

Programa de Asignatura



1 Identificación Asignatura

Nombre:	Programación I: Introducción		Código:	IN1010
Carrera:	Ingeniería Civil Industrial	Unidad Académica:	Ciencias Naturales y Tecnología	
Ciclo Formativo:	Ciclo Inicial	Línea formativa:	Básica	
Semestre:	II	Tipo de actividad:	Obligatoria	
N° SCT:	6	Horas Cronológicas Semanales		
		Presenciales:	6	Trabajo Autónomo:
Pre-requisitos	IN1004 - Herramientas informáticas.			

2 Propósito formativo

La asignatura de Programación I se ubica en el ciclo formativo inicial y corresponde a la línea formativa básica. Tiene como propósito que el estudiante desarrolle el razonamiento algorítmico y lógico, a través de un proceso de análisis, diseño e implementación de soluciones a problemas de ingeniería, y usando como herramienta programas computacionales escritos en un lenguaje de alto nivel y de propósito general.

Para cumplir con el propósito mencionado, se abordarán problemas en donde la complejidad de sus soluciones evolucionará desde cálculos simples hasta el diseño de estructuras de datos con responsabilidades y funcionalidades debidamente modularizadas e integradas, en donde se deban utilizar paradigmas de programación específicos, y donde se deban diseñar componentes reutilizables basados en dichos paradigmas.

Esta asignatura aporta a la formación integral del estudiante desde dos perspectivas. Por un lado, los conocimientos sobre programación y lenguajes entregados en esta asignatura le permitirán tener una noción general de las características de las tecnologías utilizadas en proyectos que involucran el desarrollo de sistemas y tecnologías de información, y los desafíos involucrados en su utilización. Por otro lado, los elementos metodológicos que se aplican en esta asignatura le permitirán al estudiante incorporar estrategias eficientes para abordar problemáticas complejas de forma sistemática, desde la perspectiva de la ingeniería, aplicando habilidades de abstracción, técnicas de descomposición y priorización sobre los problemas abordados.

Esta asignatura profundiza en los contenidos vistos previamente en Herramientas Informáticas, en lo que se refiere a la utilización de herramientas con enfoque práctico, y al mismo tiempo profundiza en el análisis, diseño e implementación de nuevas herramientas de propósito específico. A su vez, esta asignatura sienta las bases fundamentales para otras asignaturas que están asociadas a tecnologías de información, especialmente con Programación II, ya que introduce elementos de la informática y computación que dichas asignaturas necesitan.

3 Contribución al perfil de egreso

Esta asignatura contribuye a los siguientes desempeños o resultados de aprendizaje globales declarados en el Perfil de Egreso de la carrera:

- Demuestra un sólido dominio de las ciencias básicas y de las ciencias de la ingeniería
- Concibe soluciones a los problemas que surgen en las organizaciones
- Diseña, selecciona y adapta desarrollos tecnológicos y científicos propios de la ingeniería industrial a los desafíos de las organizaciones

4 Resultados de aprendizaje específicos (verbo + objeto + condición + finalidad)

- RA1. Diseña algoritmos de pequeña a mediana envergadura que le permitan resolver sistemáticamente problemas básicos de ciencias e ingeniería.
- RA2. Implementa funciones en concordancia a una receta de diseño que induce la especificación de su contrato, objetivo y ejemplos para su uso.
- RA3. Desarrolla programas interactivos, con base a una especificación concreta de funcionalidades, entradas y salidas, para construir mecanismos de interacción entre los usuarios y los programas desarrollados.
- RA4. Diseña componentes de software utilizando funciones y procedimientos, de forma que se puedan aplicar sus conceptos fundamentales al desarrollo de programas.
- RA5. Implementa estructuras de datos de complejidad básica a mediana para su aplicación en programas de distinta naturaleza.

5 Unidades de Aprendizaje

1. Introducción a la programación

- 1.1. Motivación
- 1.2. Conceptos de algoritmos
- 1.3. Lenguajes de programación
- 1.4. Tipos de datos

2. Programación básica

- 2.1. Problemas y operaciones lógico-aritméticas (Razonamiento lógico)
- 2.2. Técnicas para analizar y diseñar algoritmos
- 2.3. Estructuras de control
- 2.4. Programas interactivos
- 2.5. Listas y matrices
- 2.6. Funciones

3. Programación en Python

- 3.1. Interfaz gráfica de usuario (IDLE)
- 3.2. Uso de editores en Anaconda
- 3.3. Métodos de búsqueda y ordenamiento (matrices y vectores)
- 3.4. Programación modular
- 3.5. Diseño y presentación de resultados mediante gráficos
- 3.6. Aplicaciones

6 Recursos de Aprendizaje

Bibliografía:

- B1. Downey, A. (2012). Think Python. " O'Reilly Media, Inc.". Versión en español gratuita y digital:
<http://sourceforge.net/projects/httlcseifspa>
- B2. Matthes, E. (2015). Python crash course: a hands-on, project-based introduction to programming. No Starch Press.
- B3. Phillips, D. (2010). Python 3 Object Oriented Programming. Packt Publishing Ltd.
- B4. Al Sweigart. (2019). "Automate the Boring Stuff with Python, 2nd Edition". Free to read under a Creative Commons license.
- B5. Martin, R. (2012). "Código Limpio", Manual de estilo para el desarrollo ágil de Software.

Recursos materiales e infraestructura:

- Laboratorio de computación.
- Acceso a Ucampus.
- Computadores debidamente equipados para utilizar lenguajes de alto nivel (por ej.: Python).

7 Comportamiento y ética académica:

Se espera que los estudiantes actúen en sus diversas actividades académicas y estudiantiles en concordancia con los principios de comportamiento ético y honestidad académica propios de todo espacio universitario y que están estipulados en el *Reglamento de Estudiantes de la Universidad de Aysén*, especialmente aquéllos dispuestos en los artículos 23°, 24° y 26°.

Todo acto contrario a la honestidad académica realizado durante el desarrollo, presentación o entrega de una actividad académica del curso sujeta a evaluación, será sancionado con la suspensión inmediata de la actividad y con la aplicación de la nota mínima (1.0).

Planificación del curso

8 Responsables

Académico (s) Responsable (s) y equipo docente	Profesor: Roberto Rivera Mardones		
Contacto	Correo: roberto.rivera@uaysen.cl		
Año	2021	Periodo Académico	Primer semestre
Horario clases	Cátedra: (Sesión sincrónica) Lu 10:15 – 13:30 hrs. Mi 10:15 – 13:30 hrs.	Horario de atención estudiantes	Todos los Viernes de 12:00 a 13:00 Hrs.
Sala / Campus	Sala Virtual Ucampus – Google Sites		

9 Metodología de Trabajo:

En el curso se contemplan cuatro tipos de actividades docentes, las cuales se asocian a requerimientos de sala y al nivel de intervención del profesor/ayudante:

Actividad docente	Descripción	Intervención del profesor/ayudante	Requerimiento de sala
Exposición conceptual	El profesor introduce conceptos de programación preliminares y necesarios a otras actividades de índole práctica, de forma expositiva.	Alta	Sala de clases UCampus Online UCampus Offline
Programación expositiva	El profesor profundiza en la comprensión de elementos conceptuales a través de la exposición directa de la resolución de problemas de programación como ejemplos.	Alta	Sala de clases UCampus Online UCampus Offline
Programación tutorial	Funciona como la programación expositiva, pero el profesor realiza pausas para que los alumnos completen “pasos requeridos” antes de continuar. El objetivo es que todos los alumnos completen un paso definido por el profesor antes de continuar al siguiente.	Media	Laboratorio de computación Computador persona
Actividad práctica / Programación autónoma	Los estudiantes abordan y resuelven problemas de programación de forma autónoma, idealmente en grupos y sólo guiados por el profesor.	Baja	Laboratorio de computación Computador persona

En cualquier semana del semestre en curso se podría realizar una **evaluación menor** sobre las temáticas estudiadas a la fecha. Esta evaluación menor puede ser de los siguientes tipos:

- **Control:** evaluación individual o grupal, que se realiza en papel. Ocupará la primera parte de la clase (45 mins.).
- **Trabajo en clases:** Evaluación individual o grupal, que se realiza en computador. Ocupará la primera parte de la clase (45. mins.) o toda la clase.
- **Entrega tarea:** Evaluación grupal que se realiza en computador y fuera del horario de clases.

Al final del semestre, se eliminarán las dos peores evaluaciones menores.

Como parte de la nota de presentación del curso, los estudiantes en parejas deberán hacer una **exposición de un problema de programación**, considerando su análisis, diseño e implementación de solución. Se entregará un instructivo específico para esta evaluación.

Finalmente, se contempla la realización de **pruebas de cátedras**, que son individuales, escritas y que ocupan toda una clase. En cada una se evaluará distintos contenidos asociados a las unidades de aprendizaje de la asignatura.

10 Evaluaciones:

Evaluación	Ponderaciones específicas	Ponderación nota presentación
Pruebas de cátedra	<ul style="list-style-type: none"> • Prueba 1 (P1): 50% • Prueba 2 (P2): 50% 	50%
Evaluaciones menores	El promedio simple entre las notas consideradas corresponde al 100%.	30%
Exposición problema programación	Nota única (acorde a pauta específica)	20%

Calificación final:

- Nota de presentación: 70%
- Examen Final: 30 %

Condiciones de eximición:

- Nota de presentación igual o superior a nota 5,0
- Ponderación específica pruebas de cátedra $\geq 4,0$

Derecho a rendir examen:

Nota de presentación $\geq 3,5$

Requisito de Aprobación

- Asistencia: 70%
- Nota Final: 4,0

11 Otros aspectos asociados al funcionamiento del curso:

- Toda la coordinación del curso (comunicaciones, actualización de notas, material, etc.) será realizada a través de UCampus. El estudiante deberá informar con tiempo suficiente si presenta dificultades de conexión para trasladar el requerimiento a la coordinación de programa. Adicionalmente los estudiantes deberán acceder al material disponible en el google sites que el profesor preparó para este propósito.
- **Es deber del estudiante mantenerse informado de las noticias, avisos y material entregado por el profesor a través de estos medios, se sugiere instalar en su dispositivo móvil la aplicación de UCampus.**

12 Planificación de las actividades de enseñanza- aprendizaje y de evaluación

Semana	Resultado(s) de Aprendizaje (*)	Tema (Unidades de aprendizaje) y actividades	Recursos utilizados o lecturas (*)	Actividad(es)
1	RA1	1. Introducción a la programación 1.1. Motivación 1.2. Conceptos de algoritmos	B1, B2	
2	RA1	1. Introducción a la programación 1.3. Lenguajes de programación 1.4. Tipos de datos	B1, B2	
3	RA1	2. Programación básica 2.1. Problemas y operaciones lógico-aritméticas (Razonamiento lógico)	B1, B2	
4	RA1, RA3	2. Programación básica 2.2. Técnicas para analizar y diseñar algoritmos Ejercitación	B1, B2	
5	RA1, RA3, RA5	2. Programación básica 2.4. Estructuras de control (Secuenciales y condicionales)	B1, B2	Cátedra 1 (P1)
6	RA1, RA2, RA3 RA5	2. Programación básica 2.5. Programas interactivos (ciclos de repetición tipo “for” y “while”)	B1, B2	
7		Ejercitación	B1, B2	
8	RA1, RA2, RA3 RA5	2. Programación básica 2.6. Listas y matrices (arreglos unidimensionales y bidimensionales)	B1, B2	
9	RA1, RA2, RA3 RA5	2. Programación básica 2.7. Funciones y procedimientos	B1, B2	
10	RA3, RA4, RA5	3. Programación en Python 3.1 Interfaz gráfica de usuario (IDLE) 3.2. Uso de editores en Anaconda	B1, B2, B3	Cátedra 2 (P2)
11	RA3, RA4, RA5	3. Programación en Python 3.3 Métodos de búsqueda y ordenamiento (matrices y vectores).	B1, B2, B3	
12		Ejercitación		
13	RA3, RA4, RA5	3. Programación en Python 3.4. Programación modular	B1, B2, B3	
14	RA3, RA4, RA5	3. Programación en Python 3.5. Diseño y presentación de resultados mediante gráficos.	B1, B2, B3	Exposición de un problema de programación
15	RA3, RA4, RA5	3. Programación en Python 3.6. Aplicaciones	B1, B2, B3	Exposición de un problema de programación
16	RA3, RA4, RA5	Ejercitación	B1, B2, B3	
17				Cátedra 3 (P3) Recuperativa
18		Periodo de exámenes		Examen



(*) Para referencias de Resultados de Aprendizaje y Recursos, consultar el programa del curso.