



UNIVERSIDAD AUTÓNOMA DE SINALOA
FACULTAD DE CIENCIAS FÍSICO-MATEMÁTICAS
CARRERA: INGENIERÍA ELECTRÓNICA



PROGRAMA DE ESTUDIOS

1. DATOS DE IDENTIFICACIÓN			
UNIDAD DE APRENDIZAJE	LENGUAJE DE PROGRAMACIÓN		
Clave:	1247		
Semestre:	II		
Eje Curricular:	<input type="checkbox"/> Básica <input checked="" type="checkbox"/> Profesionalizante		
Área:	<input type="checkbox"/> Física-Matemática <input type="checkbox"/> Cs. Sociales y Humanidades <input type="checkbox"/> Idiomas <input checked="" type="checkbox"/> Básico Profesional <input type="checkbox"/> Profesional		
Horas y créditos:	Teóricas: 40	Prácticas: 40	Estudio Independiente: 16
	Total de horas: 96		Créditos: 6
Tipo de curso:	Teórico ()	Teórico-práctico (x)	Práctico ()
Competencias del perfil de egreso a la que aporta	<p>Diseña sistemas electrónicos analógicos y digitales para resolver problemas del entorno haciendo uso de diversas tecnologías atendiendo las normas y reglamentos para su uso.</p> <p>Identifica y evalúa tecnologías electrónicas emergentes para ser consideradas en futuros diseños de forma continua y oportuna.</p> <p>Desarrolla software y firmware para dispositivos electrónicos atendiendo las normas de calidad y reglamentación establecidas.</p> <p>Desarrolla telecomunicaciones, instrumentación y control para resolver problemas del sector industrial de forma eficaz y atendiendo los criterios de calidad necesarios.</p>		
Componentes	<p>Desarrollo de firmware para microcontroladores diversos.</p> <p>Desarrollo de firmware para plataformas de arquitectura reconfigurable.</p> <p>Desarrollo de software para sistemas embebidos.</p> <p>Manejo de sistemas operativos diversos para sistemas embebidos.</p>		
Unidades de aprendizaje relacionadas	Tecnologías de la información y de la comunicación, programación orientada a objetos, sistemas embebidos, microcontroladores.		
Responsables de elaborar y/o actualizar el	Lic. Erick Alfredo Urías Bernal Dr. Carlos Duarte Galván		

programa:		
Fecha de:	Elaboración: agosto de 2017	Actualización:
2. PROPÓSITO		
Aportar al estudiante la capacidad para desarrollar aplicaciones en un lenguaje de programación de alto nivel para la solución de problemas de ingeniería.		
3. SABERES		
Teóricos:	<ul style="list-style-type: none"> - Conocer la teoría de programación de software, su relación con el mundo actual y la aplicación en resolución de problemas de ingeniería. - Conocer el lenguaje de programación orientado a objetos - Comprender la diferencia entre programación estructurada y programación orientada a objetos. - Conocer las características de la programación orientada a objetos, uso de palabras reservadas, procedimientos y librerías disponibles en la WEB. 	
Prácticos:	<ul style="list-style-type: none"> - Desarrollar un proceso de abstracción que permita simplificar un problema, discriminar partes y trabajar de manera puntual en la resolución del problema. - Describir un sistema como un conjunto de bloques interconectados con una lógica que permita entender con mayor claridad cómo funcionan las etapas del sistema. - Interconectar el software desarrollado con una etapa electrónica de hardware a través de algún periférico de la computadora. - Trabajar en diferentes sistemas operativos. 	
Actitudinales:	<ul style="list-style-type: none"> - Valorar el papel de la Ciencia en el entendimiento de la naturaleza. - Demostrar rigor científico en el planteamiento y solución de problemas. - Actitud de participación en la solución de ejercicios. - Cultivar el autoaprendizaje. - Desarrollar la lectura de textos científicos. - Actitud reflexiva en la asimilación de nuevos conceptos. - Valorar la potencialidad de la mecánica estadística como puente para la ciencia interdisciplinaria. 	
4. CONTENIDO TEMÁTICO		
1. Fundamentos de programación <ul style="list-style-type: none"> 1.1. Sistemas operativos 1.2. Importancia de la programación de computadoras 1.3. Evolución y clasificación de los lenguajes de programación 1.4. Interpretes y compiladores 1.5. Diseño de algoritmos 		

2. Elementos del lenguaje de programación

- 2.1. Introducción al entorno de programación
- 2.2. Estructura básica de un programa
- 2.3. Palabras reservadas
- 2.4. Variables y constantes
- 2.5. Tipos de datos
 - 2.5.1. Simples
 - 2.5.2. Compuestos (abstractos)
- 2.6. Despliegue y formateo de datos
- 2.7. Operadores aritméticos, lógicos y de asignación (relacionales)
- 2.8. Control de flujo
- 2.9. Ciclos y ruptura de ciclos

3. Programación modular

- 3.1. Declaración de funciones
 - 3.1.1. Simples
 - 3.1.2. Con parámetros
- 3.2. Uso de bibliotecas y funciones
 - 3.2.1. Entrada y salida
 - 3.2.2. Archivos
- 3.3. Cadenas

4. Estructuras de datos

- 4.1. Arreglos unidimensionales
 - 4.1.1. Concepto y forma general
 - 4.1.2. Arreglos numéricos y de caracteres
- 4.2. Arreglos bidimensionales
 - 4.2.1. Concepto y forma general
 - 4.2.2. Arreglos numéricos y de caracteres
- 4.3. Apuntadores
 - 4.3.1. Concepto
 - 4.3.2. Tipos de apuntadores
 - 4.3.3. Operaciones con apuntadores
 - 4.3.4. Relación de apuntadores con arreglos

5. Aplicaciones de puertos de comunicación

- 5.1. Puertos de comunicación (RS-232, paralelo, USB)
- 5.2. Especificaciones de los puertos RS-232 y paralelo
- 5.3. Transmisión y recepción de datos.

Proyecto final sugerido:

Comunicación a través de puerto serial. Puede ser programar un chat entre dos computadoras comunicadas por el puerto RS-232 (para computadoras modernas utilizar adaptadores USB-Serial, ya sea un cable STEREN o un chip FTDI FT232RL).

También puede asignarse un proyecto para diseñar un programa que adquiera datos por el

puerto RS-232 y los guarde en un archivo de texto.

Diseño de un programa para importar archivos de texto, procesamiento y graficación de los datos procesados.

5. ACCIONES ESTRATÉGICAS PARA EL APRENDIZAJE

Sensibilización y atención: realizar una exposición introductoria de los temas en cada unidad, haciendo mención del contexto histórico en que los conceptos fueron desarrollados, así como de los problemas teóricos o tecnológicos que ayudaron a resolver los temas que se verán en dicha unidad temática.

Recomendar lectura previa de temas selectos, para crear discusiones y debates en torno al tema

En la plataforma virtual: transferencia de información al alumno de algunos temas concretos, entregar al profesor de tareas como programas de práctica, resúmenes y reportes de investigación, apertura de foros de discusión y seguimiento a ellos.

Estrategias y técnicas de aprendizaje: aprendizaje basado en problemas, aprendizaje colaborativo en la resolución de problemas, programación de algoritmos y en exposiciones. Método de proyectos.

6. EVALUACIÓN DEL APRENDIZAJE

6.1. Evidencias de aprendizaje	6.2. Criterios de desempeño	6.3. Calificación y acreditación
<ul style="list-style-type: none">Exámenes por unidad.Reportes de investigación.Exposiciones en clase.Tareas.Entrega de prácticas.	<p>Exámenes por unidad: Explicación clara y concreta de los conceptos relacionados con la materia. Solución correcta de problemas de ingeniería propuestos.</p> <p>Entrega de prácticas: 70% por funcionalidad del circuito electrónico, 30% por el reporte impreso con la descripción del hardware de la práctica.</p> <p>En lo que respecta a los demás criterios de evaluación, se asignará 30% al formato, 40% al contenido y 30% a las conclusiones que el alumno presente.</p>	<p>70% exámenes.</p> <p>30% Prácticas y demás trabajos.</p>

7. FUENTES DE INFORMACIÓN

Fuentes de Información Básica:

Antonakos, J. L., Mansfield, K. C., Carballeira, F. G., & Costoya, F. P. (1997). **Application programming in structured C**: Pearson Educación.

Bronson, Gary, J., C++ para ingeniería y ciencias, International Thomson Editores, Impreso en México, 2000, 862 pp, ISBN: 968-7529-87-3

Aitken, Peter; B.Jones. **“Teach yourself C in 21 Days”**. SAMS Publishing. “Aprendiendo C en 21 días”.

Deitel, Harvey M.; P.J. Deitel. **“C, How to program”**. Prentice Hall. “Cómo programar en C”.

Kernighan, Brian W.; D.M. Ritchie. **“The C Programming Language”**. Prentice Hall. “El lenguaje de programación C”.

Ceballos, Francisco Javier, Enciclopedia del Lenguaje C, Computec-Rama, México, 1994.

Kernighan, B. W.; D.M. Ritchie, Lenguaje de Programación C, Prentice-Hall.

Fuentes de Información Complementaria:

Schildt, Hebert. **“C: The Complete Reference”**. Osborne/McGraw-Hill.”C: Manual de Referencia”.

Pitts, David, La Biblia de Red Hat Linux, Anaya Multimedia, Madrid

Tacker, Arlen B., Lenguajes de programación, McGraw-Hill

Tanembaum, A.S., Sistemas Operativos, análisis y diseño, Prentice-Hall.

8. PERFIL DEL PROFESOR:

Posee un profundo conocimiento de programación en diferentes lenguajes, de manera que le permita conectar los saberes del curso con otras asignaturas, así como con el perfil de egreso del electrónico. Conocer y aplicar las diferentes potencialidades de la programación en la resolución de problemas de ingeniería.

Conoce y aplicar adecuadamente la lógica de programación orienta a objetos a la resolución de problemas de ingeniería.

Demuestra habilidades didácticas de enseñanza y evaluación del aprendizaje.