

1.- DATOS DE LA ASIGNATURA

Nombre de la asignatura:	LENGUAJES DE INTERFAZ
Carrera:	INGENIERÍA EN SISTEMAS COMPUTACIONALES
Clave de la asignatura:	SCC-1014
(Créditos) SATCA ¹	2-2-4

2.- PRESENTACIÓN

Caracterización de la asignatura.

La presente asignatura aporta al perfil del Ingeniero en Sistemas Computacionales los conocimientos para el diseño e implementación de interfaces hombre-máquina y máquina-máquina para la automatización de sistemas. El desarrollo, implementación y administración de software de sistemas o de aplicación que cumpla con los estándares de calidad con el fin de apoyar la productividad y competitividad de las organizaciones. Para que desempeñe sus actividades profesionales considerando los aspectos legales, éticos, sociales y de desarrollo sustentable y a la vez le permita poseer las habilidades metodológicas de investigación que fortalezcan el desarrollo cultural, científico y tecnológico en el ámbito de sistemas computacionales y disciplinas afines.

Intención didáctica.

En la primera unidad se pretende dar un panorama general de lo que es el lenguaje ensamblador y donde se puede utilizar. La segunda unidad se da a conocer las principales instrucciones y funciones del lenguaje ensamblador, la forma de estructurar un programa, los parámetros que se tienen que tomar en cuenta realizando ejemplos sencillos.

En la unidad tres el estudiante desarrollará programas en lenguaje ensamblador que haga uso de macros o procedimientos, posteriormente, analizar el funcionamiento interno de los programas desarrollados haciendo el uso del software que permita obtener datos estadísticos sobre el funcionamiento de los mismos. En la última unidad el estudiante obtendrá los conocimientos necesarios para la programación de los puertos de la computadora, así como, poder analizar la interfaz de video del buffer en modo texto. En esta unidad se pretende que el estudiante realice una interfaz de hardware, la cual será la base para la creación de las aplicaciones de software, ya que la interfaz de hardware será controlada mediante la interfaz de software.

¹ Sistema de asignación y transferencia de créditos académicos

3.- COMPETENCIAS A DESARROLLAR

<p>Competencias específicas:</p> <ul style="list-style-type: none">• Desarrollar software para establecer la interfaz hombre-máquina y máquina-máquina.	<p>Competencias genéricas</p> <p>Competencias instrumentales:</p> <ul style="list-style-type: none">• Procesar e interpretar datos.• Representar e interpretar conceptos en diferentes formas.• Modelar fenómenos y situaciones.• Pensamiento lógico, algorítmico, heurístico, analítico y sintético.• Potenciar las habilidades para el uso de tecnologías de información.• Resolución de problemas.• Analizar la factibilidad de las soluciones.• Optimizar soluciones.• Toma de decisiones. <p>Competencias interpersonales.</p> <ul style="list-style-type: none">• Capacidad crítica y autocrítica.• Trabajo en equipo.• Habilidades interpersonales.• Capacidad de trabajar en equipo interdisciplinario. <p>Competencias sistémicas.</p> <ul style="list-style-type: none">• Capacidad de aplicar los conocimientos en la práctica.• Habilidades de investigación.• Capacidad de aprender.• Capacidad de adaptarse a nuevas situaciones.• Capacidad de generar nuevas ideas.• Liderazgo.• Habilidad para trabajar en forma autónoma.• Capacidad para diseñar y gestionar proyectos.• Iniciativa y espíritu emprendedor.
--	---

4.- HISTORIA DEL PROGRAMA

Lugar y fecha de elaboración o revisión	Participantes	Observaciones (cambios y justificación)
Instituto Tecnológico de Saltillo Fecha: del 05 al 09 de octubre de 2009, Saltillo, Coahuila.	Representantes de los Institutos Tecnológicos de: Pinotepa, San Luis Potosí, Superior de Libres, Celaya, Tapachula.	Reunión nacional de Diseño e innovación curricular de la carrera de Ingeniería en Sistemas Computacionales.
Instituto Tecnológico Superior de Pinotepa Fecha: del 12 de octubre al 19 de febrero de 2010	Representante de la Academia de Sistemas Computacionales de Pinotepa, San Luis Potosí, Celaya, Superior de Libres, Tapachula, Huetamo.	Análisis, enriquecimiento y elaboración del programa de estudio propuesto en la Reunión Nacional de Diseño Curricular de la carrera de Ingeniería en Sistemas Computacionales.
Instituto Tecnológico Superior de Poza Rica fecha: del 22 al 26 de febrero de 2010.	Representantes de los Institutos Tecnológicos participantes en el diseño de la carrera de Ingeniería en Sistemas Computacionales: Pinotepa, San Luis Potosí, Celaya, Superior de Libres, Tapachula, Huetamo.	Reunión Nacional de Consolidación Curricular de la carrera de Ingeniería en Sistemas Computacionales.

5.- OBJETIVO GENERAL DEL CURSO

- Desarrollar software para establecer la interfaz hombre-máquina y máquina-máquina.

6.- COMPETENCIAS PREVIAS

- Comprender el funcionamiento general de las computadoras.
- Elaborar algoritmos computacionales para la solución de problemas de cómputo.
- Comprender las características de la arquitectura básica de una computadora.

7.- TEMARIO

Unidad	Temas	Subtemas
1	Introducción al lenguaje ensamblador.	1.1 Importancia de la programación en lenguaje ensamblador. 1.2 El procesador y sus registros internos 1.3 La memoria principal (RAM) 1.4 El concepto de interrupciones 1.5 Llamadas a servicios del sistema 1.6 Modos de direccionamiento 1.7 Proceso de ensamblado y ligado 1.8 Despliegado de mensajes en el monitor
2.	Programación básica	2.1 Ensamblador (y ligador) a utilizar 2.2 Ciclos numéricos 2.3 Captura básica de cadenas 2.4 Comparación y prueba 2.5 Saltos 2.6 Ciclos condicionales 2.7 Incremento y decremento 2.8 Captura de cadenas con formato 2.9 Instrucciones aritméticas 2.10 Manipulación de la pila 2.11 Obtención de cadena con representación decimal 2.12 Instrucciones lógicas 2.13 Desplazamiento y rotación 2.14 Obtención de una cadena con la representación hexadecimal 2.15 Captura y almacenamiento de datos numéricos 2.16 Operaciones básicas sobre archivos de disco
3.	Modularización	3.1 Procedimientos 3.2 Macros
4.	Programación de dispositivos	4.1 El buffer de video en modo texto 4.2 Acceso a discos en lenguaje ensamblador 4.3 Programación del puerto serial 4.4 Programación del puerto paralelo 4.5 Programación híbrida 4.6 Programación de puerto usb

8.- SUGERENCIAS DIDÁCTICAS

- Usar un portal de Internet para apoyo didáctico de la materia, el cual cuente por lo menos con un foro, preguntas frecuentes, material de apoyo y correo electrónico.
- Trabajar en grupos pequeños, para sintetizar y construir el conocimiento necesario para resolver problemas relacionados con situaciones reales.
- Solicitar al estudiante propuestas de problemas a resolver y que sean significativas para él.
- Propiciar que el estudiante experimente con diferentes programas encontrados en revistas, Internet y libros de la especialidad, que lo lleven a descubrir nuevos conocimientos.
- Elaborar de manera conjunta con el estudiante una guía de ejercicios para actividades extra clase.
- Plantear problemas reales para que diseñen soluciones utilizando los conceptos de lenguajes de interfaz.
- Desarrollo de un proyecto con aplicación real.
- En la solución de problemas, motivar que seleccione la metodología que permita que la solución computacional sea pertinente y viable.
- Propiciar, en el estudiante, el desarrollo de actividades intelectuales de inducción-deducción y análisis-síntesis, las cuales lo encaminan hacia la investigación, la aplicación de conocimientos y la solución de problemas.
- Llevar a cabo actividades prácticas que promuevan el desarrollo de habilidades para la experimentación, tales como: observación, identificación manejo y control de variables y datos relevantes, planteamiento de hipótesis, de trabajo en equipo.
- Desarrollar actividades de aprendizaje que propicien la aplicación de los conceptos, modelos y metodologías que se van aprendiendo en el desarrollo de la asignatura.
- Propiciar el uso adecuado de conceptos y de terminología científico tecnológica.
- Proponer problemas que permitan al estudiante la integración de contenidos de la asignatura y entre distintas asignaturas, para su análisis y solución.
- Observar y analizar fenómenos y problemáticas propias del campo ocupacional.
- Relacionar los contenidos de esta asignatura con las demás del plan de estudios, para desarrollar una visión interdisciplinaria en el estudiante.
- Presentar proyectos finales.

9.- SUGERENCIAS DE EVALUACIÓN

La evaluación de la asignatura se hará con base en siguiente desempeño:

- Ponderación de tareas.

- Participación y desempeño en el aula y el laboratorio, a través de dinámicas grupales, trabajo individual o en equipo.
- Dar seguimiento al desempeño en el desarrollo del programa (dominio de los conceptos, capacidad de la aplicación de los conocimientos en problemas reales, transferencia del conocimiento).
- Desarrollo de un proyecto que integre todas las unidades de aprendizaje.
- Cumplimiento de los objetivos y desempeño en las prácticas.

10.- UNIDADES DE APRENDIZAJE

Unidad 1: Introducción a los lenguajes de bajo nivel

Competencia específica a desarrollar	Actividades de Aprendizaje
<p>Conocer cómo trabajan los microprocesadores con lenguaje ensamblador y qué aplicación pueden tener.</p>	<ul style="list-style-type: none"> • Investigar la estructura y organización de un procesador (CPU). Discutir y formalizar grupalmente lo investigado. • Analizar los registros más importantes que contiene un CPU, así como las funciones de los mismos. • Describir modos de direccionamiento a memoria y efectuar ejercicios. • Realizar ejemplos sencillos de cómo estructurar un programa fuente y que esté despliegue mensajes en el monitor con instrucciones básicas en lenguaje ensamblador.

Unidad 2: Programación básica

Competencia específica a desarrollar	Actividades de Aprendizaje
<p>Conocer y utilizar las principales instrucciones del lenguaje ensamblador, para realizar determinadas aplicaciones de interfaz.</p>	<ul style="list-style-type: none"> • Realizar investigación y listar las principales instrucciones de programación en lenguaje ensamblador. • Desarrollar programas por medio de prácticas en lenguaje ensamblador, los cuales ejemplifiquen las diferentes instrucciones y funciones básicas así como la forma de estructurarlas.

Unidad 3: Modularización

Competencia específica a desarrollar	Actividades de Aprendizaje
<p>Comprender la importancia de la utilidad de las macros y procedimientos dentro del desarrollo de una aplicación de software.</p>	<ul style="list-style-type: none">• Realizar investigaciones sobre los conceptos macro y procedimiento, analizando sus semejanzas y diferencias.• Analizar el funcionamiento de un programa que no utiliza macros o procedimientos en su funcionamiento, todo esto utilizando un software que permita obtener datos estadísticos del funcionamiento de los programas en depuración.• Desarrollar programas en un lenguaje de programación que haga uso de macros o procedimientos, posteriormente analizar el funcionamiento interno de los programas desarrollados haciendo el uso del software que permita obtener datos estadísticos sobre el funcionamiento de los mismos.

Unidad 4: Programación de dispositivos

Competencia específica a desarrollar	Actividades de Aprendizaje
<p>Programar interfaces de software y hardware para la manipulación de puertos y dispositivos de computadora.</p>	<ul style="list-style-type: none">• Analizar el funcionamiento del buffer de video de una computadora, mediante la lectura en modo texto del mismo.• Desarrollar programas en lenguaje ensamblador para acceder a los dispositivos de almacenamiento de la computadora.• Diseñar una interfaz de hardware utilizando algún tipo de integrado programable (ej. Microcontrolador).• Diseñar una interfaz de software en algún lenguaje de programación para controlar la interfaz de hardware utilizando los puertos paralelos, seriales y USB de la computadora.

11.- FUENTES DE INFORMACIÓN

1. FROUFE, Agustín & Jorge, Patricia. *J2me java 2* , micro edition manual de usuario y tutorial. Alfaomega, Ra-Ma. 2004.
2. PETER Abel, *Lenguaje Ensamblador y programación para PC IBM y Compatibles*, 3ª. Edición, Pearson Prentice Hall, 1966.
3. KIP R. Irvine, *Lenguaje ensamblador para computadoras basadas en Intel*, Pearson Prentice Hall, 5ª edición, 2008.
4. PRIETO, Manuel, *Desarrollo de juegos con j2me java 2*, micro edition. Alfaomega, Ra-Ma. 2005
5. CEBALLOS, Francisco Javier. *Aplicaciones .Net Multiplataforma*. Ed. Ra-Ma, 2004
6. Paul Yao & David Durant, *Programming the .NET Compact Framework in C# and VB.NET* , <http://blog.neuronaltraining.net/?p=676>
7. Programación de juegos para móviles con J2ME.
<http://www.agserrano.com/libros/j2me/j2me.zip>
8. JSRs: Java Specification Requests
<http://jcp.org/en/jsr/all>
9. JSR-82 : Java Bluetooth, <http://www.jsr82.com/>

12.- PRÁCTICAS PROPUESTAS

1. Generar el glosario de conceptos clave de cada tema.
2. Desarrollar mapas conceptuales para cada tema.
3. Desarrollar programas en ensamblador que realicen operaciones básicas en memoria.
4. Programación de dispositivos en lenguaje ensamblador.
5. Desarrollar un proyecto final donde aplique los conocimientos del desarrollo de software de aplicación en lenguaje ensamblador.