

# SÍLABO



## FACULTAD DE INGENIERÍA

MARZO 2024-AGOSTO 2024

NOMBRE DE LA ASIGNATURA		CÓDIGO:	18566
PROGRAMACIÓN III (ESTRUCTURA DE ARCHIVOS) - GRUPO: 1			
CARRERA	COMPUTACION REDISEÑO		
CICLO O SEMESTRE	CUARTO NIVEL	EJE DE FORMACIÓN	PROFESIONALES, PRAXIS PROFESIONAL
CRÉDITOS DE LA ASIGNATURA	4	MODALIDAD:	PRESENCIAL

### CARGA HORARIA

COMPONENTES DEL APRENDIZAJE	Horas / Semana	Horas / Periodo Académico
APRENDIZAJE EN CONTACTO CON EL DOCENTE (ACD)	4.0	64.0
APRENDIZAJE PRÁCTICO EXPERIMENTAL - ASIGNATURA (APE/A)	2.0	32.0
APRENDIZAJE AUTÓNOMO (AA)	6.0	96.0
<b>Total Horas:</b>	12.0	192.0

### PROFESOR(ES) RESPONSABLE(S):

GRANDA JUCA MARIA FERNANDA - (M.G.)	( fernanda.granda@ucuenca.edu.ec )	PRINCIPAL
-------------------------------------	------------------------------------	-----------

### DESCRIPCIÓN DE LA ASIGNATURA:

Resumen descriptivo en torno al propósito, la estrategia metodológica y el contenido fundamental de la asignatura.

En este curso se repasarán las principales estructuras de datos que se utilizan en problemas prácticos de programación, se profundizará en el conocimiento de técnicas de programación como base para desarrollar aplicaciones que utilizan almacenamiento secundario y se estudian diferentes técnicas de compresión y encriptación de archivos.

El curso se divide en tres partes. En la primera parte se revisa las diferentes estructuras de datos descritas en cursos de programación anteriores. En el transcurso de la revisión del lenguaje de programación se hace hincapié en la capacidad de abstracción que debe tener un ingeniero en computación como elemento fundamental para utilizar estructuras de datos sin ambigüedades. Se realiza una introducción a la teoría de estructura de archivos y la relación con las estructura de datos. Se cubren aspectos relacionados con la utilización de un lenguaje de programación orientado a objetos para la implementación de los diferentes algoritmos de estructura de archivos. En la segunda parte del curso se estudia conceptos de programación por capas para el ordenamiento del proceso de programación y reducción del impacto de los cambios de los requerimientos y casos de uso sobre el código fuente de una aplicación. Además, se estudia la comprobación del correcto funcionamiento del código/sistema con pruebas unitarias y de integración. En la tercera parte del curso se estudia conceptos generales sobre compresión y encriptación de archivos.

### REQUISITOS DE LA ASIGNATURA

Esta asignatura no tiene co-requisitos

PRE-REQUISITOS	
Asignatura	Código
PROGRAMACIÓN II: ESTRUCTURA DE DATOS Y ANÁLISIS DE ALGORITMOS	17952

### OBJETIVO(S) DE LA ASIGNATURA:

Objetivos general y específicos de la asignatura en relación al Perfil de salida de la carrera.

**Objetivo general:** Capacitar al estudiante en el conocimiento de técnicas de programación orientadas a objetos como base para desarrollar aplicaciones que utilizan almacenamiento secundario, técnicas de compresión y encriptación de archivos.

**Objetivos específicos:**

1. Introducir un lenguaje de programación orientado a objetos para el desarrollo de aplicaciones informáticas.
2. Comprender, evaluar, utilizar y dominar estructuras dinámicas de datos y su aplicación para el diseño de sistemas informáticos.
3. Analizar y diseñar arquitecturas de programación por capas para el desarrollo de sistemas informáticos.
4. Conocer y aplicar los conceptos de pruebas unitarias y de integración.
5. Identificar los mejores algoritmos y estructuras para gestionar la persistencia de los datos.
6. Evaluar y utilizar diferentes algoritmos para la encriptación y compresión de archivos.

**LOGRO DE LOS RESULTADOS DE APRENDIZAJE, INDICADOR(ES) Y ESTRATEGIA(S) DE EVALUACIÓN**

*Resultados o Logros de Aprendizaje (RdA's) de la Unidad de Organización Curricular (UOC) correspondiente, Indicadores y Estrategias de Evaluación de la Asignatura, tomando como referencia el Perfil de salida (PdS) y la Organización Curricular (OC) del Proyecto de Carrera (PdC).*

RESULTADOS O LOGROS DE APRENDIZAJE	INDICADORES	ESTRATEGIAS DE EVALUACIÓN
<b>RdA1.</b> Identifica las diferentes estructuras de datos para seleccionar las más adecuadas para el desarrollo de aplicaciones	<ul style="list-style-type: none"> <li>• Justifica la utilización de un tipo de estructuras para una situación determinada.</li> <li>• Determina las ventajas y desventajas de las estructuras estáticas y dinámicas de datos.</li> </ul>	<ul style="list-style-type: none"> <li>• Comprende las ventajas y desventajas de las diferentes estructuras de datos.</li> <li>• Selecciona adecuadamente las estructuras de datos de acuerdo al problema planteado.</li> <li>• Selecciona adecuadamente las estructuras de datos de acuerdo al problema planteado.</li> </ul>
<b>RdA2.</b> Diseña una arquitectura de programación por capas.	<ul style="list-style-type: none"> <li>• Diseña y construye las capas a utilizar en el proceso de desarrollo de aplicaciones.</li> </ul>	<ul style="list-style-type: none"> <li>• Comprende los conceptos de una arquitectura de programación por capas.</li> <li>• Diseña e implementa una arquitectura de programación por capas.</li> </ul>
<b>RdA3.</b> Aplica las diferentes estructuras de archivos para el desarrollo de aplicaciones.	<ul style="list-style-type: none"> <li>• Identifica métodos para el manejo de las diferentes estructuras de datos.</li> <li>• Aplica algoritmos y métodos acorde al tipo de estructura de datos.</li> <li>• Construye aplicaciones reutilizando librerías existentes.</li> </ul>	<ul style="list-style-type: none"> <li>• Comprende las ventajas y desventajas de las diferentes estructuras de archivos.</li> <li>• Selecciona adecuadamente el tipo de estructura de archivos de acuerdo al problema a resolver.</li> <li>• Desarrolla software usando el código apropiado para la estructura de archivos a implementar</li> </ul>
<b>RdA4.</b> Aplica los diferentes algoritmos de compresión y encriptación de archivos.	<ul style="list-style-type: none"> <li>• Identifica los diferentes formatos de compresión y encriptación existentes.</li> <li>• Aplica algoritmos básicos de compresión y encriptación de archivos.</li> <li>• Investiga algoritmos de compresión y encriptación de archivos.</li> <li>• Crea algoritmos básicos de encriptación y compresión de archivos.</li> </ul>	<ul style="list-style-type: none"> <li>• Comprende los diferentes algoritmos de compresión y encriptación de archivos.</li> <li>• Utiliza diferentes algoritmos de compresión y encriptación de archivos.</li> </ul>

**CONTENIDOS, SESIONES Y ACTIVIDADES DE APRENDIZAJE**

*Título de la Unidad, sub -unidades, nro. de sesión y actividades para los componentes de aprendizaje.*

SUB-UNIDADES	Nro. SESIÓN	COMPONENTE DE APRENDIZAJE	ACTIVIDADES DE APRENDIZAJE
<b>1. PROGRAMACIÓN EN JAVA</b>			

SUB-UNIDADES	Nro. SESIÓN	COMPONENTE DE APRENDIZAJE	ACTIVIDADES DE APRENDIZAJE	
1. Presentación de Sílabo y Evaluación Diagnóstica	1	APRENDIZAJE EN CONTACTO CON EL DOCENTE (ACD)	EVALUACIÓN DIAGNÓSTICA Y EXPLICACIÓN DEL SÍLABO	0.5 horas
2. Programación Orientada a Objetos			EVALUACIÓN DIAGNÓSTICA: TEST	0.5 horas
3. Lenguaje Java			EXPLICACIÓN DEL DOCENTE	5 horas
4. Código Limpio		APRENDIZAJE PRÁCTICO EXPERIMENTAL - ASIGNATURA (APE/A)	TRABAJO EN LABORATORIO	4 horas
		APRENDIZAJE AUTÓNOMO (AA)	TAREAS EN CASA	4 horas
2. ESTRUCTURA DE DATOS				
1. Librerías de java para pilas, colas, listas enlazadas, conjuntos, hash, map.	2	APRENDIZAJE EN CONTACTO CON EL DOCENTE (ACD)	EXPLICACIÓN DEL DOCENTE	10 horas
2. Uso de clases Genéricas		APRENDIZAJE PRÁCTICO EXPERIMENTAL - ASIGNATURA (APE/A)	TRABAJO EN LABORATORIO	8 horas
3. Programación gráfica con Java		APRENDIZAJE AUTÓNOMO (AA)	TAREAS EN CASA	20 horas
4. Implementación de Herencia y Polimorfismo en Java				
3. PROGRAMACIÓN EN CAPAS				
1. Programación en Capas	3	APRENDIZAJE EN CONTACTO CON EL DOCENTE (ACD)	EXPLICACIÓN DEL DOCENTE	12 horas
2. Pruebas unitarias			PRUEBA 1	2 horas
3. Pruebas de integración		APRENDIZAJE PRÁCTICO EXPERIMENTAL - ASIGNATURA (APE/A)	TRABAJO EN LABORATORIO	6 horas
		APRENDIZAJE AUTÓNOMO (AA)	TAREAS EN CASA	19 horas
4. ESTRUCTURA DE ARCHIVOS				
1. Registros, tipos y operaciones	4	APRENDIZAJE EN CONTACTO CON EL DOCENTE (ACD)	EXPLICACIÓN DEL DOCENTE	12 horas
2. Organización y manejo de archivos		APRENDIZAJE PRÁCTICO EXPERIMENTAL - ASIGNATURA (APE/A)	TRABAJO EN LABORATORIO	6 horas
3. Búsqueda y ordenamientos externos				
4. Acceso a datos		APRENDIZAJE AUTÓNOMO (AA)	TAREAS EN CASA	19 horas
5. TÉCNICAS DE COMPRESIÓN DE ARCHIVOS				
1. Conceptos de compresión de datos	5	APRENDIZAJE EN CONTACTO CON EL DOCENTE (ACD)	EXPLICACIÓN DEL DOCENTE	10 horas
2. Técnicas de compresión			PRUEBA 2	2 horas
3. Medidas de calidad		APRENDIZAJE PRÁCTICO EXPERIMENTAL - ASIGNATURA (APE/A)	TRABAJO EN LABORATORIO	4 horas
		APRENDIZAJE AUTÓNOMO (AA)	TAREAS EN CASA	19 horas
6. TÉCNICAS DE ENCRIPCIÓN DE ARCHIVOS				

SUB-UNIDADES	Nro. SESIÓN	COMPONENTE DE APRENDIZAJE	ACTIVIDADES DE APRENDIZAJE	
1. Problemas de seguridad  2. Criptografía  3. Principios criptográficos fundamentales  4. Algoritmos de clave simétrica y de clave pública  5. Firmas digitales	6	APRENDIZAJE EN CONTACTO CON EL DOCENTE (ACD)	EXPLICACIÓN DEL DOCENTE	10 horas
		APRENDIZAJE PRÁCTICO EXPERIMENTAL - ASIGNATURA (APE/A)	TRABAJO EN LABORATORIO	4 horas
		APRENDIZAJE AUTÓNOMO (AA)	TAREAS EN CASA	15 horas
		APRENDIZAJE EN CONTACTO CON EL DOCENTE (ACD)	64 horas	
		APRENDIZAJE PRÁCTICO EXPERIMENTAL - ASIGNATURA (APE/A)	32 horas	
		APRENDIZAJE AUTÓNOMO (AA)	96 horas	
		Total Planificación:	192 horas	

## RECURSOS O MEDIOS PARA EL APRENDIZAJE

Equipos, materiales, instrumentos tecnológicos, reactivos, entre otros, que serán utilizados durante el desarrollo de la asignatura.

- Herramienta de Desarrollo (IDE) Java
- Herramientas para modelado de clases
- Diapositivas preparadas por la docente
- Sistema de Gestión del Aprendizaje (evirtual) de la Universidad de Cuenca como repositorio del material utilizado en el curso, así como para receptar trabajos, evaluaciones y otros tipos de actividades.
- Material Bibliográfico especificado en las referencias
- HERRAMIENTAS TECNOLÓGICAS (Metimeter, kahoot, entre otros) PARA LOGRAR UN APRENDIZAJE EFECTIVO Y EFICIENTE

## CRITERIOS PARA LA ACREDITACIÓN DE LA ASIGNATURA

Parámetros de acreditación, tomando como referencia los Resultados de Aprendizaje (RdA's), indicadores y criterios de evaluación planteados y en base a la normativa de evaluación y calificaciones vigente en la Universidad de Cuenca y Consejo de Educación Superior (CES).

CRITERIO GENERAL DE ACREDITACIÓN	PUNTAJE
PRUEBAS	20
EXAMENES	50
TRABAJO	30
<b>TOTAL:</b>	100

	DETALLE DE CRITERIOS DE ACREDITACIÓN	PUNTAJE / CRITERIO GENERAL	
C94	APROVECHAMIENTO I		
	Prueba 1	10	PRUEBAS
	Trabajos hasta el Interciclo	15	TRABAJO
C95	INTERCICLO		
	Examen interciclo	20	EXAMENES
C96	APROVECHAMIENTO II		
	Prueba 2	10	PRUEBAS
	Trabajos de la segunda parte del ciclo	15	TRABAJO
C97	FINAL		
	Examen Final	30	EXAMENES

	DETALLE DE CRITERIOS DE ACREDITACIÓN	PUNTAJE / CRITERIO GENERAL
C98	SUSPENSIÓN	
	Total:	100

## TEXTOS U OTRAS REFERENCIAS REQUERIDAS PARA EL APRENDIZAJE DE LA ASIGNATURA

*Libros, revistas, bases digitales, periódicos, direcciones de Internet y demás fuentes de información, pertinentes y actuales.*

### BÁSICA

1. Schildt H., Java: A Beginner's Guide, Ninth Edition, McGraw-Hill, 2022.
2. Marti, R. Clean Code. A Handbook of Agile Software Craftsmanship, Prentice Hall, 2009.
3. Khurana, R. Data and File Structure, 2nd ed. Vikas Publishing House, Delhi, India, 2016.
4. Sayood, K., Introduction to data compression, 4th ed. ed. Morgan Kaufmann, Waltham, MA, 2012.

### COMPLEMENTARIA

1. Goodrich, M.T., Tamassia, R., Goldwasser, M.H., Data structures and algorithms in Java, 6th ed. Wiley, Singapore, 2015.

---

**Docente:** GRANDA JUCA MARIA FERNANDA

---

**Director:** VEINTIMILLA REYES JAIME EDUARDO

**Finalizado:** 19/3/2024

**Publicado:** 27/3/2024