



UNIVERSIDAD AUTÓNOMA DE SINALOA

FACULTAD DE INGENIERÍA MOCHIS LICENCIATURA EN INGENIERÍA DE SOFTWARE

PROGRAMA DE ESTUDIO



1. DATOS DE IDENTIFICACIÓN			
UNIDAD DE APRENDIZAJE O MÓDULO:	LABORATORIO LENGUAJES DE PROGRAMACIÓN		
Clave:			
Ubicación:	SEMESTRE V	Área: Profesionalizante	
Horas y créditos:	Teóricas: 0	Prácticas: 48	Estudio Independiente: 48
	Total de horas: 96		Créditos: 6
Competencia(s) del perfil de egreso al que aporta:	<p>Genéricas:</p> <p>CG7. Cultiva el compañerismo, el trabajo en equipo y la coordinación de esfuerzos bajo la aspiración de mejorar las tareas académicas, los entornos laborales y la convivencia social en beneficio para la consecución de metas que impactan en las formas de entablar y mantener relaciones humanas positivas.</p> <p>CG8. Asimila, de manera autónoma y convencida, la necesidad de promover conductas que le orienten hacia el desarrollo del saber, del hacer y del convivir como formas trascendentales de la existencia, en lo inmediato y en lo futuro.</p> <p>Específicas:</p> <p>CE13. Aplica conocimientos teóricos y prácticos, en conjunto con las metodologías para la buena construcción de programas y sistemas de software, considerando su análisis y diseño, planeación, procesos y controles dentro de escenarios de pruebas para asegurar confiabilidad, funcionalidad, costo, seguridad, facilidades de mantenimiento y otros aspectos relacionados.</p> <p>CE20. Desarrolla software para diferentes tipos de aplicaciones utilizando técnicas, metodologías y paradigmas de programación en el contexto de los ciclos de vida del software, con los atributos de calidad requeridos.</p>		
Unidades de aprendizaje relacionadas:	Algoritmos y lógica computacional, programación I, programación II, estructura de datos, programación orientada a objetos, graficación, desarrollo de aplicaciones web I, desarrollo de aplicaciones web II.		
Responsable(s) de elaborar el programa:	MIA. JUAN FRANCISCO FIGUEROA PÉREZ		Fecha: 6 de Julio del 2011
Responsable(s) de actualizar el programa:			Fecha:
2. PROPÓSITO			



PROGRAMA DE ESTUDIO

Conocer y aplicar correctamente los conceptos teóricos y prácticos relacionados los principales paradigmas de programación, la programación distribuida y la programación multihilo

3. SABERES

Teóricos:	Identifica cada paradigma de lenguaje de programación, analiza su aplicabilidad según el problema específico a resolver y adquiere práctica en programación con por lo menos un lenguaje de cada paradigma. Conoce e identifica los aspectos y elementos más relevantes del enfoque de principios. Conoce e identifica los aspectos y elementos más relevantes de los enfoques de la programación orientada a objetos y de la programación en paralelo. Conoce e identifica los aspectos y elementos más relevantes de los enfoques de la programación funcional y programación lógica.
Prácticos:	Elabora programas que utilizan los elementos del enfoque de principios. Elabora programas que utilizan elementos de los enfoques de la programación orientada a objetos y de la programación en paralelo. Elabora programas que utilizan elementos de los enfoques de la programación funcional y la programación lógica.
Actitudinales:	Comunicación oral y escrita en la propia lengua. Habilidades de gestión de la información (habilidad para buscar y analizar información proveniente de diversas fuentes). Capacidad de crítica y autocrítica. Trabajo en equipo. Compromiso ético. Capacidad para aprender a trabajar de forma autónoma.

4. CONTENIDOS

Unidad 1 Introducción

- 1.1 Razones para estudiar los conceptos de lenguajes de programación
- 1.2 Criterios de evaluación de un lenguaje
- 1.3 Influencias en el diseño de un lenguaje de programación
- 1.4 Entornos de programación

Unidad 2 Evolución de los principales lenguajes de programación

- 2.1 Zuse's Plankalkül
- 2.2 Pseudocódigos
- 2.3 IBM 704 y Fortran
- 2.4 LISP, Algol 60, COBOL
- 2.5 Basic, PL/I, APL / SNOBOL, SIMULA 67
- 2.6 AALGOL 68, ALGOL, PROLOG, Ada, SmallTalk, C++, Java, Lenguajes de script
- 2.7 C#, lenguajes híbridos.

Unidad 3 Concurrencia

- 3.1 Introducción
- 3.2 Concurrencia a nivel sub programa
- 3.3 Semáforos
- 3.4 Monitores
- 3.5 Pase de mensajes



- 3.6 Hilos en Java
- 3.7 Hilos en C#
- 3.8 Concurrencia en lenguajes funcionales
- 3.9 Concurrencia a nivel de sentencias

Unidad 4 Manejo de excepciones y eventos

- 4.1 Introducción al manejo de excepciones
- 4.2 Manejo de excepciones en C++
- 4.3 Manejo de excepciones en Java
- 4.4 Introducción al manejo de eventos
- 4.5 Eventos en Java
- 4.6 Eventos en C#

Unidad 5 Lenguajes de programación funcionales

- 5.1 Introducción
- 5.2 Funciones matemáticas
- 5.3 Fundamentos de los lenguajes de programación funcionales
- 5.4 Lisp
- 5.5 Scheme
- 5.6 F#
- 5.7 Soporte de programación funcional en los lenguajes imperativos
- 5.8 Comparación entre lenguajes funcionales e imperativos

Unidad 6 Lenguajes de programación lógica

- 6.1 Introducción
- 6.2 Orígenes de Prolog
- 6.3 Elementos básicos de Prolog
- 6.4 Deficiencias de Prolog
- 6.5 Aplicaciones de la programación lógica

5. ACTIVIDADES PARA DESARROLLAR LAS COMPETENCIAS

Actividades del docente:

- Establecer el encuadre del curso al inicio de este.
- Asistir puntualmente a clases y cumplir el tiempo indicado.
- Preparar material didáctico para las clases.
- Aclarar las dudas de los alumnos.
- Asesorar a los alumnos cuando lo soliciten.
- Evaluar el aprendizaje conforme a lo acordado en el encuadre.
- Entregar resultados de las evaluaciones en tiempo y forma.

Actividades del estudiante:

- ❖ Asistir y llegar puntualmente a clases.
- ❖ Tener como mínimo, el 80 % de asistencias.
- ❖ Participar de manera activa en los procesos de enseñanza y aprendizaje.



- ❖ Prepararse para realizar las evaluaciones que se establezcan.

6. EVALUACIÓN DE LAS COMPETENCIAS

6.1. Criterios de desempeño

- Investigación bibliográfica sobre temas selectos y de aplicación.
- Elaboración de programas de práctica en laboratorio.
- Exposición con sesión de preguntas y respuestas.
- Resolución de problemas con libros relacionados con la materia.
- Utilización de lenguaje de programación.
- Evaluación de informes sobre tareas o trabajos de investigación.
- Evaluación en laboratorio.
- Evaluación escrita.
- Desempeño y participación del alumno en el aula.

6.2 Portafolio de evidencias

- Ensayo sobre los aspectos y elementos más relevantes del enfoque de principios.
- Programas con el enfoque de principios.
- Ensayo sobre los aspectos y elementos más relevantes la programación orientada a objetos y la programación en paralelo.
- Programas que utilizan elementos de los paradigmas antes mencionados.
- Ensayo sobre los aspectos y elementos más relevantes la programación funcional y la programación lógica.
- Programas que utilizan elementos de los paradigmas antes mencionados.

6.3. Calificación y acreditación:

Parcial:

- Evaluación escrita.
- Programas elaborados en laboratorio.
- Desempeño y participación.
- Asistencia.

Final:

- Promedio de las evaluaciones parciales.

7. RECURSOS DIDÁCTICOS

8. FUENTES DE INFORMACIÓN

Bibliografía básica

Autor(es)	Título	Editorial	Año	URL o biblioteca digital donde está disponible
Kenneth C. Louden	Lenguajes de programación: Principios y práctica	2da Ed. Thomson.	2003	



UNIVERSIDAD AUTÓNOMA DE SINALOA

FACULTAD DE INGENIERÍA MOCHIS LICENCIATURA EN INGENIERÍA DE SOFTWARE



PROGRAMA DE ESTUDIO

Francisco Charte	Introducción a Microsoft Visual C# 2005	Microsoft Press.	2005	
<i>Bibliografía complementaria</i>				
Autor(es)	Título	Editorial	Año	URL o biblioteca digital donde está disponible
Rolando Burgos	Manual de referencia rápida de LISP	Universidad de concepción	1997	
Javier Suarez	Introducción al prolog		1996	
9. PERFIL DEL DOCENTE				
LICENCIADO EN INFORMÁTICA, LICENCIADO EN SISTEMAS COMPUTACIONALES, LICENCIADO EN INGENIERÍA DE SOFTWARE, INGENIERO EN SISTEMAS COMPUTACIONALES, INGENIERO EN COMPUTACIÓN				