



1.- Identificación de la Unidad de Aprendizaje					
Nombre de la Unidad de Aprendizaje					
<b>Fundamentos de las Ciencias Computacionales</b>					
Clave de la UA	Modalidad de la UA	Tipo de UA		Valor de créditos	Área de formación
IG737	Presencial	Curso-Taller		6	Básica Particular Obligatoria
Hora semana		Horas teoría/semestre	Horas práctica/semestre	Total de horas:	Seriación
3		40	20	60	No tiene
Departamento			Academia		
<i>Ciencias de la Información y Desarrollos Tecnológicos</i>			Ingeniería de Software		
Presentación					
En esta unidad de aprendizaje el alumno comprenderá el funcionamiento básico de un sistema computacional mediante el análisis de cada uno de sus componentes, así como la revisión de los antecedentes históricos a partir de los cuales operan los sistemas modernos.					
Tipos de saberes					
Saber (Conocimientos)		Saber hacer (Habilidades)		Saber ser (Actitudes y valores)	
Partes de la computadora, nombres y tipos de los dispositivos, modelo de von Newman. Sistemas de numeración, cambio de base, aritmética modular, codificación y formatos de archivos. Estructura de un programa en memoria, fases de la compilación y ejecución, características de los lenguajes de alto y bajo nivel. Características del lenguaje ensamblador, partes de los CPU's modernos, características del sistema operativo. Licencias de software propietario, software gratuito y software libre; comandos básicos de Linux.		Identificar sistemas que cumplan con el modelo de von Newman. Operaciones aritméticas en diferentes sistemas de numeración, obtener el módulo de un número. Codificar, compilar y ejecutar un programa sencillo. Codificar, compilar y ejecutar un programa rudimentario en lenguaje ensamblador. Observar la ejecución de un programa en el CPU. Instalar una máquina virtual. Instalar un sistema operativo.		(Patrones de comportamiento que caracterizan el actuar del individuo, íntimamente ligados con los valores y las creencias. Principios que rigen los comportamientos, formas de pensar y de ser, son los patrones de significación mas profundos de los sujetos.)	
Competencia genérica			Competencias profesionales		
Distingue las arquitecturas de cómputo existentes para conocer la aplicación de cada una en problemas específicos.			<ul style="list-style-type: none"> <li>• Memoriza el concepto de computadora para posteriormente referirlo en el contexto de las ciencias computacionales.</li> <li>• Explica el funcionamiento del lenguaje ensamblador para mostrar su ejecución en los registros físicos del CPU.</li> <li>• Señala el ciclo de arranque del sistema operativo para identificar problemas asociados con el mismo.</li> </ul>		



	<ul style="list-style-type: none"><li>• Identifica las diferentes etapas de la programación para seguirlas en la práctica diaria de la programación.</li><li>• Relaciona las fases de la ejecución de un programa para identificar problemas en la ejecución de los mismos.</li><li>• Argumenta las características de los lenguajes de programación para clasificarlos en alto y bajo nivel.</li><li>• Explica el funcionamiento del lenguaje ensamblador para mostrar su ejecución en los registros físicos del CPU.</li><li>• Señala el ciclo de arranque del sistema operativo para identificar problemas asociados con el mismo.</li><li>• Aplica los conocimientos adquiridos para instalar un sistema operativo en una máquina virtual.</li><li>• Sintetiza las diferentes licencias de software para enumerar sus restricciones de manera general.</li></ul>
<b>Saberes previos del alumno</b>	
Comprensión oral y escrita, aritmética básica.	
<b>Perfil de egreso al que se abona</b>	
Utiliza modelos matemáticos, aplica algoritmos y principios de las ciencias computacionales en el modelado y diseño de sistemas de información.	
<b>Perfil deseable del docente</b>	
Conocimientos: <ul style="list-style-type: none"><li>• Arquitecturas de computadoras</li><li>• Aritmética modular</li><li>• Sistemas de numeración</li><li>• Programación en lenguajes de bajo nivel</li></ul>	
Habilidades: <ul style="list-style-type: none"><li>• Buena comunicación oral y escrita</li><li>• Planificar el proceso de enseñanza</li><li>• Interactuar de manera empática con los alumnos</li></ul>	



2.- Contenidos temáticos	
Contenido	
<ol style="list-style-type: none"><li>1. Arquitecturas de Computadoras<ol style="list-style-type: none"><li>1.1. Conceptos básicos</li><li>1.2. Computabilidad</li><li>1.3. El modelo de von Newman</li><li>1.4. Unidad central de procesamiento</li><li>1.5. Tipos de memoria</li><li>1.6. Subsistemas de entrada y salida</li><li>1.7. Dispositivos de almacenamiento</li></ol></li><li>2. Representación de datos<ol style="list-style-type: none"><li>2.1. Tipos de datos</li><li>2.2. Representación interna (bites y bytes)</li><li>2.3. Códigos</li><li>2.4. Representación de textos</li><li>2.5. Representación de imágenes</li><li>2.6. Representación de sonidos</li></ol></li><li>3. Características del software de sistema<ol style="list-style-type: none"><li>3.1. Programación de bajo nivel</li><li>3.2. Descripción funcional de la memoria</li><li>3.3. Lenguajes máquina y simbólico</li><li>3.4. Programas fuente</li><li>3.5. Código objeto</li><li>3.6. Mapa de memoria</li><li>3.7. Ciclo de ejecución del CPU</li><li>3.8. Tipos de direccionamiento de memoria</li></ol></li><li>4. Software de sistema<ol style="list-style-type: none"><li>4.1. Ensambladores</li><li>4.2. Macroprocesadores</li><li>4.3. Cargadores</li><li>4.4. Compiladores e intérpretes</li><li>4.5. Sistema operativo</li></ol></li><li>5. Software libre<ol style="list-style-type: none"><li>5.1. Antecedentes</li><li>5.2. Comandos básicos</li><li>5.3. Entorno gráfico</li><li>5.4. Aplicaciones más conocidas</li><li>5.5. Licencias de software</li></ol></li></ol>	
<b>Estrategias generales para impartir la unidad de aprendizaje</b>	
Disertación tradicional, material audiovisual, material de apoyo en línea, videos en línea, emuladores de ensamblador en línea, uso de maquina virtual.	
<b>Módulo I</b>	
<b>Arquitecturas de Computadoras</b>	
<b>Competencia Específica</b>	
<ul style="list-style-type: none"><li>• Memoriza el concepto de computadora para posteriormente referirlo en el contexto de las ciencias computacionales.</li><li>• Distingue las arquitecturas de cómputo existentes para conocer la aplicación de cada una en problemas específicos.</li></ul>	
<b>Tipos de saberes</b>	



Saber (Conocimientos)	Saber hacer (Habilidades)	Saber ser (Actitudes y valores)
Partes de la computadora, nombres y tipos de los dispositivos, modelo de von Newman.	Identificar sistemas que cumplan con el modelo de von Newman.	
<b>Módulo II</b>		
<b>Representación de datos</b>		
<b>Competencia Específica</b>		
<ul style="list-style-type: none"> <li>• Aplica los cambios de base para realizar operaciones aritméticas en los diferentes sistemas de representación.</li> <li>• Identifica los diferentes formatos de archivos para asociar el correcto según la información a representar.</li> </ul>		
<b>Tipos de saberes</b>		
Saber (Conocimientos)	Saber hacer (Habilidades)	Saber ser (Actitudes y valores)
Sistemas de numeración, cambio de base, aritmética modular, codificación y formatos de archivos.	Operaciones aritméticas en diferentes sistemas de numeración, obtener el módulo de un número.	
<b>Módulo III</b>		
<b>Características del software de sistema</b>		
<b>Competencia Específica</b>		
<ul style="list-style-type: none"> <li>• Identifica las diferentes etapas de la programación para seguirlas en la práctica diaria de la programación.</li> <li>• Relaciona las fases de la ejecución de un programa para identificar problemas en la ejecución de los mismos.</li> <li>• Argumenta las características de los lenguajes de programación para clasificarlos en alto y bajo nivel.</li> </ul>		
<b>Tipos de saberes</b>		
Saber (Conocimientos)	Saber hacer (Habilidades)	Saber ser (Actitudes y valores)
Estructura de un programa en memoria, fases de la compilación y ejecución, características de los lenguajes de alto y bajo nivel.	Codificar, compilar y ejecutar un programa sencillo.	
<b>Módulo IV</b>		
<b>Software de sistema</b>		
<b>Competencia Específica</b>		
<ul style="list-style-type: none"> <li>• Explica el funcionamiento del lenguaje ensamblador para mostrar su ejecución en los registros físicos del CPU.</li> <li>• Señala el ciclo de arranque del sistema operativo para identificar problemas asociados con el mismo.</li> </ul>		
<b>Tipos de saberes</b>		
Saber (Conocimientos)	Saber hacer (Habilidades)	Saber ser (Actitudes y valores)
Características del lenguaje ensamblador, partes de los CPU's modernos, características del sistema operativo.	Codificar, compilar y ejecutar un programa rudimentario en lenguaje ensamblador. Observar la ejecución de un programa en el CPU.	
<b>Módulo V</b>		
<b>Software libre</b>		
<b>Competencia Específica</b>		



- Aplica los conocimientos adquiridos para instalar un sistema operativo en una máquina virtual.
- Sintetiza las diferentes licencias de software para enumerar sus restricciones de manera general.

#### Tipos de saberes

Saber (Conocimientos)	Saber hacer (Habilidades)	Saber ser (Actitudes y valores)
Licencias de software propietario, software gratuito y software libre; comandos básicos de Linux.	Instalar una máquina virtual. Instalar un sistema operativo.	

#### Bibliografía básica

- Tanenbaum, A. y Austin, T. (2013). Organización de computadoras, enfoque estructurado. Prentice Hall. Estados Unidos, 2013.
- Stallings, W. (2013). Sistemas operativos: aspectos internos y principios de diseño. Pearson Education. Estados Unidos, 2013.

#### Bibliografía complementaria

- Wentzlaff, David (2018). Computer Architecture. Curso en línea de la Universidad de Princeton – Coursera. <https://www.coursera.org/learn/comparch>

### 3.-Evaluación

#### Criterios de Evaluación (% por criterio)

Exámenes	60%
Tareas	20%
Proyecto	20%

### 4.-Acreditación

### 5.- Participantes en la elaboración

Código	Nombre
2944934	César Alejandro García García

### 6.- Fecha de elaboración

**Elaboración: Noviembre 2018**

**Aprobado en Academia:**

**Aprobado en Colegio Departamental:**

**Próxima revisión:**