

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

COORDINACIÓN GENERAL DE FORMACIÓN PROFESIONAL

PROGRAMA DE UNIDAD DE APRENDIZAJE

I. DATOS DE IDENTIFICACIÓN

1. **Unidad Académica:** Facultad de Ingeniería, Arquitectura y Diseño, Ensenada; Facultad Ciencias Químicas e Ingeniería, Tijuana; y Facultad de Ciencias de la Ingeniería y Tecnología, Valle de las Palmas
2. **Programa Educativo:** Ingeniero en Software y Tecnologías Emergentes
3. **Plan de Estudios:** 2022-1
4. **Nombre de la Unidad de Aprendizaje:** Organización de Computadoras
5. **Clave:** 40003
6. **HC:** 02 **HT:** 00 **HL:** 02 **HPC:** 00 **HCL:** 00 **HE:** 02 **CR:** 06
7. **Etapas de Formación a la que Pertenece:** Básica
8. **Carácter de la Unidad de Aprendizaje:** Obligatoria
9. **Requisitos para Cursar la Unidad de Aprendizaje:** Ninguno



Equipo de diseño de PUA

José de Jesús Zamarripa Topete
Aglay González Pacheco Saldaña
Carelia Guadalupe Gaxiola Pacheco
Leocundo Aguilar Noriega

Vo.Bo. de subdirector(es) de Unidad(es) Académica(s)

Humberto Cervantes De Ávila
Daniela Mercedes Martínez Platas
Noemí Hernández Hernández

Fecha: 12 de marzo de 2020

II. PROPÓSITO DE LA UNIDAD DE APRENDIZAJE

Proporcionar los fundamentos de la Organización de Computadoras e identificar los factores que determinan su desempeño, comprendiendo sus atributos al hacer uso de ellos en el desarrollo de programación con lenguaje de alto y bajo nivel. Se ubica dentro de la etapa básica, con carácter obligatorio y pertenece al área de conocimiento de Infraestructura de Sistemas.

III. COMPETENCIA GENERAL DE LA UNIDAD DE APRENDIZAJE

Desarrollar código en diferentes niveles de programación de computadoras o sistemas basados en procesador, mediante la comprensión de su organización y arquitectura, para proporcionar soluciones óptimas en rendimiento, con responsabilidad y eficiencia.

IV. EVIDENCIA(S) DE APRENDIZAJE

Portafolio con de prácticas de laboratorio junto con reporte que sustente análisis, diseño, implementación y pruebas de los programas.

V. DESARROLLO POR UNIDADES
UNIDAD I. Organización de Computadoras

Competencia:

Identificar los componentes internos de una computadora, mediante el estudio de su evolución histórica, organización, arquitectura y funcionalidad, para reconocer sus capacidades y limitaciones, con actitud crítica y responsable.

Contenido:

Duración: 4 horas

- 1.1 Historia de las computadoras.
- 1.2 Arquitectura Von Neumann.
 - 1.2.1 Diagrama de bloques
 - 1.2.2 Trayectoria de datos
- 1.3 Unidad Central de Proceso.
 - 1.3.1 Ciclo básico.
 - 1.3.2 Unidad de control.
 - 1.3.3 Registros
 - 1.3.4 Unidad Aritmético-Lógica.
- 1.4 Sección de memoria
 - 1.4.1 Jerarquía de memoria
 - 1.4.2 Segmentación y paginación
- 1.5 Entrada/Salida.
- 1.6 Procesadores avanzados
 - 1.6.1 Arquitecturas CISC y RISC.
 - 1.6.2 Arquitectura línea de montaje
 - 1.6.3 Arquitectura Harvard
 - 1.6.4 Arquitectura de múltiples núcleos (Taxonomía de Flynn)

UNIDAD II. Aritmética binaria y representación de datos.

Competencia:

Representar información en distintos sistemas numéricos, mediante la aplicación de procedimientos de codificación y conversión entre diferentes bases numéricas, para el tratamiento de información y la resolución de problemas aritméticos, mostrando compromiso y puntualidad para la realización de las tareas.

Contenido:

Duración: 4 horas

2.1 Sistemas numéricos.

2.1.1 Sistema binario

2.1.2 Sistema hexadecimal

2.1.3 Conversión entre bases.

2.2 Aritmética binaria

2.2.1 Representación de números con signo.

2.2.1.1 Complemento a 2

2.2.2 Representación de punto flotante

2.2.3 Suma

2.2.4 Resta

2.3 Codificación

2.3.1 Código BCD

2.3.2 Código ASCII

UNIDAD III. Introducción al lenguaje ensamblador.

Competencia:

Emplear el proceso de ensamblaje y enlazamiento, mediante la programación en lenguaje ensamblador, para generar un programa ejecutable a partir de un lenguaje de bajo nivel, con actitud crítica y responsable.

Contenido:**Duración:** 2 horas

3.1 Conceptos

3.1.1 Máquina multinivel

3.1.2 Lenguaje de bajo nivel y lenguaje de alto nivel

3.1.3 Lenguaje ensamblador

3.1.4 Estructura básica de un programa en lenguaje ensamblador

3.2 Generación de programa ejecutable

3.2.1 Proceso de ensamblaje

3.2.2 Proceso de enlazamiento.

UNIDAD IV. Recursos del procesador

Competencia:

Identificar registros, modos de direccionamiento instrucciones básicas e interrupciones para comprender el funcionamiento interno de una computadora, con actitud de responsabilidad y creatividad.

Contenido:

Duración: 8 horas

- 4.1 Conjunto de registros.
 - 4.1.1 Registro de contador de programa
 - 4.1.2 Registros de propósito general.
 - 4.1.3 Registros para manejo de memoria.
 - 4.1.3.1 Registro de apuntador de pila
 - 4.1.4 Registros de bandera.
- 4.2 Modos de direccionamiento.
 - 4.2.1 Direccionamiento a Registro.
 - 4.2.2 Direccionamiento Inmediato.
 - 4.2.3 Direccionamiento Directo.
 - 4.2.4 Direccionamiento de Registro Indirecto.
 - 4.2.5 Direccionamiento Base-más-Índice.
 - 4.2.6 Direccionamiento Relativo a Registro.
 - 4.2.7 Direccionamiento Relativo a Base-más-Índice.
 - 4.2.8 Otros modos de direccionamiento.
- 4.3 Conjunto de Instrucciones.
 - 4.3.1 Instrucciones de transferencia de datos
 - 4.3.2 Instrucciones aritméticas
 - 4.3.3 Instrucciones lógicas y de manipulación de bits
 - 4.3.4 Instrucciones para control del programa
- 4.4 Interrupciones

UNIDAD V. Generalidades del Lenguaje Ensamblador

Competencia:

Identificar las capacidades de un procesador, mediante el análisis de su organización y características de software, para su aplicación en el desarrollo de sistemas, de forma organizada y eficiente.

Contenido:

Duración: 4 horas

- 5.1 Estructura general de un programa de lenguaje ensamblador.
 - 5.1.1 Estructura de un programa ejecutable.
 - 5.1.2 Proceso de depuración (debug)
- 5.2 Tipos de datos del lenguaje ensamblador
- 5.3 Variables
- 5.4 Referencias
- 5.5 Arreglos
- 5.6 Estructuras
- 5.7 Definición de nuevos tipos de datos
- 5.8 Macros
- 5.9 Estructuras de control de alto nivel en lenguaje ensamblador

UNIDAD VI. Entrada y salida de datos

Competencia:

Desarrollar funciones para el manejo de los recursos del sistema, a través de la selección de los componentes y aplicación de mecanismos de manejo de puertos adecuados, para su uso en interfaces de comunicación con dispositivos de E/S, de forma analítica, organizada y eficiente.

Contenido:

6.1 Acceso a Recursos del Sistema

6.2 Sistema Básico de Entrada y Salida

6.2.1 Acceso a dispositivos mediante los recursos del equipo

6.2.2 Acceso a dispositivos mediante el sistema operativo

6.2.3 Acceso a dispositivos de forma directa

Duración: 4 horas

UNIDAD VII. Interfaz Lenguaje de Alto Nivel con Lenguaje Ensamblador

Competencia:

Integrar subrutinas en lenguaje ensamblador a programas de alto nivel, aplicando buenas prácticas en cuanto al uso de convenciones, para el desarrollo de interfaces entre sistemas, de manera responsable y creativa.

Contenido:

- 7.1 Uso de las instrucciones en línea
- 7.2 Llamado a funciones de ensamblador desde el lenguaje de alto nivel
- 7.3 Convención de nombres
- 7.4 Convención de paso de parámetros
- 7.6 Convención de retorno de valores
- 7.7 Uso de registros
- 7.8 Ejemplos

Duración: 6 horas

VI. ESTRUCTURA DE LAS PRÁCTICAS DE LABORATORIO

No.	Nombre de la Práctica	Procedimiento	Recursos de Apoyo	Duración
UNIDAD I				
1	Elementos en la organización de una computadora de propósito general.	<ol style="list-style-type: none"> 1. El docente proporciona una explicación del uso del simulador y del modelo a analizar. 2. Ejecutar la simulación para identificar los elementos en la organización de una computadora de propósito general. 3. Entregar reporte correspondiente con las siguientes características: búsqueda bibliográfica, resultados, discusión, referencias. 	<ul style="list-style-type: none"> • Equipo de cómputo con simulador instalado. 	2 horas
2	Interconexión de elementos en la organización de una computadora de propósito general.	<ol style="list-style-type: none"> 1. El docente proporciona una explicación del uso del simulador y del modelo a analizar. 2. Ejecutar la simulación para identificar la interconexión en los elementos en la organización de una computadora de propósito general. 3. Entregar reporte correspondiente donde incluya búsqueda bibliográfica, resultados, discusión, referencias. 	<ul style="list-style-type: none"> • Equipo de cómputo con simulador instalado. 	2 horas
UNIDAD II				

3	Sistemas numéricos	<ol style="list-style-type: none"> 1. El docente plantea ejercicios de los diferentes sistemas numéricos para la representación de cantidades. 2. Realizar conversiones entre diferentes bases numéricas. 3. Entregar reporte de las representaciones, donde describa el procedimiento aplicado. 	<ul style="list-style-type: none"> • Documento (físico o electrónico) con ejercicios de conversiones entre sistemas numéricos. • Software de apoyo de conversión de bases, para que el alumno después de realizar el procedimiento de forma manual, pueda verificar sus resultados. 	2 horas
4	Aritmética binaria	<ol style="list-style-type: none"> 1. El docente plantea ejercicios de problemas aritméticos en los diferentes sistemas numéricos para la representación de cantidades. 2. Resolver operaciones de suma y resta en sistema binario. 3. Resolver operaciones de suma y resta con cifras con signo utilizando complemento a 2, signo y magnitud. 4. Verificar que no ocurran errores 5. de desbordamiento en operaciones y propone solución. 6. 5. Entregar reporte de las operaciones, donde describa el procedimiento aplicado. 	<ul style="list-style-type: none"> • Documento (físico o electrónico) con ejercicios de problemas aritméticos en los diferentes sistemas numéricos. • Software de apoyo de operaciones aritméticas binarias, para que el alumno después de realizar el procedimiento de forma manual, pueda verificar sus resultados. 	2 horas
UNIDAD III				
5	Generación de un programa ejecutable	<ol style="list-style-type: none"> 1. El docente especifica el nombre de las herramientas que se utilizan para elaborar programas en ensamblador e identificar los sitios Web en los que se podrá encontrar información de apoyo. 2. Realizar un programa ejemplo de bajo nivel que hace uso de los recursos básicos de entrada 	<ul style="list-style-type: none"> • Equipo de cómputo con acceso a internet. • Programa para realizar el programa ejecutable en lenguaje ensamblador. 	2 horas

		(teclado) y salida (pantalla) del sistema. 3. Entregar reporte correspondiente, que describa lo realizado, resultados obtenidos, discusión y conclusiones		
UNIDAD IV				
6	Modos de Direcccionamiento	<ol style="list-style-type: none"> 1. El docente plantea que basado en programa ejemplo desarrollar un programa básico de bajo nivel que haga uso de los modos de direccionamiento del microprocesador. 2. Realizar un programa que utilice los direccionamientos. 3. Entregar reporte que describa lo realizado, resultados obtenidos, discusión y conclusiones. 	<ul style="list-style-type: none"> • Equipo de cómputo. • Software para realizar el programa de bajo nivel, que usa los direccionamientos. 	2 horas
7	Instrucciones aritméticas, lógicas y de manipulación de bits	<ol style="list-style-type: none"> 1. El docente plantea que Basado en programa ejemplo desarrollar un programa básico de bajo nivel que haga uso de las instrucciones aritméticas, lógicas y de manipulación de bits del microprocesador. 2. Realizar el programa que contenga instrucciones aritméticas, lógicas y manipulación de bits. 3. Entregar reporte que describa lo realizado, resultados obtenidos, discusión y conclusiones 	<ul style="list-style-type: none"> • Equipo de cómputo. • Software para realizar el programa de bajo nivel que emplea las instrucciones aritméticas, lógicas y manipulación de bits. 	2 horas
8	Instrucciones de control de flujo	<ol style="list-style-type: none"> 1. El docente plantea que basado en programa ejemplo desarrollar un programa básico de bajo nivel que haga uso de las 	<ul style="list-style-type: none"> • Equipo de cómputo. • Software para realizar el programa de bajo nivel que 	2 horas

		<p>instrucciones de control de flujo del programa del microprocesador.</p> <p>2. Realizar el programa que utilice las instrucciones de control de flujo.</p> <p>3. Entregar reporte que describa lo realizado, resultados obtenidos, discusión y conclusiones..</p>	<p>utiliza las instrucciones de control de flujo.</p>	
9	Interrupciones	<p>1. El docente plantea que basado en programa ejemplo desarrollar un programa de bajo nivel que haga uso del esquema de interrupciones para llamadas de servicio del sistema operativo.</p> <p>2. Realizar el programa que utilice las interrupciones.</p> <p>3. Entregar reporte que describa lo realizado, resultados obtenidos, discusión y conclusiones.</p>	<ul style="list-style-type: none"> • Equipo de cómputo. • Software para realizar el programa de bajo nivel que utiliza las interrupciones. 	2 horas
UNIDAD V				
10	Depuración	<p>1. El docente explica cómo refinar un programa con ayuda de un depurador (debug).</p> <p>2. Identificar las partes del depurador, así como el significado de los resultados del depurador.</p> <p>3. Entregar reporte correspondiente que describa lo realizado, resultados obtenidos, discusión y conclusiones</p>	<ul style="list-style-type: none"> • Equipo de cómputo con un depurador (debug) instalado. • Programa ejemplo a depurar. 	2 horas
11	Macros y Lenguaje Ensamblador de Alto Nivel	<p>1. El docente plantea basado en programa ejemplo desarrollar un programa de contenga Macros.</p>	<ul style="list-style-type: none"> • Equipo de cómputo. • Software para realizar el programa de bajo nivel que contiene Macros. 	2 horas

		<ol style="list-style-type: none"> Realiza el programa que contiene macros. El docente explica el Lenguaje Ensamblador de Alto Nivel y explica un programa ejemplo. Realiza un programa en el Lenguaje Ensamblador de Alto Nivel. Entregar reporte correspondiente que describa lo realizado, resultados obtenidos, discusión y conclusiones. 	<ul style="list-style-type: none"> Software de Lenguaje Ensamblador de Alto Nivel. 	
UNIDAD VI				
12	Sección de Entrada/Salida	<ol style="list-style-type: none"> El docente plantea basado en programa ejemplo desarrollar un programa que sea capaz de manejar entradas y salidas de datos. Realiza el programa que maneja entradas y salidas de datos. Entregar reporte correspondiente que describa lo realizado, resultados obtenidos, discusión y conclusiones. 	<ul style="list-style-type: none"> Equipo de cómputo. Software para realizar el programa de bajo nivel que maneja entradas y salidas de datos. 	4 horas
UNIDAD VII				
13	Subrutinas de ensamblador	<ol style="list-style-type: none"> El docente plantea basado en un programa ejemplo desarrollar subrutinas en lenguaje de bajo nivel que serán utilizadas por un programa de alto nivel. Realizar las subrutinas con el lenguaje de bajo nivel. Entregar reporte correspondiente que describa lo realizado, resultados obtenidos, discusión y conclusiones. 	<ul style="list-style-type: none"> Equipo de cómputo. Software para realizar subrutinas de bajo nivel. 	2 horas

14	Acceso a subrutinas de ensamblador por programas de alto nivel	<ol style="list-style-type: none"> 1. El docente plantea basado en un programa ejemplo desarrollar un programa de alto nivel que utilice subrutinas de lenguaje de bajo nivel. 2. Realizar el programa que utilice las subrutinas de bajo nivel. 3. Entregar reporte correspondiente que describa lo realizado, resultados obtenidos, discusión y conclusiones. 	<ul style="list-style-type: none"> • Equipo de cómputo. • Software para realizar subrutinas de bajo nivel. • Software para realizar programas de alto nivel. 	4 horas
----	---	--	---	---------

VII. MÉTODO DE TRABAJO

Encuadre: El primer día de clase el docente debe establecer la forma de trabajo, criterios de evaluación, calidad de los trabajos académicos, derechos y obligaciones docente-alumno.

Estrategia de enseñanza (docente): Desarrollar sesiones para la presentación de la información teórica, mediante el método expositivo con el apoyo de equipo audiovisual, facilitar material bibliográfico introductorio para la comprensión de conceptos y el cuerpo de conocimiento actual de un tema, coordinar discusión dirigida en temas específicos para promover el trabajo colaborativo, pensamiento crítico y reflexivo, asesorar de forma personalizada para el análisis, diseño, construcción y prueba de programas en lenguaje de bajo y alto nivel, coordinar y supervisar las prácticas de laboratorio, elaborar y aplicar las evaluaciones parciales.

Estrategia de aprendizaje (alumno):

Participar activamente en clase en actividades individuales y grupales, seleccionar, organizar y comprender la información, generar un análisis, diseño, construcción y prueba de programas en lenguaje de bajo y alto nivel, emplear el aprendizaje auto-dirigido.

VIII. CRITERIOS DE EVALUACIÓN

La evaluación será llevada a cabo de forma permanente durante el desarrollo de la unidad de aprendizaje de la siguiente manera:

Criterios de acreditación

- Para tener derecho a examen ordinario y extraordinario, el estudiante debe cumplir con los porcentajes de asistencia que establece el Estatuto Escolar vigente.
- Calificación en escala del 0 al 100, con un mínimo aprobatorio de 60.

Criterios de evaluación

- Evaluaciones parciales (3).....40%
- Prácticas de Laboratorio.....40%
- Tareas.....10%
- Evidencia de desempeño 10%
(Portafolio de prácticas de laboratorio y repositorio de software desarrollado)

Total.....100%

IX. REFERENCIAS

Básicas	Complementarias
<p>Brey, B. (2009). <i>The Intel microprocessors: 8086/8088, 80186/80188, 80286, 80386, 80486, Pentium, Pentium Pro processor, Pentium II, Pentium III, Pentium 4, and Core2 with 64-bit extensions: architecture, programming, and interfacing</i>. Nueva Jersey, Estados Unidos: Pearson Prentice Hall. [clásica]</p> <p>Eckhouse R. & Morris, R. (S/F). <i>Minicomputer Systems. Organization, Programming and Applications (PDP-11)</i>".</p> <p>Hayes, J. (1998). <i>Computer Architecture and Organization</i>. Edison</p> <p>Null, L. y Lobur, J. (2019). <i>The essentials of computer organization and architecture</i>. Massachusetts, Estados Unidos: Jones y Bartlett Learning.</p> <p>Patterson, D., & Hennessy, J. (1994). <i>Computer Organization & Design, The Hardware/Software Interface (2nd ed.)</i>, Morgan Kaufmann.</p> <p>Stallings, W. (2007) <i>Organización y arquitectura de computadores. Séptima Edición</i>. España: Pearson Educación. [clásica]</p> <p>Stallings, W. (2016). <i>Computer organization and architecture: designing for performance</i>. Massachusetts, Estados Unidos: Pearson-Prentice Hall.</p> <p>Tanenbaum, A. y Austin, T. (2013). <i>Structured computer organization</i>. Massachusetts, Estados Unidos: Pearson. [clásica]</p> <p>Tanenbaum. A. (2005). <i>Structured Computer Organization (5th ed.)</i>, Prentice Hall. [Clásica]</p> <p>William Stallings. (2012). <i>Computer Organization and Architecture (9th edition)</i>. Prentice Hall.</p> <p>V.C. Manacher, Z.G. Vranesic & S.G. Zaky. (2001). <i>Computer Organization" (5th ed.)</i>, McGraw-Hill.</p>	<p>Barron, D.W. (1979). <i>Assemblers and Loaders</i>. Elsevier.</p> <p>Gear, W.(1980). <i>Computer Organization and Programming</i>. McGraw-Hill. [clásica]</p> <p>Hyde R. (2011). <i>High Level Assembly</i>. San Francisco, Estados Unidos: No Starch Press. [clásica]</p> <p>Kernigham, B. & Ritchie, D. (1998). <i>The C Programming Language (2nd ed.)</i>. PrenticeHall. [clásico]</p> <p>Leiterman J. (2005). <i>32/64-Bit 80x86 Assembly Language Architecture</i>. Estados Unidos: Wordware Publishing. [clásica]</p> <p>Perry, G. (1994). <i>Absolute Beginner's Guide to C"</i> (2nd ed.).</p> <p>Prinz, P. & Crawford, T. (2005). <i>C in a Nutshell</i>. O'Reilly.</p>

X. PERFIL DEL DOCENTE

El docente que imparta esta asignatura deberá contar con una formación en ciencias o Ingeniería en software o computación, o bien, una carrera afín, Es deseable contar con experiencia profesional comprobable en el área de sistemas digitales basados en microprocesadores y el manejo de los lenguajes ensamblador y - de alto nivel, así como haber acreditado cursos de formación docente y capacitación en la enseñanza y evaluación por competencias. Para el desarrollo de la actividad docente en esta asignatura es necesario contar con la capacidad para interpretar información técnica en inglés

Habilidades : Se requiere cuente con la habilidad de comunicación efectiva y liderazgo para propiciar el trabajo en equipo. Adicionalmente, ser una persona proactiva, analítica, responsable, con un alto sentido de la ética y con vocación de servicio a la enseñanza.

Actitudes y Valores : Propositiva. Honestidad. Empatía. Perseverancia. Compromiso. Tolerancia. Flexibilidad. Responsabilidad. Proactiva. Humildad. Lealtad. Eficiencia. Respeto. Confianza.