

SÍLABO

Lenguaje de Programación 1

| | | | | |
|----------------------|-----------------------------|-----------------|------------------|---|
| Código | 24UC00082 | Carácter | Obligatorio | |
| Requisito | Fundamentos de Programación | | | |
| Créditos | 3 | | | |
| Horas | Teóricas | 2 | Prácticas | 2 |
| Año académico | 2024 | | | |

I. Introducción

Lenguaje de programación I es una asignatura de especialidad, de carácter obligatorio para la Escuela Académico Profesional de Ciencia de la Computación, que se cursa en el segundo ciclo. Esta asignatura contribuye a desarrollar las competencias Diseño y Desarrollo de Soluciones y Conocimientos de Computación, en el nivel 1. Tiene como requisito la asignatura de Fundamentos de Programación. Por su naturaleza, incluye componentes teóricos y prácticos que permiten conocer los conceptos de análisis léxico, sintáctico y semántico para diseñar soluciones a problemas complejos. Por otro lado, debido a la naturaleza de los contenidos que desarrolla, la asignatura puede tener un formato presencial, virtual o blended.

Los contenidos generales que la asignatura desarrolla son los siguientes: visión de los lenguajes de programación, conceptos fundamentales de programación, programación orientada a objetos.

II. Resultado de aprendizaje de la asignatura

Al finalizar la asignatura, el estudiante será capaz de analizar las técnicas de lenguajes de programación para la implementación de programas con soluciones a problemas planteados.

III. Organización de los aprendizajes

| Unidad 1 Visión de los lenguajes de programación | | Duración en horas | 16 |
|-------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|----|
| Resultado de aprendizaje de la unidad | Al finalizar la unidad, cada estudiante será capaz de comparar los principales paradigmas de programación, seleccionando el más adecuado para la solución de un problema dado. | | |
| Ejes temáticos | <ol style="list-style-type: none"> 1. Breve revisión de los paradigmas de programación 2. Comparación entre programación funcional y programación imperativa 3. Historia de los lenguajes de programación 4. Sintaxis y semántica básica de un lenguaje de alto nivel 5. Variables y tipos de datos primitivos 6. Palabras reservadas del lenguaje | | |

| Unidad 2 Conceptos Fundamentales de Programación | | Duración en horas | 16 |
|-------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|----|
| Resultado de aprendizaje de la unidad | Al finalizar la unidad, cada estudiante será capaz de aplicar los conceptos básicos de programación para el desarrollo de algoritmos que resuelvan problemas. | | |
| Ejes temáticos | <ol style="list-style-type: none"> 1. Expresiones y asignaciones 2. Operaciones básicas 3. Librerías estándar del lenguaje 4. Estructuras de control condicional e iterativas 5. Paso de funciones y parámetros | | |

| Unidad 3 Programación Orientada a Objetos | | Duración en horas | 16 |
|------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|----|
| Resultado de aprendizaje de la unidad | Al finalizar la unidad, cada estudiante será capaz de aplicar los conceptos de la programación orientada a objetos apropiados para la solución de problemas específicos. | | |
| Ejes temáticos | <ol style="list-style-type: none"> 1. Análisis y diseño orientado a objetos 2. Lenguajes orientados a objetos para la encapsulación 3. Definición de las categorías, campos, métodos y constructores 4. Herencia | | |

| Unidad 4 Clases Abstractas y Polimorfismo | | Duración en horas | 16 |
|------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|----|
| Resultado de aprendizaje de la unidad | Al finalizar la unidad, cada estudiante será capaz de desarrollar clases abstractas para representar conceptos generales implementando comportamientos específicos, así como aplicar el polimorfismo para manipular objetos de diferentes tipos mediante una referencia común. | | |
| Ejes temáticos: | <ol style="list-style-type: none"> 1. Clases abstractas, Polimorfismo, vinculación dinámica 2. Interfaces 3. Gestión de excepciones 4. Enumerados, Arrays, Clases Internas 5. Anotaciones | | |

IV. Metodología

Modalidad A Distancia (formato virtual)

Los contenidos y actividades propuestas en las cuatro unidades de la asignatura se desarrollarán siguiendo la secuencia teórico-práctica; se usará la metodología activa desde clases magistrales activas, el trabajo colaborativo, promoviendo el debate, las exposiciones, el uso de organizadores de información con incidencia en el desarrollo de casos de investigación. El estudiante utilizará el material de trabajo para resolver casos prácticos, realizará la investigación bibliográfica e investigación vía internet.

Durante las sesiones, se guiará a los estudiantes a través de:

- aprendizaje colaborativo,
- estudio de casos,
- aprendizaje basado en problemas,
- *flipped classroom*.

V. Evaluación

Sobre la probidad académica

Las faltas contra la probidad académica se consideran infracciones muy graves en la Universidad Continental. Por ello, todo docente está en la obligación de reportar cualquier incidente a la autoridad correspondiente; sin perjuicio de ello, para la calificación de cualquier trabajo o evaluación, en caso de plagio o falta contra la probidad académica, la calificación será siempre cero (00). En función de ello, todo estudiante está en la obligación de cumplir el [Reglamento Académico](#)¹ y conducirse con probidad académica en todas las asignaturas y actividades académicas a lo largo de su formación; de no hacerlo, deberá someterse a los procedimientos disciplinarios establecidos en el mencionado documento.

¹ Descarga el documento en el siguiente enlace <https://shorturl.at/fhosu>

Modalidad A Distancia (formato virtual)

| Rubros | Unidad por evaluar | Semana | Entregable | Instrumento | Peso parcial (%) | Peso total (%) |
|---------------------------------|--------------------------------------------------------------------|----------------|-----------------------------|-----------------------|------------------|----------------|
| Evaluación de entrada | Requisito | Primera sesión | Cuestionario | Prueba objetiva | 0 | |
| Consolidado 1 C1 | Unidad 1 | 1 - 2 | Actividades virtuales | | 15 | 20 |
| | | 2 | Archivos de programación | Rúbrica de evaluación | 85 | |
| Evaluación parcial EP | Unidad 1 y 2 | 3 - 4 | Actividades virtuales | | 15 | 25 |
| | | 4 | Programa desarrollado | Rúbrica de evaluación | 85 | |
| Consolidado 2 C2 | Unidad 3 | 5 - 6 | Actividades virtuales | | 15 | 20 |
| | | 6 | Archivos de programación | Rúbrica de evaluación | 85 | |
| Evaluación final EF | Todas las unidades | 7 - 8 | Actividades virtuales | | 15 | 35 |
| | | 8 | Programa desarrollado | Rúbrica de evaluación | 85 | |
| Evaluación sustitutoria* | Todas las unidades Fecha posterior a la evaluación final | | Evaluación teórico-práctica | Prueba mixta | | |

* Reemplaza la nota más baja obtenida en los rubros anteriores.

Fórmula para obtener el promedio

$$PF = C1 (20 \%) + EP (25 \%) + C2 (20 \%) + EF (35 \%)$$

VI. Atención a la diversidad

En la Universidad Continental generamos espacios de aprendizaje seguros para todas y todos nuestros estudiantes, en los cuales puedan desarrollar su potencial al máximo. En función de ello, si un(a) estudiante tiene alguna necesidad, debe comunicarlo al o la docente. Si el estudiante es una persona con discapacidad y requiere de algún ajuste razonable en la forma en que se imparten las clases o en las evaluaciones, puede comunicar ello a la Unidad de Inclusión de Estudiantes con Discapacidad. Por otro lado, si el nombre legal del estudiante no corresponde con su identidad de género, puede comunicarse directamente con el o la docente de la asignatura para que utilice su nombre social. En caso hubiera algún inconveniente en el cumplimiento de estos lineamientos, se puede acudir al(la) director(a) o al(la) coordinador(a) de carrera o a la Defensoría Universitaria, lo que está sujeto a la normativa interna de la Universidad.

VII. Bibliografía

Básica

- Charatan, Q. y Kans, A. (2019). *Java in Two Semesters Featuring: Featuring JavaFX* (4.ª ed.). Springer.
- Deitel, P. y Deitel, H. (2016). *Cómo programar en Java* (10.ª ed.). Pearson educación.
- Kleinberg, J. y Tados, E. (2013). *Algorithms design*. Pearson.
- Sedgewick, R. y Wayne, K. (2016). *Algorithms* (4.ª ed.). Addison-Wesley Professional.

Complementaria

- Acera, M. (2012). *Manual imprescindible de C/C ++*. Anaya Multimedia.
- Joyanes, L. (2020). *Fundamentos de programación: Algoritmos, estructura de datos y objetos* (5.ª ed.). McGraw Hill. <https://cutt.ly/twBk3D2t>
- López, J. (2009). *Algoritmos y programación (Guía para docentes)* (2.ª ed.). Editorial de la Fundación Gabriel Piedrahia Uribe. <https://cutt.ly/lwBk6P1x>
- Melguizo, P. (2011). *Laboratorio de programación: índice de prácticas, ejercicios resueltos*. Universidad de Alcalá. <https://cutt.ly/bwBleFgG>
- Prialé, L. (2009). *Algoritmos y diagramas de flujo aplicaciones en C++*. Megabyte.

VIII. Recursos digitales

- Benjumea, V. y Roldán, M. (2022). *Fundamentos de Programación en Lenguaje C*. Universidad de Málaga. <https://cutt.ly/pwBxsAXF>
- Barber, F. y Ferris, R. (s.f.) *Subprogramas, programación modular*. Universidad de Valencia. <https://cutt.ly/zwBxTajZ>
- LucidChart. (2024). *Creador de diagramas de flujo*. <https://www.lucidchart.com/pages/es/ejemplos/diagrama-de-flujo-online>
- Sanchez, J. (s.f.). *Ejercicios sencillos de lenguaje C*. JorgeSanchez.net. <https://cutt.ly/LwBxTkQr>