



UNIVERSIDAD AUTÓNOMA DE SINALOA
FACULTAD DE CIENCIAS DE LA TIERRA Y EL ESPACIO
LICENCIATURA EN INGENIERÍA EN AERONÁUTICA



PROGRAMA DE ESTUDIO

1. DATOS DE IDENTIFICACIÓN			
UNIDAD DE APRENDIZAJE O MÓDULO:	FUNDAMENTOS DE PROGRAMACIÓN		
Clave:	19205		
Ubicación:	Semestre II	Área: Básico disciplinar	
Horas y créditos:	Teóricas: 40	Prácticas: 40	Estudio Independiente: 80
	Total de horas: 160		Créditos: 10
Competencia(s) del perfil de egreso al que aporta:	CG10. Asume con responsabilidad y ética el manejo de las Tecnologías del Aprendizaje y el Conocimiento y es capaz de reconducir las Tecnologías de la Información y Comunicación para la adquisición y actualización del conocimiento de manera permanente para su vida y su profesión. CE2. Modelar componentes de Aeronáutica a través del diseño asistido por computadora para optimizar su comportamiento con propuestas innovadoras y actitud creativa.		
Unidades de aprendizaje relacionadas:	Lenguajes de Programación		
Responsable(s) de elaborar el programa:	M.C. Cruz Isabel Bernal Salgueiro		Fecha: 2023
Responsable(s) de actualizar el programa:	DR. Jesús Guadalupe Monjardín Quevedo DR. Carlos Alberto Martínez Félix		Fecha: Enero 2024
2. PROPÓSITO			
El alumno adquiere los conocimientos básicos y las habilidades necesarias para comprender y aplicar los principios fundamentales de la programación. Además, se introduce a la evolución de la programación, los conceptos básicos, y las características de los lenguajes de programación. En este sentido, aprende sobre la estructura básica de un programa y los elementos del lenguaje de programación. Por consiguiente, adquiere habilidades en la resolución de problemas, la construcción de algoritmos, la sintaxis básica y semántica, las estructuras de control, el manejo de entrada/salida y las estrategias de depuración. También explora los diferentes paradigmas de programación, sus usos y las tendencias actuales. Finalmente, analiza el impacto del cómputo en diferentes áreas de la ingeniería y otras disciplinas, así como las tendencias de desarrollo de software y sus aplicaciones.			
3. SABERES			
Teóricos:	<ul style="list-style-type: none">Comprende conocimientos sobre la evolución de la programación, los conceptos básicos, los paradigmas de programación existentes, así como las características de los lenguajes de programación.		



PROGRAMA DE ESTUDIO

	<ul style="list-style-type: none">● Analiza la lógica necesaria para el desarrollo de algoritmos a partir de la solución de problemas numéricos y secuenciales empleando herramientas de aprendizaje tales como diagramas de flujo y pseudocódigo.● Concibe la estructura básica de un programa y los elementos del lenguaje, como tipos de datos, literales, constantes, variables, identificadores, parámetros, operadores y salida de datos.
Prácticos:	<ul style="list-style-type: none">● Aplica habilidades en la resolución de problemas, incluyendo la definición, planteamiento y modelado del problema.● Diseña algoritmos para resolver problemas, construye el modelo computacional y compone los algoritmos paso a paso.● Desarrolla habilidades fundamentales para la construcción de código, como la comprensión de la sintaxis básica y semántica, la manipulación de variables, expresiones y asignaciones, así como el uso de estructuras de control condicional e iterativo.
Actitudinales:	<ul style="list-style-type: none">● Crea habilidades como el pensamiento lógico, la creatividad, la perseverancia y la atención.● Demuestra una actitud de resolución de problemas y un enfoque metódico en el desarrollo de soluciones.● Incluye la capacidad de trabajo en equipo y la comunicación efectiva en el ámbito de la programación.

4. CONTENIDOS

<p>1.- Introducción a la Programación</p> <ul style="list-style-type: none">1.1 Conceptos básicos.1.2 Evolución de la programación.1.3 Beneficios de la programación (a la sociedad, a la industria, a la medicina, entre otros).1.4 Características del lenguaje de programación.1.5 Estructura básica de un programa.1.6 Elementos del lenguaje: tipos de datos, literales, constantes, variables, identificadores, parámetros, operadores y salida de datos.1.7 Traducción de un programa: compilación, enlace, ejecución y errores. <p>2.- Resolución de problemas</p> <ul style="list-style-type: none">2.1 Definición, planteamiento y modelado del problema.2.2 Algoritmos para la resolución del problema.2.3 Definición del modelo computacional.2.4 Refinamiento del algoritmo paso a paso. <p>3.- Fundamentos para la construcción de código a partir del algoritmo</p> <ul style="list-style-type: none">3.1 Sintaxis básica y semántica.3.2 Variables, constantes, tipos de datos, expresiones y asignación.3.3 Estructuras de control condicional e iterativo.3.4 Funciones y paso de parámetros.3.5 Manejo de archivos y E/S de datos.3.6 Estrategias de depuración. <p>4.- Paradigmas de programación</p> <ul style="list-style-type: none">4.1 Programación estructurada.
--



- 4.2 Programación orientada a objetos.
- 4.3 Programación lógica.
- 4.4 Programación en paralelo.
- 4.5 Principales usos de los paradigmas para la solución de problemas.
- 4.6 Nuevas tendencias.

5.- Cómputo aplicado a diferentes áreas de la ingeniería y otras disciplinas

- 5.1 Tendencia de desarrollo de software.
- 5.2 Aplicaciones.

5. ACTIVIDADES PARA DESARROLLAR LAS COMPETENCIAS

Actividades del docente:

- Realiza una sesión de discusión sobre las temáticas.
- Organiza un debate sobre los conceptos básicos de la programación, donde los estudiantes discutan y definan términos clave.
- Fomenta la investigación individual o en grupos sobre los beneficios de la programación en diferentes áreas, y luego compartir y debatir los hallazgos en clase.
- Realiza una actividad práctica en la que los estudiantes analicen las características de diferentes lenguajes de programación y los comparen.
- Presenta a los estudiantes problemas de la vida real que requieran soluciones algorítmicas y guiarlos en el proceso de definición, planteamiento y modelado de dichos problemas.
- Realiza ejercicios prácticos en los que los estudiantes diseñen algoritmos para resolver problemas específicos, y luego los compartan y discutan en clase.
- Guía a los estudiantes en el proceso de refinamiento de algoritmos paso a paso, brindándoles retroalimentación y orientación.
- Organiza actividades en las que los estudiantes practiquen la declaración y manipulación de variables, tipos de datos, expresiones y asignaciones.
- Presenta ejemplos de estructuras de control condicional e iterativo y guiar a los estudiantes en la implementación de algoritmos que los utilicen.
- Organiza charlas o presentaciones sobre los diferentes paradigmas de programación, donde los estudiantes investiguen y compartan información sobre sus características y usos.
- Realiza ejercicios prácticos en los que los estudiantes implementen soluciones utilizando diferentes paradigmas de programación.
- Fomenta la discusión y el debate sobre las ventajas y desventajas de cada paradigma, así como sobre las nuevas tendencias en programación.
- Invita a profesionales y expertos en diferentes áreas de la ingeniería y disciplinas relacionadas para que compartan casos de estudio y aplicaciones prácticas del cómputo.
- Realiza investigaciones individuales o en grupos sobre las tendencias de desarrollo de software en áreas específicas y presentar los resultados en clase.
- Organiza mesas redondas o debates sobre las aplicaciones del cómputo en diferentes áreas, donde los estudiantes compartan sus perspectivas y experiencias.



Actividades del estudiante:

- Analiza el contexto básico de la temática general de programación.
- Razona y elabora definiciones de conceptos básicos.
- Busca y presenta los beneficios de la programación en un área específica.
- Analiza y compara las características de diferentes lenguajes de programación.
- Desarrolla el esquema básico de la solución para un problema en particular.
- Explica los pasos necesarios de traducción de un programa simple.
- Modela y diseña algoritmos para resolver problemas de la vida real.
- Declara y manipula variables, tipos de datos y expresiones.
- Implementa algoritmos en un lenguaje de programación particular.
- Desarrolla soluciones implementando estructuras de control cíclicas en algoritmos.
- Trabaja en equipo para discutir y definir los paradigmas de programación existentes.
- Codifica programas utilizando la sintaxis y semántica de un lenguaje.
- Analiza la aplicación de los paradigmas de programación en la vida real.
- Investiga casos de estudio y tendencias tecnológicas.
- Discute aplicaciones del cómputo en diferentes disciplinas.

6. EVALUACIÓN DE LAS COMPETENCIAS

6.1. Criterios de desempeño	6.2 Portafolio de evidencias
<ul style="list-style-type: none"> ● Cuadro Sinóptico ● Examen Escrito ● Solución de Ejercicios <ul style="list-style-type: none"> ● Exposición ● Proyecto Integrador <ul style="list-style-type: none"> ● Cuestionario 	<ul style="list-style-type: none"> ● Guía de Observación. ● Reactivos. ● Solucionario de problemas y algorítmicos. ● Rubricas.

6.3. Calificación y acreditación:

- Exámenes parciales y exposición: 50%
- Tareas: 15%
- Participación: 5%
- Proyecto final: 30%

El alumno deberá asistir al menos el 80% de las sesiones para tener derecho a la evaluación y calificación que favorezca su acreditación. Siendo la calificación final mínima de 6.

7. RECURSOS DIDÁCTICOS

Materiales de lectura.
 Ejemplos y ejercicios para la elaboración de algoritmos y resolución de problemas.
 Problemas reales para análisis y discusión.
 Biblioteca y accesos libres virtuales.



UNIVERSIDAD AUTÓNOMA DE SINALOA
FACULTAD DE CIENCIAS DE LA TIERRA Y EL ESPACIO
LICENCIATURA EN INGENIERÍA EN AERONÁUTICA



PROGRAMA DE ESTUDIO

Plataformas educativas (Aula Virtual UAS, Moodle).
Recursos materiales en aulas (proyector, internet).

8. FUENTES DE INFORMACIÓN

Bibliografía básica

Autor(es)	Título	Editorial	Año	URL o biblioteca digital donde está disponible
CAIRÓ Osvaldo	Metodología de la Programación. Algoritmos, Diagramas de Todos Flujo y Programas	Alfaomega	2003	https://fliphtml5.com/jhkbl/owgt/basic#google_vignette
FELLEISEN Matthias, FINDLET Robert Bruce	How to Design Programs. An Introduction to Programming and Computing	MIT Press	2001	https://mitpress.mit.edu/9780262534802/how-to-design-programs/
SZNAJDLEDER Pablo	Algoritmos a fondo: con implementación en C y JAVA	Alfaomega	2012	https://www.inacap.cl/web/material-apoyo-cedem/alumno/Electricidad/Algoritmos_a_fondo_con_implementation_en_c_y_java.pdf

Bibliografía complementaria

Autor(es)	Título	Editorial	Año	URL o biblioteca digital donde está disponible
Jorge Espinal	Manual de Introducción a la Programación	SDQ-Training Center	2015	https://sdq.com.do/descarga/Manual%20Introduccion%20a%20Programacion%20-%20SDQ%20Training%20Center%202015.pdf
Carlos Casado Martínez	Introducción a la Programación	UOC	2009	https://openaccess.uoc.edu/bitstream/10609/9621/1/Introdu



UNIVERSIDAD AUTÓNOMA DE SINALOA
FACULTAD DE CIENCIAS DE LA TIERRA Y EL ESPACIO
LICENCIATURA EN INGENIERÍA EN AERONÁUTICA



PROGRAMA DE ESTUDIO

				ccion_a_la_Programacion.pdf
Álvarez Escudero Juan Jesús	Introducción a la Programación	UNAM	2008	https://www.uacm.edu.mx/portals/0/repository/Introduccion_a_la_programacion%20(2).078198f7-f08e-40b0-b16a-906f24ff33c4.pdf
9. PERFIL DEL DOCENTE				
Licenciatura en informática o ciencias de la computación, Ingeniería en sistemas o afines, maestría y/o doctorado en informática, ciencias de la computación o información. Además, contar con habilidades desarrolladas en pensamiento crítico, empatía, comunicación fluida frente al grupo, organizar actividades así como adaptabilidad al aprendizaje para ajustar los métodos de enseñanza con el objetivo de satisfacer las necesidades dinámicas de los estudiantes.				