

Programa de Asignatura

1. Identificación Asignatura

| | | | |
|-------------------------|------------------------------|-------------------------------------|---------------------------------|
| Nombre: | Fundamentos de Programación | Código: | IF1001 |
| Carrera: | Ingeniería Civil Informática | 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: | 4.5 |
| Pre-requisitos | Herramientas informáticas | | |

2. Propósito formativo

La asignatura de Fundamentos de Programación se ubica en el ciclo formativo inicial y corresponde a la línea formativa de **Fundamentos de la informática y las Cs. Básicas**. 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.

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 Avanzada, Algoritmos y Estructuras de datos, entre otras, ya que introduce elementos de la informática y computación que dichas asignaturas requieren.

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:

1. Entiende problemas a través de la construcción de abstracciones conceptuales, cualitativas y cuantitativas, utilizando formalismos establecidos, que permitan formular soluciones
2. Diseña y programa soluciones, utilizando estrategias algorítmicas, que permitan resolver problemas de forma eficaz y acorde a múltiples objetivos de diseño.

4. Resultados de aprendizaje específicos

| Resultado de Aprendizaje Específico | Criterios de evaluación | Evidencia |
|--|---|-----------------------------------|
| 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. | 1.1. El estudiante resuelve problemas utilizando lógica algorítmica. 1.2. El estudiante esta en la capacidad de identificar diferentes tipos de flujos en las estructuras algorítmicas. | Laboratorios, guías de ejercicio. |
| RA2. Diseña algoritmos de pequeña a mediana envergadura que le permitan resolver sistemáticamente problemas básicos de ciencias e ingeniería. | 1.3. El estudiante logra elaborar de manera autónoma instrucciones tipo algoritmo. 1.4. El estudiante reconoce los diferentes tipos de datos, palabras reservadas y sentencias en un lenguaje de alto nivel. | Laboratorios, guías de ejercicio. |
| RA3. Implementa y valida estructuras de datos de complejidad básica a mediana para su aplicación en programas de distinta naturaleza. | 1.5. El estudiante tiene las competencias de desarrollar de forma autónoma un proyecto de programación. | Laboratorios, guías de ejercicio. |

5. Unidades de Aprendizaje

| |
|---|
| <p>1. Introducción a la programación</p> <p>1.1. Motivación</p> <p>1.2. Conceptos de algoritmos</p> <p>1.3. Lenguajes de programación</p> <p>1.4. Técnicas para analizar y diseñar algoritmos</p> <p>1.5. Tipos de datos y expresiones</p> |
| <p>2. Programación básica en python</p> <p>2.1. Uso de editores en Anaconda (instalar previamente Anaconda o Python3 en usuarios Linux)</p> <p>2.2. Problemas y operaciones lógico-aritméticas (Razonamiento lógico)</p> <p>2.3. Estructuras de control</p> <p>2.4. Ciclos</p> <p>2.5. Funciones de alto nivel</p> |
| <p>3. Estructuras Avanzadas</p> <p>3.1. Funciones y recursividad</p> <p>3.2. Tipos de datos compuestos: Diccionarios, listas, vectores y matrices</p> <p>3.3. Métodos de búsqueda y ordenamiento (matrices y vectores)</p> <p>3.4. Archivos</p> |

6. Recursos de Aprendizaje

| |
|--|
| <p>Bibliografía:</p> <p>B1. Urquía Moraleda, A. Martín Villalba, C. y Rubio González, M. Á. (2021). Lenguajes de programación. UNED -</p> |
|--|

Universidad Nacional de Educación a Distancia. <https://elibro.net/es/lc/uaysen/titulos/184827>

- B2. Downey, A. (2012). Think Python. " O'Reilly Media, Inc.". Versión en español gratuita y digital: <http://sourceforge.net/projects/httlcseifspa>
- B3. Algar Díaz, M. J. y Fernández de Sevilla Vellón, M. (2019). Introducción práctica a la programación con Python. Editorial Universidad de Alcalá. <https://elibro.net/es/ereader/uaysen/124259?page=34>

Recursos materiales e infraestructura:

- Laboratorio de computación.
- Acceso a Ucampus.
- Acceso a Googlesites con credenciales institucionales.

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: Gabriel Núñez Vivanco, Luz Cardona | | |
| Contacto | Correo: gabriel.nunez@uaysen.cl , luz.cardona@uaysen.cl | | |
| Año | 2021 | Periodo Académico | Segundo semestre |
| Horario clases | Cátedra: Ma 10:15 – 11.45 hrs. Laboratorio: Ju 8:30 – 11.45 hrs | Horario de atención estudiantes | Ma 8.30 – 10:15 (Online previa confirmación) |
| Sala / Campus | Sala Virtual Ucampus | | |

9. Metodología de Trabajo:

| | | | |
|---|-----------|---|-----------|
| La asignatura contiene: | | | |
| Actividades de vinculación con el medio | No | Actividades relacionadas con proyectos de investigación | No |
| 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: | | | |

| 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. Se dispone de materiales complementarios en la plataforma Ucampus. | 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, algunas con guía y apoyo docente y otras no. | Baja/Media | 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:

- **Laboratorio:** Evaluación individual o grupal, que se realiza en el computador. Ocupará los bloques del día jueves.
- **Guía de ejercicios:** Evaluación individual que se realiza en computador durante el tiempo de trabajo autónomo.
- **Prueba Parcial:** Evaluación individual que se realiza en computador en el horario de clases.

10. Evaluaciones:

| Evaluación | Ponderaciones específicas | Ponderación nota presentación |
|----------------------|---|-------------------------------|
| Pruebas de cátedra | <ul style="list-style-type: none"> • Prueba Unidad 1 (P1): 30% • Prueba Unidad 2 (P2): 35% • Prueba Unidad 3 (P2): 35% | 60% |
| Laboratorios y Guías | El promedio simple entre las notas consideradas corresponde al 100%. | 40% |

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 / Sesión | Resultado(s) de Aprendizaje | Tema (Unidades de aprendizaje) y actividades | Recursos utilizados o lecturas | Actividad(es) de Trabajo Autónomo |
|---|-----------------------------|--|--------------------------------|-----------------------------------|
| <u>Semana 1</u> 31/08 Online 02/08 Online | ORA1 | Presentación programa de estudio. Introducción a la programación a) Motivación b) Conceptos de algoritmos | B1 | Guía de ejercicios. |
| <u>Semana 2</u> 7/09 Clase Presencial + transmisión online. Sala1 o 2 Campus Obispo Vielmo 9/09 Laboratorio Presencial + apoyo en por videoconferencia en paralelo Sala de Computo Obispo Vielmo | ORA1 | Introducción a la programación c) Lenguajes de programación | B1 | Guía de ejercicios y laboratorio. |
| <u>Semana 3</u> 14/09 Online 16/09 Online | ORA1 | Introducción a la programación c) Técnicas para analizar y diseñar algoritmos d) Tipos de datos y | B1 | Guía de ejercicios y laboratorio. |

| | | | | |
|--|----------|--|--------|--|
| | | expresiones | | |
| Semana 4 21/09 Online 23/09 Online | ORA1 | Introducción a la programación d) Tipos de datos y expresiones | B1 | Guía de ejercicios y laboratorio. |
| Semana 5 | | | | Prueba 1 (P1) |
| Semana 6 5/10 Clase Presencial + transmisión online. Sala1 o 2 Campus Obispo Vielmo 7/10 Laboratorio Presencial + apoyo en por videoconferencia en paralelo Sala de Computo Obispo Vielmo | RA2, RA3 | Programación básica en python a) Uso de editores en Anaconda (instalar previamente Anaconda o Python3 en usuarios Linux) | B2, B3 | Instalación IDLE y prueba de editores. |
| Semana 7 12/10 Online 14/10 Online | RA2, RA3 | Programación básica en python b) Problemas y operaciones lógico-aritméticas (Razonamiento lógico) c) Estructuras de control | B2, B3 | Guía de ejercicios y laboratorio. |
| Semana 8 19/10 Online 21/10 Online | RA2, RA3 | Programación básica en python d) Ciclos y Funciones | B2, B3 | Guía de ejercicios y laboratorio. |
| Semana 9 Online | | | | Prueba 2 |
| Semana 10 2/11 Clase Presencial + transmisión online. Sala1 o 2 Campus Obispo Vielmo 4/11 Laboratorio Presencial + apoyo en por videoconferencia en paralelo Sala de Computo Obispo Vielmo | RA2, RA3 | Estructuras Avanzadas a) Funciones y recursividad | B2, B3 | Guía de ejercicios y laboratorio. |

| | | | | |
|---|----------|---|--------|-----------------------------------|
| <u>Semana 11</u> 9/11 Online 1/11 Online | RA2, RA3 | Estructuras Avanzadas b) Tipos de datos compuestos: Diccionarios, listas, vectores y matrices | B2, B3 | Guía de ejercicios y laboratorio. |
| <u>Semana 11</u> 16/11 Online 18/11 Online | RA2, RA3 | Estructuras Avanzadas c) Métodos de búsqueda y ordenamiento (matrices y vectores) | B2, B3 | Guía de ejercicios y laboratorio. |
| <u>Semana 11</u> 23/11 Online 25/11 Online | RA2, RA3 | Estructuras Avanzadas d) Archivos | B2, B3 | Guía de ejercicios y laboratorio. |
| <u>Semana 14</u> 30/11 Clase Presencial + transmisión online. Sala1 o 2 Campus Obispo Vielmo 2/12 Laboratorio Presencial + apoyo en por videoconferencia en paralelo Sala de Computo Obispo Vielmo | RA2, RA3 | Estructuras Avanzadas d) Archivos | B2, B3 | Guía de ejercicios y laboratorio. |
| <u>Semana 15</u> Online | | | | Prueba 3 (P3) |
| <u>Semana 16</u> | | Semana de exámenes | | Examen final |
| <u>Semana 17</u> | | Semana de exámenes | | Examen final |
| | | | | |