

PROGRAMA DE CURSO PROGRAMACIÓN AVANZADA

A. Antecedentes generales del curso:

Departamento	Ingeniería Eléctrica (DIE)					
Nombre del curso	Programación avanzada	Código	EL4203	Créditos	6	
Nombre del curso en inglés	<i>Advanced programming</i>					
Horas semanales	Docencia	3	Auxiliares	2	Trabajo personal	5
Carácter del curso	Obligatorio	X		Electivo		
Requisitos	EL3102 Sistemas Digitales, EL4114 Optimización					

B. Propósito del curso:

El curso de Programación avanzada tiene como propósito que los y las estudiantes diseñen y codifiquen algoritmos en diversos lenguajes de programación aplicables a problemas en diversas áreas de la Ingeniería Eléctrica, ejecutando y documentando las soluciones que permitan la replicabilidad de estas. Para esto, utilizan herramientas de programación orientada a objetos, estructuras de datos, métodos de ordenamiento y búsqueda, y metodologías de diseño y programación, con foco en la integración de las herramientas de software en los sistemas y servicios de la ingeniería eléctrica.

Asimismo, se busca habilitar a los y las estudiantes en competencias de programación estructurada, con énfasis en el desarrollo de códigos óptimos en cuanto a su eficiencia y eficacia, útiles en la implementación de dispositivos inteligentes, integrando aspectos ético-profesionales y estándares de calidad asociados.

El curso tributa a las siguientes competencias específicas (CE) y genéricas (CG):

CE1: Formular, analizar, simular y usar modelos físico-matemáticos que caractericen sistemas dinámicos y fuentes de incertidumbre.

CE2: Concebir y aplicar conocimientos de ciencias físicas y matemáticas para el desarrollo de soluciones tecnológicas a problemáticas de la Ingeniería Eléctrica y áreas afines.

CE5: Resolver problemas y optimizar soluciones en el ámbito de la Ingeniería Eléctrica, utilizando conceptos, enfoques y metodologías apropiadas.

CG1: Comunicación académica y profesional

Comunicar en español de forma estratégica, clara y eficaz, tanto en modalidad oral como escrita, puntos de vista, propuestas de proyectos y resultados de investigación fundamentados, en situaciones de comunicación compleja, en ambientes sociales, académicos y profesionales.

CG3: Compromiso ético

Actuar de manera responsable y honesta, dando cuenta en forma crítica de sus propias acciones y sus consecuencias, en el marco del respeto hacia la dignidad de las personas y el cuidado del medio social, cultural y natural.

C. Resultados de aprendizaje:

Competencias específicas	Resultados de aprendizaje
CE1, CE5	RA1: Diseña e implementa algoritmos para resolver problemas de la Ingeniería Eléctrica, utilizando herramientas de programación orientada a objetos, estructuras de datos, métodos de ordenamiento y búsqueda, así como metodologías de diseño y programación.
CE2, CE5	RA2: Analiza diversos algoritmos, seleccionando el óptimo que por su eficiencia permite resolver problemas en el ámbito de la Ingeniería Eléctrica, dadas las restricciones del problema a solucionar.
CE1, CE2, CE5	RA3: Aplica criterios técnicos de las metodologías de diseño y herramientas computacionales al desarrollo de un proyecto informático, integrando, además, aspectos ético-profesionales y estándares de calidad asociados.
CE2	RA4: Codifica algoritmos en diversos lenguajes de programación aplicables a problemas en diversas áreas de la Ingeniería Eléctrica, ejecutando soluciones que permitan su replicabilidad.
Competencias genéricas	Resultados de aprendizaje
CG1	RA5: Reporta técnicamente, de forma clara y precisa, los resultados de la codificación de los algoritmos, adjuntando la documentación de las soluciones de dichas codificaciones para su replicabilidad, en distintos soportes digitales.
CG3	RA6: Analiza aspectos ético-profesionales y estándares de calidad relacionados con el desarrollo de un proyecto informático en cuanto a las metas, objetivos, el diseño y su aplicabilidad.

D. Unidades temáticas:

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
1	RA1, RA4	Fundamentos de Programación para la Ingeniería Eléctrica	2 semanas
Contenidos		Indicador de logro	
1.1. Tipos de datos, funciones y expresiones. 1.2. Programación orientada a objetos. 1.3. Programación iterativa. 1.4. Recursividad.		El/la estudiante: <ol style="list-style-type: none"> Identifica tipos de datos, funciones y expresiones, a partir de ejemplos. Utiliza conceptos de programación orientada a objetos (clases, estado) en el diseño de algoritmos. Diseña algoritmos para resolver problemas de diversa naturaleza en Ingeniería Eléctrica, usando sintaxis de lenguaje de programación. Implementa algoritmos en un lenguaje de programación, usando programación orientada a objetos, programación iterativa y recursividad. 	
Bibliografía de la Unidad		[1] - Chapter 1: Introduction; Chapter 2: Recursion and Backtracking. [3] - Chapter 7: Object-Oriented programming part 1.	

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
2	RA1, RA4	Estructuras y tipos de datos	3 semanas
Contenidos		Indicador de logro	
2.1. Arreglos. 2.2. Punteros y listas enlazadas. 2.3. Árboles. 2.4. Pilas y colas. 2.5. Representación de matrices y vectores. 2.6. Casos de estudio de estructuras y tipos de datos.		El/la estudiante: <ol style="list-style-type: none"> Distingue diferentes tipos de datos, considerando sus restricciones o limitaciones. Selecciona la estructura de datos más adecuada para la resolución de problemas en Ingeniería Eléctrica. Utiliza e implementa tipos de datos en un lenguaje de programación aplicables a problemas de la Ingeniería Eléctrica. Programa código genérico por medio de punteros a funciones. 	
Bibliografía de la unidad		[1] Chapter 5: Queues; Chapter 6: Trees. [2] Chapter 3: Objects, types and values; Chapter 4: Computation; Chapter 18: Vectors and arrays.	

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
3	RA1, RA4	Búsquedas y ordenamientos	3 semanas
Contenidos		Indicador de logro	
3.1. Búsqueda secuencial y binaria. 3.2. Árboles de búsqueda binaria y balanceados. 3.3. Hashing. 3.4. Algoritmos de ordenación. <i>(quicksort, heapsort, mergesort, etc.).</i> 3.5. Casos de estudio de búsqueda y ordenamientos.		El/la estudiante: <ol style="list-style-type: none"> 1. Distingue y analiza diferentes métodos para búsqueda y ordenamiento. 2. Selecciona los métodos de búsqueda y ordenamiento más adecuados para resolver un problema. 3. Resuelve problemas de Ingeniería Eléctrica, procesando información mediante búsquedas y ordenamientos. 4. Implementa la solución de problemas en un lenguaje de programación. 5. Aplica los métodos de búsquedas y ordenamientos, por ejemplo, en problemas de optimización y en análisis de datos. 	
Bibliografía de la unidad		[1] - Chapter 6: Trees; Chapter 10: Sorting; Chapter 11: Searching; Chapter 14: Hashing	

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
4	RA2, RA4	Diseño y análisis de algoritmos	2 semanas
Contenidos		Indicador de logro	
4.1. Conceptos matemáticos. 4.2. Notación "O". 4.3. Análisis de algoritmos sencillos. 4.4. Programación dinámica. 4.5. Casos de estudio de diseño y análisis de algoritmos.		El/la estudiante: <ol style="list-style-type: none"> 1. Efectúa análisis matemático de algoritmos, eligiendo el enfoque según el problema a resolver. 2. Evalúa la eficiencia de la implementación de la solución en cuanto al uso de recursos computacionales, por ejemplo, uso de memoria y tiempo de ejecución. 3. Compara y optimiza algoritmos de programación, a partir de ejemplos en diferentes áreas de estudio. 4. Demuestra matemáticamente qué tan eficiente es la solución algorítmica propuesta para un problema o caso de aplicación de la Ingeniería Eléctrica. 5. Aplica programación dinámica para la evaluación de complejidad computacional en la optimización de algoritmos. 	
Bibliografía de la unidad		[1] -Chapter 1: Introduction; Chapter 19: Dynamic programming; Chapter 20: Complexity classes. [4] -Chapter 13: Recursion; Chapter 14: Dynamic programming.	

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
5	RA3, RA4	Estándares y metodologías de proyectos de software	1,5 semanas
Contenidos		Indicador de logro	
5.1. Normas de codificación. 5.2. Estilos de programación. 5.3. Sistemas de versiones. 5.4. Conceptos de verificación y validación de software.		El/la estudiante: <ol style="list-style-type: none"> 1. Identifica y analiza algunas metodologías y herramientas usadas para el desarrollo de software. 2. Programa mediante el uso de metodologías de diseño y herramientas computacionales, considerando normas de codificación y estándares de calidad para el diseño. 3. Valida la ejecución del software, mediante técnicas y herramientas de debugging para diagnosticar errores de programación. 4. Evalúa y califica la calidad del proyecto de software, considerando estándares técnicos y ético-profesionales tales como tiempos de ejecución y el uso de memoria. 	
Bibliografía de la Unidad		[5] - Chapter 2: Software project management; Chapter 7: Software testing; Chapter 11: Software reliability and dependability.	

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
6	RA3, RA4, RA5, RA6	Aplicaciones de programación a la Ingeniería Eléctrica	3,5 semanas
Contenidos		Indicador de logro	
6.1. Programación 'embebida' para Robótica e Instrumentación. 6.2. Problemas de optimización en el área de Energía. 6.3. Programación de Redes, área de Comunicaciones. 6.4. Programación de sistemas inteligentes de Control. 6.5. Programación para análisis estadístico de datos.		El/la estudiante: <ol style="list-style-type: none"> 1. Programa algoritmos, aplicando conceptos de la programación orientada a objetos, estructuras y tipos de datos, métodos de búsquedas y ordenamientos. 2. Diseña programas eficientes para la resolución de problemas en Sistemas embebidos, Problemas de optimización clásicos con restricciones estáticas, redes de comunicaciones, Estadística descriptiva e inferencial sobre datos, entre otros, considerando estándares de calidad del software. Los programas que diseñan los estudiantes corresponden a desafíos de programación con enunciados específicos por áreas de control, energía, automatización, comunicaciones y de inteligencia computacional. 3. Ejecuta, valida y documenta los diseños y algoritmos para su replicabilidad. 4. Evalúa la solución considerando estándares ético-profesionales tales como sustentabilidad ambiental y social de los proyectos de programación y calidad de estos. 	

	5. Reporta en diversos soportes digitales los resultados de los desarrollos en las distintas áreas de la Ingeniería Eléctrica.
Bibliografía de la unidad	[2] - Chapter 24: Numerics; Chapter 25: Embedded system programming [6] -Chapter 9: Root finding and nonlinear sets of equations; Chapter 10: Minimization or maximization of functions; Chapter 13: Fourier and spectral applications; Chapter 14: Statistical description of data; Chapter 15: Modeling of data.

E. Estrategias de enseñanza - aprendizaje:

<p>El curso considera las siguientes estrategias:</p> <ul style="list-style-type: none"> ● Clases expositiva. ● Resolución de problemas. ● Análisis de casos.
--

F. Estrategias de evaluación:

El curso considera diversas instancias de evaluación:	
Tipo de evaluación	Resultado de aprendizaje asociado a la evaluación
▪ Entrega parcial 1 del proyecto (15%).	Se evalúan los RA1 y RA2.
▪ Entrega parcial 2 del proyecto (15%).	Se evalúa el RA3.
▪ Controles (30%).	Se evalúan los RA1, RA2, RA3 y RA4.
▪ Examen final proyecto (40%).	Se evalúan los RA4, RA5 y RA6.
<p><i>Al inicio de cada semestre el académico o académica informará al y la estudiante sobre los tipos y cantidad de evaluaciones, así como las ponderaciones correspondientes.</i></p>	

G. Recursos bibliográficos:

Bibliografía obligatoria:

- [1] Karumanchi, Narasimha (2015). Data Structure and Algorithmic Thinking with Python: Data Structure and Algorithmic Puzzles. Publisher: CareerMonk Publications; 1st edition.
 [2] Stroustrup, Bjarne (2014). Programming: Principles and Practice Using C++. Publisher: Addison-Wesley Professional; second edition.

Bibliografía complementaria:

- [3] Chan, Jamie (2015). Learn C# Fast and Learn It Well. Master C# Programming with a unique Hands-On Project. Publisher: Learn Coding Fast.
 [4] Green, Bradley (2013). Programming Problems: Advanced Algorithms (Volume 2). Publisher: CreateSpace Independent Publishing Platform.
 [5] O'Regan, Gerard (2017). Concise Guide to Software Engineering: From Fundamentals to Application Methods (Undergraduate Topics in Computer Science). Publisher: Springer; 1st ed..
 [6] William H. Press, Saul A. Teukolsky, et al. (2007). Numerical Recipes 3rd Edition: The Art of Scientific. Publisher: Cambridge University Press; 3 edition.

H. Datos generales sobre elaboración y vigencia del programa de curso:

Vigencia desde:	Otoño, 2021
Elaborado por:	Sandra Céspedes, Alberto Castro, Claudia Rahmann
Validado por:	CTD ampliado Eléctrica
Revisado por:	Área de Gestión Curricular