



UNIVERSIDAD DEL BÍO-BÍO
La Libertad del Conocimiento

UNIVERSIDAD DEL BÍO-BÍO
VICERRECTORÍA ACADÉMICA – DIRECCIÓN DE DOCENCIA

PROGRAMA DE ASIGNATURA

I. IDENTIFICACIÓN

Nombre asignatura: Introducción a la Programación		Período de Vigencia: 2013-2014
Código:		
Tipo de Curso: Obligatorio, Formación de Especialidad		

Carrera: Ingeniería Civil en Informática	Departamento: Ciencias de la Computación y Tecnologías de Información, Sistemas de Información	Facultad: Ciencias Empresariales
Nº Créditos SCT: 9	Total de horas: Cronológicas: 270 Pedagógicas: 405	Año / semestre: 1 / 1
Horas presenciales: 162 HT: 5 HP: 2 HL: 2		Horas trabajo autónomo: 243 HT: 5 HP: 4 HL: 4
Prerrequisitos: No Tiene		Correquisitos: No Tiene

II.- DESCRIPCIÓN

II.1 Presentación: Relación de la Asignatura con las Competencias del Perfil de Egreso

Esta asignatura permitirá al estudiante resolver problemas simples utilizando un lenguaje de programación. Se imparte en el primer semestre de la carrera.

Tributa a las subcompetencias específicas:

- Analizar las problemáticas de las organizaciones y de los individuos con el objeto de determinar requerimientos de software usando técnicas definidas para este propósito.
- Resolver problemas de programación utilizando lenguajes de programación y modelado de acuerdo a reglas y estándares existentes, y aplicando estrategias que aseguren la generación de soluciones eficientes.
- Construir aplicaciones de software, probando su funcionalidad y eficiencia, mediante el uso de arquitecturas, modelos, patrones, técnicas y herramientas de programación pertinentes para distintas plataformas.

Además, contribuye a la competencia genérica:

- Manifestar una actitud permanente de búsqueda y actualización de sus aprendizajes, incorporando los cambios sociales, científicos y tecnológicos en el ejercicio y desarrollo de su profesión.

II.2 Descriptor de competencias

Resolver problemas simples de programación utilizando un lenguaje de programación procedimental.

Resultados de Aprendizaje:

1. Explica los elementos sintácticos y semánticos de un lenguaje simbólico para posibilitar la construcción de algoritmos sintáctica y semánticamente correctos.
2. Reconoce datos de entrada, salida, reglas de validación y acciones de transformación a partir del enunciado de un problema para la confección del algoritmo que permite resolver el problema
3. Construye algoritmos representados en un lenguaje simbólico para resolver problemas simples que tienen una solución algorítmica.
4. Explica los elementos sintácticos y semánticos de un lenguaje de programación procedimental para posibilitar la construcción de programas sintáctica y semánticamente correctos.
5. Construye programas en un lenguaje de programación procedimental para resolver problemas que tienen una solución algorítmica.

II.3 Aprendizajes Previos

- Lee en forma comprensiva enunciados o instrucciones escritas.
- Resuelve problemas matemáticos sencillos contemplados en enseñanza media.
- Utiliza el computador a nivel de usuario.

III. RESULTADOS DE APRENDIZAJE

Resultados de Aprendizaje	Criterios de Evaluación	Contenidos conceptuales, procedimentales y actitudinales.
1. Explica los elementos sintácticos y semánticos de un lenguaje simbólico para posibilitar la construcción de algoritmos sintáctica y semánticamente correctos.	1.1 Identifica los elementos de un lenguaje simbólico que permite representar algoritmos. 1.2 Distingue los elementos presentes en un algoritmo representado mediante un lenguaje simbólico.	Conceptuales <ul style="list-style-type: none"> • Elementos de un lenguaje simbólico y su descripción sintáctica. • Reglas de combinación de los elementos de un lenguaje simbólico. Procedimentales <ul style="list-style-type: none"> • Trazado de algoritmos simples a través de una tabla de seguimiento Actitudinales <ul style="list-style-type: none"> • Valora la precisión necesaria presente en los algoritmos
2. Reconoce datos de entrada, salida, reglas de validación y acciones de transformación a partir del enunciado de un problema para la confección del algoritmo que permite resolver el problema	2.1 Define los datos de entrada a partir de la lectura del enunciado de un problema 2.2 Identifica las posibles reglas de validación requeridas en los datos de entrada definidos 2.3 Determina las acciones de transformaciones (pasos o instrucciones) generales a ser implementadas 2.4 Determina los datos de salida a partir del enunciado de un	Conceptuales <ul style="list-style-type: none"> • Concepto de problema algorítmico y algoritmo. • Conceptos de proceso, entrada, salida y su relación. • Mecanismos de validación de datos. • Operadores y expresiones lógicas y matemáticas. • Operadores Relacionales Procedimentales <ul style="list-style-type: none"> • Definición de datos de entrada requeridos para la resolución de un

Resultados de Aprendizaje	Criterios de Evaluación	Contenidos conceptuales, procedimentales y actitudinales.
	problema	<p>problema</p> <ul style="list-style-type: none"> • Identificación de posibles reglas de validación requeridas en los datos de entrada para la resolución de un problema • Determinación de acciones de transformaciones generales a ser implementadas en la confección del algoritmo. • Determinación de datos de salida solicitados en el enunciado de un problema <p>Actitudinales</p> <ul style="list-style-type: none"> • Lectura analítica y reflexiva de problemas presentados en forma de un enunciado • Criterios de rigurosidad en el desarrollo de las actividades
3. Construye algoritmos representados en un lenguaje simbólico para resolver problemas algorítmicos simples.	<p>3.1 Descubre los pasos o instrucciones que permiten obtener el conjunto de datos de salida a partir de un conjunto de datos de entrada utilizando un enfoque top-down.</p> <p>3.2 Construye expresiones lógicas necesarias en un algoritmo</p> <p>3.3 Transfiere una secuencia de pasos o instrucciones a un algoritmo mediante un lenguaje simbólico.</p> <p>3.4 Verifica el correcto funcionamiento del algoritmo construido.</p>	<p>Conceptuales</p> <ul style="list-style-type: none"> • Definición semántica de los elementos simples y complejos de un lenguaje simbólico (Secuencia, Selección e Iteración). • Tipos de datos básicos posibles de almacenar en un computador • Procedimiento de construcción de expresiones lógicas. • Técnica de seguimiento de algoritmos. • Técnica top-down y refinamiento sucesivo para la resolución de problemas. <p>Procedimentales</p> <ul style="list-style-type: none"> • Definición de pasos o instrucciones que permiten construir el algoritmo que resuelve el enunciado planteado • Construcción de algoritmos mediante el uso de un lenguaje simbólico. • Construcción de traza de un algoritmo para verificar correcto funcionamiento <p>Actitudinales</p> <ul style="list-style-type: none"> • Valoración de la importancia de identificar/describir adecuadamente un problema y construir algoritmos que los resuelvan. • Criterios de rigurosidad en el desarrollo de las actividades
4. Explica los elementos sintácticos y semánticos de un lenguaje de	<p>4.1 Identifica los elementos sintácticos y semánticos de un lenguaje de programación procedimental.</p> <p>4.2 Distingue los elementos</p>	<p>Conceptuales</p> <ul style="list-style-type: none"> • Concepto de lenguaje de programación y su uso. • Definición semántica de los elementos de un lenguaje programación.

Resultados de Aprendizaje	Criterios de Evaluación	Contenidos conceptuales, procedimentales y actitudinales.
programación procedimental para posibilitar la construcción de programas sintáctica y semánticamente correctos.	<p>procedimentales presentes en un programa representado mediante un lenguaje de programación</p> <p>4.3 Reconoce estructuras anidadas presentes en un programa</p> <p>4.4 Transforma algoritmos desarrollados a un lenguaje de programación procedimental.</p>	<ul style="list-style-type: none"> Tipos de datos simples y compuestos Elementos de un lenguaje de programación procedimental simples y complejos y su descripción sintáctica. Reglas de combinación de los elementos de un lenguaje de programación procedimental. <p>Procedimentales</p> <ul style="list-style-type: none"> Construcción de traza de un programa para verificar correcto funcionamiento <p>Actitudinales</p> <ul style="list-style-type: none"> Valoración de la precisión de las instrucciones que componen un programa Criterios de rigurosidad en el desarrollo de las actividades
5. Construye programas en un lenguaje de programación procedimental para resolver problemas de programación simples.	<p>5.1 Organiza las instrucciones de un programa mediante un lenguaje de programación procedimental, a partir de un algoritmo, usando una herramienta de edición.</p> <p>5.2 Verifica el correcto funcionamiento de un programa procedimental utilizando las herramientas de compilación/interpretación y ejecución correspondientes.</p> <p>5.3 Construye funciones que permiten descomponer el programa a implementar.</p>	<p>Conceptuales</p> <ul style="list-style-type: none"> Procedimiento de construcción de instrucciones de programa. Técnicas de seguimiento y prueba de programas. Concepto de compilación / interpretación y ejecución de programas. Herramientas de compilación/interpretación y ejecución de programas. Opciones de una herramienta de compilación/interpretación, ejecución de programas y estrategias para su uso. <p>Procedimentales</p> <ul style="list-style-type: none"> Construcción de programas que permiten resolver problemas simples Construcción y uso de funciones. Construcción de traza de un programa para verificar correcto funcionamiento <p>Actitudinales</p> <ul style="list-style-type: none"> Valoración de la importancia de construir programas correctos que no dañen a terceros. Criterios de rigurosidad en el desarrollo de las actividades

IV. ACTIVIDADES DE APRENDIZAJE Y EVALUACIÓN

Resultados de Aprendizaje	Actividades de Aprendizaje	Actividades de Evaluación	Tiempo Estimado
1. Explica los elementos sintácticos y semánticos de un lenguaje simbólico para posibilitar la construcción de algoritmos sintáctica y semánticamente correctos.	1.1 Docente <ul style="list-style-type: none"> Entrega y socialización del programa de asignatura. Obligaciones del profesor y deberes de los estudiantes Presenta tópicos con apoyo audiovisual y apuntes Proporciona material a través de plataforma virtual Entrega listado de ejercicios 1.2 Estudiante <ul style="list-style-type: none"> Participa activamente en clases Estudia apuntes y bibliografía recomendada 	Test 1	Horas presenciales: HT: 6 HP: 6 HL: 4 H Ayudantía: 4 Horas trabajo autónomo: HT: 6 HP: 12 HL: 8 H Ayudantía: 4
2. Reconoce datos de entrada, salida, reglas de validación y acciones de transformación a partir del enunciado de un problema para la confección del algoritmo que permite resolver el problema	2.1 Docente <ul style="list-style-type: none"> Presenta tópicos con apoyo audiovisual y apuntes Proporciona material a través de plataforma virtual Entrega listado de ejercicios 2.2 Estudiante <ul style="list-style-type: none"> Participa activamente en clases Resuelve guías de ejercicios Estudia apuntes y bibliografía recomendada 	Test 2 Tarea 1	Horas presenciales: HT: 6 HP: 6 HL: 4 H Ayudantía: 4 Horas trabajo autónomo: HT: 6 HP: 12 HL: 8 H Ayudantía: 4
3. Construye algoritmos representados en un lenguaje simbólico para resolver problemas algorítmicos simples.	3.1 Docente <ul style="list-style-type: none"> Presenta tópicos con apoyo audiovisual y apuntes Proporciona material a través de plataforma virtual Entrega listado de ejercicios 3.2 Estudiante <ul style="list-style-type: none"> Participa activamente en clases Resuelve guías de ejercicios Utiliza software de apoyo para la implementación de algoritmos (Ejemplo: PsInt – DFDSmart) Estudia apuntes y bibliografía recomendada. 	Test 3 Tarea 2 Trabajo en equipo Certamen 1	Horas presenciales: HT: 18 HP: 18 HL: 12 H Ayudantía: 12 Horas trabajo autónomo: HT: 18 HP: 36 HL: 24 H Ayudantía: 12

<p>4. Explica los elementos sintácticos y semánticos de un lenguaje de programación procedimental para posibilitar la construcción de programas sintáctica y semánticamente correctos.</p>	<p>4.1 Docente</p> <ul style="list-style-type: none"> • Presenta tópicos con apoyo audiovisual y apuntes • Proporciona material a través de plataforma virtual • Entrega listado de ejercicios <p>4.2 Estudiante</p> <ul style="list-style-type: none"> • Participa activamente en clases • Resuelve guías de ejercicios • Estudia apuntes y bibliografía recomendada 	<p>Test 4</p>	<p>Horas presenciales: HT: 6 HP: 6 HL: 4 H Ayudantía: 4</p> <p>Horas trabajo autónomo: HT: 6 HP: 12 HL: 8 H Ayudantía: 4</p>
<p>5. Construye programas en un lenguaje de programación procedimental para resolver problemas de programación simples.</p>	<p>5.1 Docente</p> <ul style="list-style-type: none"> • Presenta tópicos con apoyo audiovisual y apuntes • Proporciona material a través de plataforma virtual • Entrega listado de ejercicios <p>5.2 Estudiante</p> <ul style="list-style-type: none"> • Participa activamente en clases • Resuelve guías de ejercicios • Utiliza compilador de lenguaje de programación para la implementación de programas • Estudia apuntes y bibliografía recomendada 	<p>Test 5 Tarea 3 Trabajo en equipo Certamen 2</p>	<p>Horas presenciales: HT: 18 HP: 18 HL: 12 H Ayudantía: 12</p> <p>Horas trabajo autónomo: HT: 18 HP: 36 HL: 24 H Ayudantía: 12</p>

V. SISTEMA DE EVALUACIÓN

- 30% Certamen 1 y 30% Certamen 2
- 25 % Promedio de test y trabajos en equipo
- 15 % Promedio Tareas (incluye informe y desarrollo de algoritmos y programas).

VI. BIBLIOGRAFÍA

Fundamental

Deitel, H. M. y Deitel, P. J. (2003). *Como programar en C/C++* (4ª ed.): Pearson Educación.
 Joyanes, L. (2002). *Programación en C: libro de problemas*: McGraw Hill.
 Joyanes, L. (2006). *Programación en C++: un enfoque práctico*: McGraw Hill.

Complementaria

Sánchez, M. y Chamorro, F. (1996). *Programación Estructurada y Fundamentos de Programación*: McGraw Hill.