

SÍLABO

Lenguaje de Programación 2

Código	24UC00686	Carácter	Obligatorio	
Requisito	Lenguaje de Programación 1			
Créditos	3			
Horas	Teóricas	2	Prácticas	2
Año académico	2025			

I. Introducción

Lenguaje de Programación 2 es una asignatura de especialidad y de carácter obligatorio para la Escuela Académico Profesional de Ciencia de la Computación, que se cursa en el tercer ciclo de estudios. Esta asignatura contribuye a desarrollar las competencias Diseño y desarrollo de soluciones y Conocimientos de computación, en el nivel 2. Tiene como requisito Lenguaje de Programación 1. Por su naturaleza, incluye componentes teóricos y prácticos que permiten evaluar los fundamentos matemáticos y principios algorítmicos para diseñar requerimientos y restricciones de forma clara en el modelamiento de problemas computacionales. Por otro lado, debido a la naturaleza de los contenidos que desarrolla, la asignatura puede tener un formato virtual en la modalidad A Distancia.

Los contenidos generales que la asignatura desarrolla son los siguientes: análisis y diseño orientado a objetos, aplicaciones y API para POO, programación reactiva y dirigida por eventos, modelo MVC.

II. Resultado de aprendizaje de la asignatura

Al finalizar la asignatura, el estudiante será capaz de evaluar estructuras discretas elementales para el planteamiento y solución de problemas complejos.

III. Organización de los aprendizajes

Unidad 1 Análisis y diseño orientado a objetos		Duración en horas	16
Resultado de aprendizaje de la unidad	Al finalizar la unidad, cada estudiante será capaz de aplicar los principios de análisis y diseño orientado a objetos para el desarrollo de modelos de <i>software</i> eficientes y escalables.		
Ejes temáticos	<ol style="list-style-type: none"> 1. Principios de POO 2. Modelado de clases y relaciones 3. Diagramas UML (Unified Modeling Language) 		

Unidad 2 Desarrollo de aplicaciones e interfaces de programación de aplicaciones para POO		Duración en horas	16
Resultado de aprendizaje de la unidad	Al finalizar la unidad, cada estudiante será capaz de desarrollar aplicaciones y las API utilizando POO, integrando técnicas de modularidad y reutilización de código para la optimización del desarrollo del <i>software</i> .		
Ejes temáticos	<ol style="list-style-type: none"> 1. Implementación de clases y objetos 2. Interfaz de programación de aplicaciones (API) 3. Gestión de eventos y excepciones 		

Unidad 3 Programación reactiva y dirigida por eventos		Duración en horas	16
Resultado de aprendizaje de la unidad	Al finalizar la unidad, cada estudiante será capaz de implementar conceptos de programación reactiva y dirigida por eventos para la creación de sistemas responsivos y eficientes en tiempo real.		
Ejes temáticos	<ol style="list-style-type: none"> 1. Conceptos de programación reactiva 2. Manejo de eventos y <i>callbacks</i> 3. Patrones de diseño reactivos 		

Unidad 4 Modelo MVC (modelo-vista-controlador)		Duración en horas	16
Resultado de aprendizaje de la unidad	Al finalizar la unidad, cada estudiante será capaz de evaluar un modelo MVC considerando indicadores de eficiencia y escalabilidad en la solución de problemas complejos.		
Ejes temáticos	<ol style="list-style-type: none"> 1. Introducción al modelo MVC 2. Implementación del patrón MVC en aplicaciones 3. Separación de responsabilidades 		

IV. Metodología

Modalidad A Distancia - formato virtual

Se hará uso de la metodología activa desde clases magistrales activas y el trabajo colaborativo, promoviendo el debate además del uso de organizadores de conocimiento, incidiendo en la aplicación práctica de la teoría obtenida en cada unidad. El estudiante hará uso del material de trabajo para la realización de los casos prácticos, que deberá complementar con la investigación bibliográfica e investigación vía internet.

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

- Clase expositiva/lección magistral (CE-LM)
- Aprendizaje colaborativo
- Estudio de casos
- Aprendizaje basado en problemas

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 reglamento.

¹ Descargar el documento: <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	Evaluación individual teórica	Prueba objetiva	0	
Consolidado 1 C1	Unidad 1	1 - 3	Actividades virtuales		15	20
			Trabajo práctico individual: archivos de programación	Rúbrica de evaluación	85	
Evaluación parcial EP	Unidad 1 y 2	4	Trabajo práctico individual: programa desarrollado	Rúbrica de evaluación	25	
Consolidado 2 C2	Unidad 3	5 - 7	Actividades virtuales		15	20
			Trabajo práctico grupal: archivos de programación	Rúbrica de evaluación	85	
Evaluación final EF	Todas las unidades	8	Trabajo práctico individual: programa desarrollado	Rúbrica de evaluación	35	
Evaluación sustitutoria*	Todas las unidades Fecha posterior a la evaluación final		Trabajo práctico individual: programa desarrollado	Rúbrica de evaluación		

* 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 comunicarla a el 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 a su director(a) o coordinador(a) de carrera o a la Defensoría Universitaria, lo que está sujeto a la normativa interna de la Universidad.

VII. Bibliografía

Básica

- Blasco, F. (2019). *Programación orientada a objetos en Java*. RaMa. <https://d82m.short.gy/tqHObK>
- Charatan, Q. y Kans, A. (2019). *Java in two semesters: featuring JavaFx*. (4.ª ed.). Springer. <https://d82m.short.gy/uJMGQg>
- Cormen, T., Leiserson, C., Rivest, R. y Stein, C. (2022). *Introduction to Algorithms* (4.ª ed.). The MIT Press. <https://bit.ly/3RWasJA>
- Deitel, P. y Deitel, H. (2016). *Cómo programar en Java*. (10.ª ed.). Pearson educación. <https://bit.ly/4cQYUQ7>
- Sedgewick, R. y Wayne, K. (2011). *Algorithms*. (4.ª ed.). Pearson Educación. <https://d82m.short.gy/mEws4X>

Complementaria

- Charatan, Q. y Kans, A. (2019). *Java in two semesters featuring*. (4th ed.). Springer.
- Cormen, T., Leiserson, C., Rivest, R., and Stein, C. (2009). *Introduction to Algorithms*. (4th ed.). The MIT Press. McGraw-Hill.
- Deitel, P. y Deitel, H. (2016). *Cómo programar en Java*. (10.ª ed.). Pearson Educación.
- Sedgewick, R., and Wayne, K. (2016). *Algorithms*. (4th ed.). Addison-Wesley Professional. ISBN-13: 978-0321573513

VIII. Recursos digitales

- Aguilar, J. (15 de octubre de 2019). ¿Qué es el patrón MVC en programación y por qué es útil?. Campus MVP. <https://bit.ly/4eQkhT9>
- GitHub. (20 de julio de 2022). *Curso completo: fundamentos do C#*. GitHub. <https://github.com/topics/poo?l=c%23>
- GitHub. (s.f.). *Programación reactiva y dirigida por eventos con RxJava*. GitHub. <https://github.com/sergiorrojas/rxJava>
- Subins 2000. (s.f.). *Draw.io herramienta de diagramación (Versión 24.6.4)* [Software de computadora]. <https://app.diagrams.net>