



Abril 04, 2014.


**M. EN C. BLANCA R. SILVA LÓPEZ**  
Jefa del Departamento de Sistemas  
P r e s e n t e.

Por este medio le envío copia del *Programa Analítico* aprobado y debidamente firmado por el Departamento y el Director de esta División y que a continuación le detallo:

CLAVE	UEA
1151038	Programación Estructurada

Sin más por el momento, agradezco a usted de antemano su atención a la presente.

A T E N T A M E N T E  
"CASA ABIERTA AL TIEMPO"



**DRA. MA. DE LOURDES DELGADO NÚÑEZ**  
Secretaria Académica de la división de  
Ciencias Básicas e Ingeniería.

\*vrg.

Fecha de aprobación:  
**Marzo 28, 2014.**

Departamento de

## PROGRAMA ANALÍTICO

Nivel : Licenciatura		Unidad de enseñanza-aprendizaje	
Clave: 1151038		PROGRAMACIÓN ESTRUCTURADA	
Horas teoría 2.5	Horas práctica 2.0	Seriación 1112013 Y 1112027	Créditos 7

L i c e n c i a t u r a  e n	I n g e n i e r í a	A m b i e n t a l	C i v i l	E n C o m p u t a c i o n	E l é c t r i c a	E l e c t r ó n i c a	F í s i c a	I n d u s t r i a l	M e c á n i c a	M e t a l ú r g i c a	Q u í m i c a
<b>OBLIGATORIA</b>											
Tronco de Nivelación Académica											
Tronco General		X	X	X	X	X	X	X	X	X	X
Tronco Inter y Multidisciplinar											
Tronco Básico Profesional											
Tronco de Integración											
<b>OPTATIVA</b>											
Tronco Inter y Multidisciplinar											
Tronco de Integración											
Otras Optativas											
<b>TRIMESTRE</b>											
<b>Observaciones</b>											

### OBJETIVO(S) GENERALES:

Al final de la UEA el alumno será capaz de:

- Describir los conceptos de algoritmo y de programa.
- Explicar el paradigma de programación estructurada.
- Explicar, elaborar y representar algoritmos.
- Implementar programas escritos en lenguaje C, usando el paradigma de programación estructurada.
- Desarrollar los programas usando el ambiente Unix.

### CONTENIDO SINTÉTICO:

1. Algoritmos y programas.
2. Diseño de programas estructurados.
3. Ambiente de desarrollo Unix.
4. Programación modular.
5. Elementos básicos de un programa en lenguaje C.
6. Estructuras de decisión.
7. Estructuras de repetición.
8. Arreglos y estructuras.
9. Cadenas de caracteres.
10. Archivos.

## TEMA 1. Algoritmos y programas

### OBJETIVOS ESPECÍFICOS:

1. Definir el concepto de algoritmo.
2. Entender los conceptos de programa y lenguaje de programación

### CONTENIDO:

1. Definición y características de algoritmo.
2. Programas y lenguajes de programación.
  - a. Definición de programa.
  - b. Lenguajes de alto nivel y el proceso de compilación.
  - c. Independencia de algoritmo respecto al lenguaje de programación

### REFERENCIAS:

Bronson G.J., C++ para ingeniería y ciencias, México: International Thomson Editores, 2000. Capítulo 1.

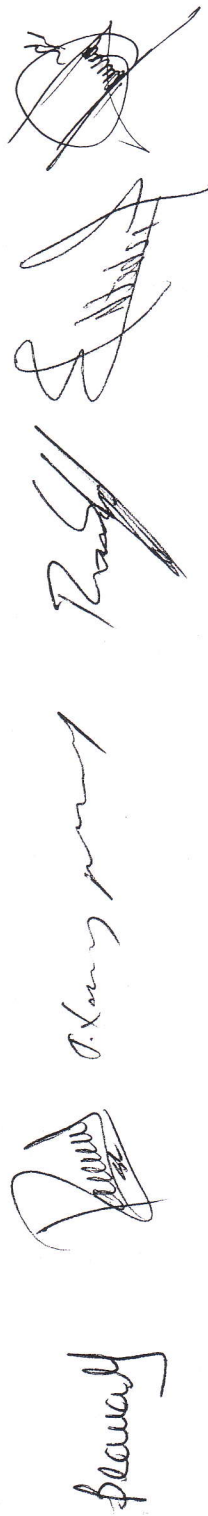
Cairó O., Fundamentos de programación. Piensa en C. México: Pearson Education, 2006. Capítulo 1.

Ceballos F.J., C/C++ Curso de Programación, 3ra. Edición, México: RA-MA Editorial, 2007. Capítulo 1

### HORAS DE CLASE:

1.5

### OBSERVACIONES:



Five handwritten signatures in blue ink, arranged vertically on the right side of the page. The signatures are: 1. A signature that appears to be 'Rosa'. 2. A signature that appears to be 'Eduardo'. 3. A signature that appears to be 'Rosa'. 4. A signature that appears to be 'A. X. ...'. 5. A signature that appears to be 'Pascual'.





## TEMA 3. Ambiente de desarrollo Unix

### OBJETIVOS ESPECÍFICOS:

1. Entender los conceptos de archivo y directorio del ambiente UNIX.
2. Manipular directorios y archivos desde la consola de comandos.
3. Ejecutar aplicaciones desde la consola de comandos.

### CONTENIDO:

1. Archivos y directorios en UNIX.
  - a. Concepto de archivo en UNIX.
  - b. Tipos de archivos y directorios como contenedores de archivos.
  - c. Distinción entre nombre y una ruta de un archivo o directorio.
2. La consola de comandos
  - a. Los comandos `ls` y `cd`.
  - b. Los comandos `cat` y `mkdir`.
  - c. Los comandos `rm` y `rmdir`.
  - d. Los comandos `cp` y `mv`.
3. Aplicaciones y programas en UNIX.
  - a. Búsqueda de programas invocados y la carpeta `bin`.
  - b. Ejecución de programas de otras carpetas.
  - c. Invocación de programas con argumentos de línea de comandos.

### REFERENCIAS:

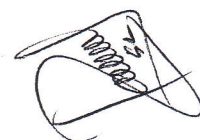
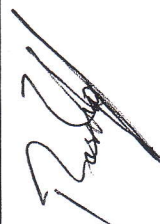
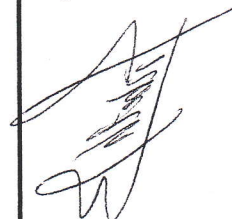
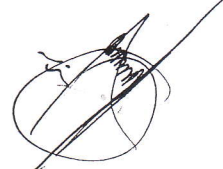
Kernighan B.W. y Pike R.,  
El entorno de programación  
UNIX, México: renticeHall  
Hispanoamericana,  
1987. Capítulos 1, 2 y 3.

Oram y Loukides M.,  
Programming with GNU  
software, EUA: O'Really  
Media, 1996. Capítulos  
1,2,3 y 4

### HORAS DE CLASE:

3.0

### OBSERVACIONES:



## TEMA 4. Programación modular

### OBJETIVOS ESPECÍFICOS:

1. Definir el concepto de función.
2. Identificar los tipos de funciones.
3. Construir programas empleando funciones

### CONTENIDO:

1. Concepto de función:
  - a. Declaración de una función.
  - b. Definición de una función.
  - c. Llamado de una función.
  - d. La función como unidad de reutilización.
2. Identificar los tipos de funciones:
  - a. Funciones con paso de parámetros por valor
  - b. Funciones con paso de parámetros por referencia.
3. Elaboración de programas que involucren la creación de funciones:
  - a. Manejo de apuntadores.
  - b. Paso de parámetros por valor.
  - c. Paso de parámetros por referencia.

### REFERENCIAS:

Bronson G.J., C++ para ingeniería y ciencias, México: International Thomson Editores, 2000. Capítulo 6.

Ceballos F.J., C/C++ Curso de Programación, 3ra. Edición, México: RA-MA Editorial, 2007. Capítulo 3.

Savitch W., Resolución de problemas con C++, México: Pearson Education, 2007. Capítulo 4.


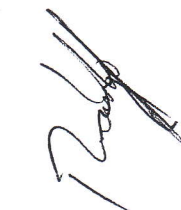
### HORAS DE CLASE:

7.5

### OBSERVACIONES:

Se recomienda proponer algoritmos que resuelvan problemas como:

- Cálculo de las funciones sucesor y predecesor.
- Intercambio de variables.
- Cálculo de la función redondea.
- Representar una cantidad de segundos en términos de horas-minutos-segundos



## TEMA 5. Elementos básicos de un programa en lenguaje C

### OBJETIVOS ESPECÍFICOS:

1. Conocer los tipos de datos básicos de C.
2. Definir variables, constantes y funciones en C.
3. Conocer las operaciones aritméticas, relacionales y lógicas disponibles en C.
4. Describir la biblioteca estándar de C y su mecanismo de inclusión.

### CONTENIDO:

1. Tipos de dato básicos del lenguaje C.
  - a. Tipos enteros.
  - b. Tipos coma flotante.
  - c. Tipos caracteres.
2. Definición de variables y funciones.
  - a. Definición de variables, inicialización y asignación.
  - b. Definición de funciones.
  - c. Definición de constantes.
3. Operaciones aritméticas relacionales y lógicas de C.
  - a. Aritméticas: Suma, resta, multiplicación, división y residuo.
  - b. Relacionales:  $>$ ,  $<$ ,  $>=$ ,  $<=$ ,  $==$ ,  $!=$ .
  - c. Lógicas: `and`, `or`, `not`.
  - d. Precedencia de operadores.
  - e. Evaluación de expresiones con tipos de datos heterogéneos.
4. Biblioteca estándar del lenguaje C.
  - a. La instrucción `include`.
  - b. Organización y división de la biblioteca estándar.
5. Entrada y salida de la biblioteca estándar.
  - a. La función `scanf` y entrada con formato de tipos básicos.
  - b. La función `printf` y salida con formato de tipos básicos.
6. Elaboración de programas que involucren el uso de funciones matemáticas y la evaluación de expresiones con tipos de datos heterogéneos

### REFERENCIAS:

Ceballos F.J., C/C++ Curso de Programación, 3ra. Edición, México: RA-MA Editorial, 2007. Capítulo 2.

Cairó O., Fundamentos de programación. Piensa en C. México: Pearson Education, 2006. Capítulo 1.

### HORAS DE CLASE:

4.5

### OBSERVACIONES:

Se recomienda proponer algoritmos que resuelvan problemas como:

- Conversiones entre unidades de medición.
- Conversiones entre unidades de temperatura.
- Cálculo de funciones trigonométricas.

Los algoritmos podrán describirse mediante pseudocódigo o alguna otra técnica.



## TEMA 6. Estructuras de decisión

### OBJETIVOS ESPECÍFICOS:

1. Construir programas usando la sentencia `if`.
2. Construir programas usando la sentencia `switch`.

### CONTENIDO:

1. La sentencia `if`:
  - a. Bifurcación del flujo del programa usando la sentencia `if`.
  - b. La sentencia `else` como rama opcional de bifurcación.
  - c. Bifurcaciones encadenadas y construcción `else-if`.
2. La sentencia `switch`.
  - a. División multicamino del flujo del programa usando la sentencia `switch`.
  - b. Las sentencias `case` y `default`.
  - c. Comportamiento en cascada de la sentencia `switch`.
  - d. Comportamiento de la sentencia `break` dentro de una sentencia `switch`.
3. Elaboración de programas que involucren estructuras de decisión.

### REFERENCIAS:

Bronson G.J., C++ para ingeniería y ciencias, México: International Thomson Editores, 2000. Capítulo 4.

Kernighan B. W. y Pike R., El entorno de programación UNIX, México: Prentice-Hall Hispanoamericana, 1987. Capítulo 3.

Ceballos F.J., C/C++ Curso de Programación, 3ra. Edición, México: RA-MA Editorial, 2007. Capítulo 5.

### HORAS DE CLASE:

6.0

### OBSERVACIONES:

Se recomienda proponer algoritmos que resuelvan problemas como:

- Simular el tiro de una moneda.
- Simular el juego de la pirinola.
- Simular el juego de piedra, papel o tijera.

Los algoritmos podrán describirse mediante pseudocódigo o alguna otra técnica.

## TEMA 7. Estructuras de repetición

### OBJETIVOS ESPECÍFICOS:

1. Entender los conceptos de ciclo e iteración.
2. Construir programas iterativos utilizando las diferentes sentencias iterativas de C.
3. Controlar la ejecución del ciclo o iteración usando las sentencias `break` y `continue`.

### CONTENIDO:

1. Ciclos e iteraciones:
  - a. Estado del programa durante un proceso iterativo.
  - b. Ciclos como grupos de iteraciones.
2. Sentencias iterativas de C:
  - a. El ciclo `while` como sentencia iterativa condicionada en la parte superior del bloque.
  - b. El ciclo `do` como sentencia iterativa en la parte inferior del bloque.
  - c. El ciclo `for` como simplificación sintáctica para ciclos contadores.
3. Sentencias `break` y `continue`:
  - a. Ejemplos de ciclos condicionados en la parte interna del bloque y la sentencia `break`.
  - b. Comportamiento de la sentencia `continue` dentro de un ciclo.
4. Elaboración de programas que involucren estructuras de repetición.

### REFERENCIAS:

Bronson G.J., C++ para ingeniería y ciencias, México: International Thomson Editores, 2000. Capítulo 5.

Kernighan B.W. y Pike R., El entorno de programación UNIX, México: Prentice-Hall Hispanoamericana, 1987. Capítulo 3.

Ceballos F.J., C/C++ Curso de Programación, 3ra. Edición, México: RA-MA Editorial, 2007. Capítulo 5.

### HORAS DE CLASE:

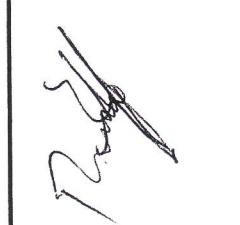
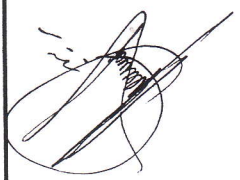
7.5

### OBSERVACIONES:

Se recomienda proponer algoritmos que resuelvan problemas como:

- Obtener los números primos existentes en un rango de valores.
- Obtener los números perfectos en un rango de valores
- Obtener una sucesión de valores fibonacci.
- Series matemáticas como: Taylor, Fourier y trigonométricas.

Los algoritmos podrán describirse mediante pseudocódigo o alguna otra técnica.



Parcial

## TEMA 8. Arreglos y estructuras

### OBJETIVOS ESPECÍFICOS:

1. Entender el concepto de arreglo y su relación con el concepto de apuntador.
2. Entender el concepto de estructura en C.
3. Construir programas que utilicen arreglos.
4. Construir programas que utilicen estructuras.

### CONTENIDO:

1. Arreglos:
  - a. Arreglos como colecciones homogéneas de variables enumeradas.
  - b. Acceso a elementos de un arreglo como desplazamientos en memoria.
  - c. Declaración, inicialización y uso de arreglos en C.
  - d. Conversión implícita entre arreglo y apuntador.
  - e. Paso por referencia usando apuntadores.
2. Estructuras:
  - a. Estructuras y tipos compuestos definidos por el usuario.
  - b. Declaración de estructuras y las sentencias `struct` y `typedef`.
  - c. Inicialización, asignación y acceso a miembros de una estructura.
  - d. Estructuras anidadas.
  - e. Arreglos de estructuras.
3. Elaboración de programas que involucren el uso de arreglos y de estructuras.

### REFERENCIAS:

Bronson G.J., C++ para ingeniería y ciencias, México: International Thomson Editores, 2000. Capítulos 10 y 12.

Savitch W., Resolución de problemas con C++, México: Pearson Education, 2007. Capítulo 10.

Kernighan B.W. y Pike R., El entorno de programación UNIX, México: Prentice-Hall Hispanoamericana, 1987. Capítulos 5 y 6.

Cairó O., Fundamentos de programación. Piensa en C. México: Pearson Education, 2006. Capítulo 5,6 y 8.

### HORAS DE CLASE:

7.5

### OBSERVACIONES:

- Se recomienda proponer algoritmos que resuelvan problemas como:
- Operaciones con vectores: producto punto, producto cruz, obtener el valor máximo y el valor mínimo.
  - Operaciones con matrices: traspuesta, producto, determinar si una matriz es simétrica.
  - Búsqueda en un arreglo.
  - Ordenamiento de un arreglo.
  - Análisis estadístico.



## TEMA 9. Cadenas de caracteres

### OBJETIVOS ESPECÍFICOS:

1. Entender el concepto de cadena de caracteres en C.
2. Conocer las funciones de manejo de cadenas de la biblioteca estándar de C.
3. Conocer las funciones de clasificación y conversión de caracteres de la biblioteca estándar de C.
4. Construir programas que utilicen cadenas y caracteres.

### CONTENIDO:

1. Cadenas de caracteres:
  - a. Cadenas de caracteres como arreglos y como secuencias delimitadas.
  - b. Diferencia entre tamaño y capacidad de una cadena.
  - c. Lectura con formato de palabras y líneas.
2. Funciones para manejo de cadenas:
  - a. La función `strlen`.
  - b. La función `strcpy`.
  - c. La función `strncpy`.
  - d. La función `strcat`.
3. Funciones para manejo de caracteres:
  - a. El código ASCII.
  - b. Funciones de la cabecera estándar `cctype.h`.
4. Elaboración de programas que involucren la manipulación de arreglos de caracteres.

### REFERENCIAS:

Ceballos F.J., C/C++ Curso de Programación, 3ra. Edición, México: RA-MA Editorial, 2007. Capítulo 6.

Cairó O., Fundamentos de programación. Piensa en C. México: Pearson Education, 2006. Capítulo 7.

### HORAS DE CLASE:

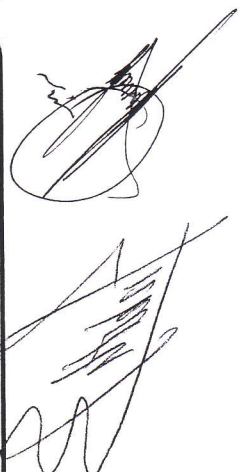
4.5

### OBSERVACIONES:

Se recomienda proponer algoritmos que resuelvan problemas como:

- Determinar si una cadena es palíndromo.
- Ubicar una subcadena (patrón) dentro de una cadena más larga.
- Generar cadenas aleatorias a partir de una cadena inicial.

Los algoritmos podrán describirse mediante pseudocódigo o alguna otra técnica.



Francis



## TEMA 10. Archivos

### OBJETIVOS ESPECÍFICOS:

1. Conocer el concepto de archivo de la biblioteca estándar de C.
2. Construir programas que manipulen el contenido de los archivos.

### CONTENIDO:

1. El concepto de archivo de la biblioteca estándar de C:
  - a. La entrada y salida estándares vistas como archivos.
  - b. Redirección de la entrada y salida estándares
  - c. Salida de errores.
2. Funciones para la manipulación de archivos:
  - a. Modos de apertura de un archivo.
  - b. Las funciones `fscanf` y `fprintf`.
  - c. Fin de archivo y comportamiento de la función `feof`.
  - d. Cierre de archivos.
3. Elaboración de programas que involucren el manejo de archivos.

### REFERENCIAS:

Ceballos F.J., C/C++ Curso de Programación, 3ra. Edición, México: RA-MA Editorial, 2007. Capítulo 9.

Cairó O., Fundamentos de programación. Piensa en C. México: Pearson Education, 2006. Capítulo 9.

### HORAS DE CLASE:

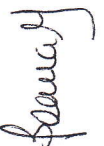
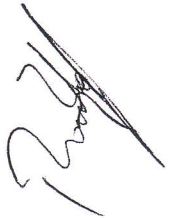
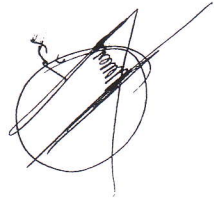
4.5

### OBSERVACIONES:

Se recomienda proponer algoritmos que resuelvan problemas como:

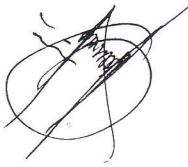
- Codificar un texto extraído de un archivo.
- Decodificar un texto y llevarlo a un archivo.
- Recuperar datos de un archivo y operar con ellos. Por ejemplo: recuperar una matriz, recuperar un registro, recuperar un vector.

Los algoritmos podrán describirse mediante pseudocódigo o alguna otra técnica.



**MODALIDADES DE CONDUCCIÓN DEL PROCESO DE ENSEÑANZA-APRENDIZAJE**

Clase teórico-práctica a cargo del profesor con participación activa del alumno con al menos seis sesiones prácticas utilizando computadoras. Alternativamente modalidad SAI, SAC.



**INFORMACIÓN ADICIONAL**



**MODALIDADES DE EVALUACIÓN**

Evaluación global:

Al menos dos evaluaciones periódicas de resolución de problemas, ejercicios o preguntas conceptuales.

Elaboración y presentación de trabajos y programas.

No hay evaluación terminal.

Las reglas de evaluación serán presentadas en forma escrita por el profesor al inicio del curso.

Evaluación de Recuperación:

Admite evaluación de recuperación, consistente en elaboración de programas, resolución problemas, ejercicios o preguntas conceptuales.

No requiere inscripción previa.



**INFORMACIÓN ADICIONAL**



#### BIBLIOGRAFÍA NECESARIA O RECOMENDABLE

1. Kernighan B.W. y Ritchie D.M., "El lenguaje de programación C, Prentice-Hall Hispanoamericana, 2da ed., México, 1995.
2. Kernighan B.W. y Pike R., "El entorno de programación UNIX", Prentice-Hall Hispanoamericana, México, 1987.
3. Rodríguez C.G. et al., "Ejercicios de programación creativos y recreativos en C++", Prentice-Hall, España, 2002.
4. Zaragoza F.J., "64 ejercicios de programación", México: UAM Azcapotzalco, 2011.
5. Bronson G.J., "C++ para ingeniería y ciencias", International Thomson, México, 2000.
6. Oram y Loukides M., "Programming With GNU software", O'Really Media, EUA, 1996.
7. Ceballos F.J., "C/C++ Curso de Programación", RA-MA, 3ra ed, México, 2007.

#### BIBLIOGRAFÍA ADICIONAL

1. Cairó O., Fundamentos de programación. Piensa en C. México: Pearson Education, 2006.
2. Gallardo López L. y Lara Velázquez P., "Notas de programación estructurada", México:UAM-Azcapotzalco, 2013
3. Joyanes Aguilar L. y Zahonero Martínez I., "Programación en C, C++, Java y UML", McGraw-Hill, 2009
4. Márquez G., Osorio B. y Olvera N., "Introducción a la programación estructurada en C", Pearson, 2012
5. Savitch W., "Resolución de problemas con C++", Pearson Education, México, 2007.

Este programa analítico fue elaborado por una comisión académica del Departamento de Sistemas integrada por los profesores: Ma. Lizbeth Gallardo López, Beatriz A. González Beltrán, Gueorgi Khatchatourov, Pedro Lara Velázquez, Risto Rangel Kouppa, José Luis Pantoja Gallegos, Rafaela Blanca Silva López, Enrique Andrés Tenorio Guillén.

Aprobado

Jefe de Departamento

Visto bueno

Director de División