

**PROGRAMA DE CURSO  
INTRODUCCIÓN A LA PROGRAMACIÓN**

**A. Antecedentes generales del curso:**

Departamento	Ciencias de la Computación				
Nombre del curso	<b>Introducción a la Programación</b>				
Nombre del curso en inglés	<b><i>Introduction to Programming</i></b>				
Código	CC1002		Créditos	6	
Horas semanales	Docencia	3,5	Auxiliares	2,0	Trabajo personal 4,5
Carácter del curso	Obligatorio	X		Electivo	<b>0</b>
Requisitos	Sin requisitos				

**B. Propósito del curso:**

Este curso tiene como propósito que los estudiantes resuelvan problemas utilizando la programación, siguiendo una ruta metodológica determinada, traduciendo, reformulando y formalizando enunciados con el propósito de generar programas capaces de dar respuestas a las distintas peticiones y finalidades.

El contexto del desarrollo de habilidades de aplicación metodológica será a través de problemas específicos, definidos en diversos dominios de aplicación cuyas soluciones se encontraran delimitadas en cuanto al alcance y tamaño.

Se espera que los estudiantes desarrollen una metodología de trabajo que los lleve a adquirir rigor procedimental para enfrentarse a la resolución de problemas, en base al razonamiento algorítmico y lógico.

Las competencias específicas (CE) y genéricas (CG) a las que tributa el curso son:

**CE6:** Aplicar una metodología de diseño e implementación para escribir programas computacionales, en la resolución de problemas, utilizando herramientas computacionales para manejar y visualizar datos.

**CE7:** Modelar matemáticamente el comportamiento de los agentes (consumidores, empresas, organizaciones, países, entre otros), aplicando conceptos fundamentales del análisis económico para la toma de decisiones.

**CG2 Compromiso Ético:**

Reflexionar sobre el propio actuar y sus consecuencias, en el marco de la honestidad, la responsabilidad y el respeto, buscando la excelencia y rigurosidad en su proceder en contextos académicos, en las relaciones interpersonales y con su entorno.

**C. Resultados de aprendizaje:**

Competencias específicas	Resultados de aprendizaje
CE6-CE7	RA1: Descompone analíticamente un problema enunciado, deduciendo los datos de entrada, de salida, o efectos esperados de un programa y derivando sus posibles ejemplos de uso, a fin de llegar a la descomposición irreductible del problema.
CE6	RA2: Implementa programas computacionales a partir de la descomposición del problema y de los elementos existentes, para obtener una solución ejecutable al problema.
CE6	RA3: Verifica la solución implementada a partir del comportamiento esperado, con el fin de validar y/o rectificar dichas implementaciones.
Competencias genéricas	Resultados de aprendizaje
CG2	RA4: Realiza las tareas de manera responsable y honesta, sin incurrir en plagio, copia o suplantación de identidad.

### C. Unidades temáticas:

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
1	RA1, RA2, RA3, RA4	Fundamentos de Programación	4 semanas
Contenidos		Indicador de logro	
1. Expresiones y tipos de datos básicos 2. Funciones 3. Diseño de programas 4. Programación modular 5. Expresiones booleanas y condicionales 6. Recursión 7. Testing y depuración		El estudiante: <ol style="list-style-type: none"> <li>1. Identifica y escribe expresiones usando los tipos de datos básicos (entero, real, texto, boolean).</li> <li>2. Reconoce el concepto de función en programación.</li> <li>3. Explica el concepto de módulo, entendiéndolo como un conjunto de funciones relacionadas entre sí.</li> <li>4. Utiliza operadores relacionales y conectores lógicos para escribir expresiones booleanas.</li> <li>5. Utiliza recursión en la resolución de problemas.</li> <li>6. Analiza los elementos de un problema propuesto: datos de entrada y sus tipos de datos básico, dato de salida y propósito, a fin de facilitar la descomposición del problema.</li> <li>7. Descompone el problema de ser necesario en subproblemas, considerando el propósito identificado.</li> <li>8. Identifica las funciones que resuelven el problema a partir del análisis realizado.</li> <li>9. Utiliza la "Receta de diseño" (Felleisen M., 2001) en la programación de una función.               <ol style="list-style-type: none"> <li>9.1 Documenta su propósito principal.</li> <li>9.2 Escribe su firma (nombre, tipos de argumentos y tipo de retorno).</li> <li>9.3 Escribe programas de ejemplos de su uso comentando resultados esperados.</li> <li>9.4 Programa el cuerpo de la función usando los elementos identificados previamente.</li> <li>9.5 Utiliza nombres representativos para las funciones y variables.</li> <li>9.6 Verifica que las funciones tengan un propósito único.</li> <li>9.7 Verifica que la función esté correctamente programada a través de la aplicación de testing, y corrige el código en caso de detectar errores.</li> </ol> </li> </ol>	

	<p>10. Cumple obligaciones y acuerdos, respetando los compromisos adquiridos en sus actividades académicas.</p> <p>11. Planifica y presenta sus trabajos, basándose en sus capacidades, sin incurrir en plagio, copia, suplantación de identidad.</p>
<b>Bibliografía de la unidad</b>	[1] Capítulos 2, 3, 4, 9.

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
2	RA1, RA2, RA3, RA4	Programación Funcional	4 semanas
Contenidos		Indicador de logro	
<p>1. Datos compuestos</p> <p>2. Estructuras de datos recursivas</p> <p>3. Abstracción funcional</p>		<p>El estudiante:</p> <ol style="list-style-type: none"> <li>Define y programa tipos de datos compuestos.</li> <li>Define estructuras de datos recursivas.</li> <li>Resuelve problemas utilizando estructuras de datos recursivas.</li> <li>Reconoce el concepto de abstracción funcional.</li> <li>Utiliza abstracción funcional para combinar funciones relacionadas en una única función.</li> <li>Programa funciones que consideran datos compuestos y estructuras de datos recursivas, utilizando la Receta de diseño (Felleisen M., 2001).</li> <li>Cumple obligaciones y acuerdos, respetando los compromisos adquiridos en sus actividades académicas.</li> <li>Planifica y presenta sus trabajos, basándose en sus capacidades, sin incurrir en plagio, copia, suplantación de identidad.</li> </ol>	
<b>Bibliografía de la unidad</b>		[1] Capítulos 6, 9, 10, 12, 14, 15, 17, 19, 20, 21, 22, 29.	

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
3	RA1, RA2, RA3, RA4	Programación Imperativa	2 semanas
Contenidos		Indicador de logro	
1. Mutación y aliasing 2. Estructuras indexadas 3. Archivos de texto 3. Testing y depuración en programación imperativa.		El estudiante: <ol style="list-style-type: none"> <li>1. Identifica los conceptos de mutación y aliasing.</li> <li>2. Utiliza variables de estado para diseñar funciones con memoria.</li> <li>3. Identifica los efectos de una función en las variables de estado.</li> <li>4. Define y programa estructuras de datos mutables.</li> <li>5. Programa funciones que modifican estructuras mutables.</li> <li>6. Identifica los efectos de una función en estructuras mutables.</li> <li>7. Reconoce estructuras indexadas como una herramienta de programación.</li> <li>8. Utiliza estructuras indexadas en la resolución de problemas.</li> <li>9. Utiliza archivos de texto para guardar/leer información desde la memoria secundaria.</li> <li>10. Programa funciones que consideran: variables de estado, estructuras de datos mutables, efectos de las funciones en las variables de estado y estructuras mutables y estructuras indexadas, utilizando la "Receta de Diseño" (Felleisen M., 2001).</li> <li>11. Cumple obligaciones y acuerdos, respetando los compromisos adquiridos en sus actividades académicas.</li> <li>12. Planifica y presenta sus trabajos, basándose en sus capacidades, sin incurrir en plagio, copia, suplantación de identidad.</li> </ol>	
Bibliografía de la unidad		[1] Capítulos 35, 36, 40, 41, 42.	

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
4	RA1, RA2, RA3, RA4	Programación Orientada al Objeto	5 semanas
Contenidos		Indicador de logro	
1. Conceptos básicos de Programación Orientada al Objeto 2. Definición de clases 3. Interacciones entre objetos		El estudiante: <ol style="list-style-type: none"> <li>1. Identifica los conceptos de clase y objeto.</li> <li>2. Utiliza objetos en la resolución de problemas.</li> <li>3. Define una clase y sus componentes: campos, constructor, métodos accesores y mutadores.</li> <li>4. Identifica interacciones entre objetos a través de llamadas de método interna y externa.</li> <li>5. Programa funciones considerando clases y objetos, utilizando la "Receta de Diseño" (Felleisen M., 2001).</li> <li>6. Cumple obligaciones y acuerdos, respetando los compromisos adquiridos en sus actividades académicas.</li> <li>7. Planifica y presenta sus trabajos, basándose en sus capacidades, sin incurrir en plagio, copia, suplantación de identidad.</li> </ol>	
Bibliografía de la unidad		[2] Capítulos 1, 2, 3.	

#### D. Estrategias de enseñanza:

La metodología de enseñanza y aprendizaje fomenta la participación del estudiante en el aula, las clases son principalmente:

- Clase expositiva, en donde el estudiante identifica las herramientas para la programación y realizan ejercicios en papel.
- Casos de estudio. En cada unidad el estudiante es expuesto a problemas de la vida cotidiana, en donde logra usar las herramientas de programación.

A lo anterior se le suman las tareas que deben ser desarrolladas con el computador, estas son enviadas a través de u-cursos.

### E. Estrategias de evaluación:

El curso tiene distintas instancias de evaluación de proceso entre ellas:

- Control 1 Unidad 1
- Control 2 la unidad 1 y 2
- Control 3 la unidad 1, 2, 3 y 4
- 60%
- Examen Unidad 1, 2, 3 y 4 (40%)
  
- Los controles son controles escritos.
  
- Tareas existe un mínimo de 4 tareas. (1/3)

### F. Recursos bibliográficos:

#### Bibliografía obligatoria:

[1] Matthias Felleisen, Robert Bruce Findre, Matthew Flatt, Shriram Krishnamurthi. How to Design Programs: An Introduction to Programming and Computing. The MIT Press, 2001.

[2] David Barnes, Michael Kölling. Objects First with Java: A Practical Introduction Using BlueJ. Prentice Hall, 2012.

#### Bibliografía Complementaria:

- Gutiérrez, F; Peña V.; Quezada M.; Bustos B.; Robbes R. (2019 en revisión) Apuntes CC1002 Introducción a la Programación, Santiago.

### G. Datos generales sobre elaboración y vigencia del programa de curso:

<b>Vigencia desde:</b>	2019
<b>Elaborado por:</b>	Benjamin Bustos, Romain Robbes, Eric Tanter
<b>Validado por:</b>	
<b>Revisado por:</b>	Área de Gestión Curricular