar scripts en Python que permiten la automatización de tareas básicas.	Elabora scripts que permiten resolver y automatizar tareas manuales y mecánicas.	<ul style="list-style-type: none"> <li>- Lectura y escritura de archivos txt, csv y xls (Excel).</li> <li>- Gráficos de datos (histogramas, gráficos circulares, etc).</li> </ul>

#### 5) RECURSOS Y METODOLOGÍAS DE ENSEÑANZA Y APRENDIZAJE

Se realizarán clases expositivas que buscarán la participación de los estudiantes. Además, se realizarán ejemplos y ejercicios para resolver en conjunto que sirvan para reforzar e interiorizar los contenidos estudiados durante la clase. La principal herramienta para utilizar será el lenguaje de programación Python, así como su documentación oficial.

#### ESTRATEGIAS Y METODOLOGÍAS DE EVALUACIÓN

RESULTADO DE APRENDIZAJE	ACTIVIDAD O METODOLOGÍA DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN	PONDERACIÓN
RA 1	Prueba escrita. Algoritmos, flujogramas y lógica.	Pauta de evaluación.	40%

RA 2	Evaluación asincrónica. Ejercicios en Python.	Pauta de evaluación.	30%
RA 3	Evaluación asincrónica. Ejercicios avanzados en Python.	Pauta de evaluación.	30%

6) CONDICIONES DE EVALUACIÓN Y APROBACIÓN
Asistencia obligatoria de 75%, condición de aprobación. Promedio ponderado $\geq 4.0$ .

7) BIBLIOGRAFÍA OBLIGATORIA		
UNIDAD	REFERENCIAS BIBLIOGRÁFICAS	TIPO DE RECURSO
3	Python Software Foundation. (2021). Python language reference, version 3.9.	Digital
3	Van Rossum, G., & Python development team. (2018). Python Tutorial Release 3.6.4. Python Software Foundation.	Digital

8) BIBLIOGRAFÍA COMPLEMENTARIA		
UNIDAD	REFERENCIAS BIBLIOGRÁFICAS	TIPO DE RECURSO
2	Seymour Lipschutz & Marc Lipson, Matemáticas Discretas, Serie Schaum, McGraw Hill.	Digital

EQUIPO DOCENTE RESPONSABLE DEL DISEÑO	Felipe Gómez
RESPONSABLE(S) DE VALIDACIÓN	Coordinación de Programa Formación General
FECHA DE ENTRADA EN VIGENCIA	Marzo 2025