

ACTIVIDAD CURRICULAR DE FORMACIÓN

Facultad o Instituto	:	Ciencias de la Ingeniería
Carrera	:	Ingeniería Civil Informática

I. IDENTIFICACIÓN DE LA ACTIVIDAD CURRICULAR

Nombre	:	Programación					
Código	:	INF-123					
Semestre lectivo	:	II Semestre					
Horas	:	Presencial:	108	Autónomas:	132	TOTAL:	240
Créditos SCT	:	8					
Duración	:	Trimestral		Semestral:	x	Anual:	
Modalidad	:	Presencial:	x	Semi-presencial:		A Distancia:	
Área de Formación	:	Disciplinar:	x	General:		Profesional:	
						Práctica:	
Pre-requisito (Si los hubiese)	:	Introducción a la Computación					

II. DESCRIPCIÓN Y CARACTERIZACIÓN DE LA ACTIVIDAD CURRICULAR

La actividad curricular de Programación, se desarrolla en el segundo semestre del Plan de estudios, pertenece al área curricular de Formación Disciplinar, al ciclo inicial y es de carácter teórico-práctico.

Corresponde a la segunda actividad curricular en la línea de ciencias de la Computación, permite profundizar y aplicar los principales conceptos y técnicas de la programación.

Se espera que le estudiante aplique metodologías que permitan transformar un diseño algorítmico en un programa eficiente utilizando como concepto fundamental la modularidad y el refinamiento sucesivo. Se considera en su implementación la aplicación de métodos de prueba de programas, la detección de puntos críticos y su documentación. Los programas deben ser implementados en algún lenguaje de programación de alto nivel.

La metodología utilizada será con clases expositivas-participativas, laboratorios con apoyo de software, aprendizaje colaborativo, aprendizaje basado en problemas, aprendizaje en base a resolución de problemas y tutorías.

La evaluación será a través de pruebas escritas, Informes de talleres, de laboratorio y problemas.

III. COMPETENCIAS DEL PERFIL DE EGRESO ASOCIADAS A LA ACTIVIDAD CURRICULAR.

III.1 COMPETENCIAS PROFESIONALES.

COMPETENCIA	SUBCOMPETENCIA
Aplicar conocimientos de ciencias de ingeniería y ciencia de la computación en el ámbito profesional, utilizando pensamiento crítico y capacidad analítica.	Comprender las bases conceptuales del tratamiento científico y algoritmos utilizadas en la solución de problemas ingenieriles.

III.2 COMPETENCIAS GENÉRICAS.

COMPETENCIA	SUBCOMPETENCIA
Demostrar coherencia ética entre sus postulados valóricos y sus acciones, respetando los derechos humanos y participando activamente en las organizaciones comunitarias, haciendo primar la responsabilidad social desde una perspectiva cristiana.	Presentar un comportamiento ético íntegro, coherente entre el discurso valórico y la práctica habitual en el ejercicio de su tarea profesional, en un contexto de tolerancia y respeto por la diversidad.
Comunicar ideas, tanto en la lengua materna como en el idioma inglés, haciendo uso de las tecnologías de la información para desenvolverse en diversos escenarios, dando soluciones a diversas problemáticas de la especialidad.	Comunicarse de forma escrita en la lengua materna e inglés, de acuerdo a los marcos conceptuales haciendo uso de las tecnologías de la información en contextos propios de su profesión.

IV. RESULTADOS DE APRENDIZAJE - APRENDIZAJE ESPERADO.

RESULTADOS DE APRENDIZAJES
1.- Utilizar bibliotecas de manejo de string en la construcción de programas.
2.- Aplicar metodologías modulares en el diseño e implementación de programas eficientes y correctos.
3.- Utilizar técnicas básicas de documentación y prueba de programas que permitan la comprensión del producto desarrollado.
4.- Identificar las características de los principales algoritmos de búsqueda y ordenamiento de arreglos.

V. UNIDADES DE APRENDIZAJE Y EJES TEMÁTICOS

R. AP.	UNIDAD	EJE(S) TEMÁTICO(S)
1	Procesamiento de texto	Caracteres. Strings de caracteres (cadenas). Lectura de caracteres y cadenas. Comparación de strings. Conversión de caracteres/números. Búsqueda en string.
2	Modularidad	Refinamientos sucesivos. Determinación de Especificaciones. Aproximación a una solución inicial. Funciones y Procedimientos: identificación y jerarquización. Enunciado y comprobación solución final. Programación estructurada. Programación modular paso a paso. Recursividad. Estilos de programación. Estructura y definición de módulos. Comunicación, jerarquía y acoplamiento. Estructura y cohesión interna. Construcción de programas modulares.
3	Prueba y documentación	Prueba formal de programas. Diseño de datos de prueba. Puntos críticos. Objetivos de la documentación. Documentación externa: especificaciones, algoritmos, restricciones. Documentación interna: comentarios en el programa.
4	Búsqueda y ordenamiento	Algoritmos de búsqueda en arreglos: búsqueda secuencial y binaria. Algoritmos de ordenamientos internos: Bubble Sort, Inserción, Selección, Shell Sort, Merge Sort, Quick Sort, otros.

VI. ESTRATEGIAS DE ENSEÑANZA-APRENDIZAJE

De acuerdo al modelo educativo de la Universidad Católica del Maule, la metodología de trabajo para el desarrollo de la actividad curricular, se basa en un enfoque activo-participativo; esto implica entregar un rol protagónico al estudiante que es entendido como eje y centro de acción y quién a través de su participación activa y con orientaciones y lineamientos que le entrega el docente va construyendo su propio aprendizaje. Para lograr este objetivo, las distintas clases consideran una serie de estrategias metodológicas, previamente seleccionadas por el docente, tales como:

- Aprendizaje Colaborativo
- Aprendizaje Basado en problemas
- Aprendizaje en base a resolución de ejercicios y problemas el aula
- Aprendizaje en base a resolución de ejercicios y problemas con uso software en laboratorio computación.
- Método expositivo- participativo
- Ayudantías.

VII. PROCEDIMIENTOS DE EVALUACION DE APRENDIZAJES.

RESULTADO DE APRENDIZAJES	INDICADORES	INSTRUMENTO Y/O TÉCNICA EVALUATIVA	PONDERACIÓN (%)
1	<p>Diferencia un string de un carácter.</p> <p>Utiliza biblioteca para procesar strings.</p> <p>Desarrolla funciones propias para manejo de strings.</p> <p>Transforma string a número y viceversa.</p>	<p>Prueba Escrita/ Talleres laboratorio/ Rúbrica</p>	20%
2	<p>Divide un problema en subproblemas.</p> <p>Reconoce sub-problemas. Implementa funciones y/o procedimientos.</p> <p>Resuelve problemas a través de refinamiento sucesivo.</p> <p>Implementa programas con uso de funciones y paso de parámetros.</p>	<p>Prueba Escrita/ Talleres laboratorio/ Rúbrica</p>	40%

	Diferencia entre solución iterativa y recursiva. Implementa soluciones recursivas.		
3	Selecciona casos de prueba críticos. Define estrategia de prueba. Documenta programas de manera externa. Documenta programas de manera interna. Genera check list para documentar y probar.	Prueba Escrita/ Rúbrica	20%
4	Usa búsqueda secuencial. Usa búsqueda binaria. Identifica cuando se puede utilizar la búsqueda binaria. Reconoce características de los algoritmos de ordenamientos estudiados. Selecciona algoritmo de ordenamiento según problema.	Prueba Escrita/ Talleres laboratorio/ Rúbrica	20%

VIII. RECURSOS DE INFRAESTRUCTURA

Sala de clases, laboratorio de computación, computadores, proyectores, telones, pizarras amplias, compiladores e intérpretes, Biblioteca, Sistemas LMS-UCM.

IX. RECURSOS BIBLIOGRÁFICOS

	Autor, Título, Editorial, Año de Edición	Biblioteca donde se encuentra	N° Libros Disponibles
BÁSICA OBLIGATORIA	- Antonakos, James; Mansfields Kenneth, Programación estructurada en C, Prentice Hall, Madrid, España, 1997.	- - -	- - -

	<ul style="list-style-type: none"> - Gottfried, Byron, Programación en C, McGraw Hill, Madrid, España, 1997. - Joyanes Aguilar, Luis, Estructura de datos: algoritmos, abstracción y objetos, McGraw Hill, España, 1999. - Kernighan, B; Ritchie, D, El lenguaje de programación C, Editorial Pearson Educación, México, 1991. - Savitch, Walter, Resolución de problemas con C++, Prentice Hall, México, 2000. 		
COMPLEMENTARIA			

X. OTROS RECURSOS

Nombre Recurso	Tipo de Recurso
<i>Sublime text, code block, Compilador GLC, Intérprete de Python, Scratch, pseirt</i>	<i>software</i>