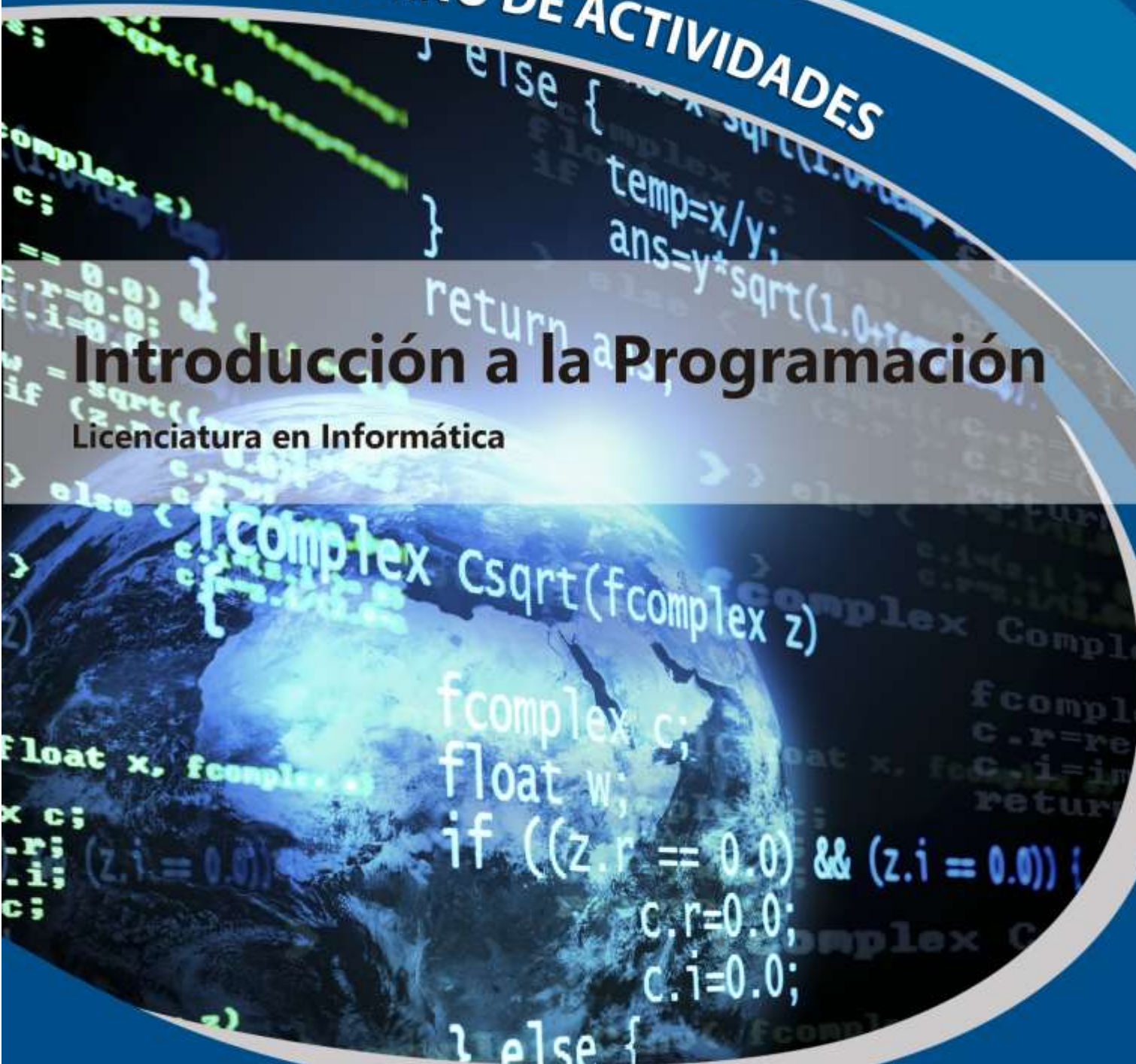




CUADERNO DE ACTIVIDADES

Introducción a la Programación

Licenciatura en Informática





COLABORADORES

DIRECTOR DE LA FCA

Dr. Juan Alberto Adam Siade

SECRETARIO GENERAL

Mtro. Tomás Humberto Rubio Pérez

COORDINACIÓN GENERAL

Mtra. Gabriela Montero Montiel

Jefe de la División SUAyED-FCA-UNAM

COORDINACIÓN ACADÉMICA

Mtro. Francisco Hernández Mendoza

FCA-UNAM

COAUTOR

L.I. Espartaco David Kanagusico Hernández

L.C. Gilberto Manzano Peñaloza

REVISIÓN PEDAGÓGICA

Mayra Lilia Velasco Chacón

CORRECCIÓN DE ESTILO

L.F. Francisco Vladimir Aceves Gaytán

DISEÑO DE PORTADAS

L.CG. Ricardo Alberto Báez Caballero

Mtra. Marlene Olga Ramírez Chavero

DISEÑO EDITORIAL

Mtra. Marlene Olga Ramírez Chavero



Dr. Enrique Luis Graue Wiechers
Rector

Dr. Leonardo Lomelí Vanegas
Secretario General



Dr. Juan Alberto Adam Siade
Director

Mtro. Tomás Humberto Rubio Pérez
Secretario General



Mtra. Gabriela Montero Montiel
Jefa del Sistema Universidad Abierta
y Educación a Distancia

Introducción a la programación

Cuaderno de actividades

Edición: agosto 2017

D.R. © 2017 UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
Ciudad Universitaria, Delegación Coyoacán, C.P. 04510, México, Ciudad de México.

Facultad de Contaduría y Administración
Circuito Exterior s/n, Ciudad Universitaria
Delegación Coyoacán, C.P. 04510, México, Ciudad de México.

ISBN: En trámite
Plan de estudios 2012, actualizado 2016.

"Prohibida la reproducción total o parcial por cualquier medio sin la autorización escrita del titular de los derechos patrimoniales"

"Reservados todos los derechos bajo las normas internacionales. Se le otorga el acceso no exclusivo y no transferible para leer el texto de esta edición electrónica en la pantalla. Puede ser reproducido con fines no lucrativos, siempre y cuando no se mutile, se cite la fuente completa y su dirección electrónica; de otra forma, se requiere la autorización escrita del titular de los derechos patrimoniales."

Hecho en México



Contenido

Datos de identificación	6
Sugerencias de apoyo	7
Instrucciones para trabajar con el cuaderno de actividades	8
Objetivo general de la asignatura y temario oficial	10
Unidad 1. Introducción a la programación	11
Objetivo particular y temario detallado	12
Actividad diagnóstica	13
Actividades de aprendizaje	14
Actividad integradora	17
Cuestionario de reforzamiento	18
Examen parcial de la unidad (de autoevaluación)	19
Repuestas	23
Unidad 2. Tipos de datos elementales	24
Objetivo particular y temario detallado	25
Actividad diagnóstica	26
Actividades de aprendizaje	27
Actividad integradora	29
Cuestionario de reforzamiento	30
Examen parcial de la unidad (de autoevaluación)	31
Repuestas	36
Unidad 3. Control de flujo	37
Objetivo particular y temario detallado	38
Actividad diagnóstica	39
Actividades de aprendizaje	40
Actividad integradora	42
Cuestionario de reforzamiento	43
Examen parcial de la unidad (de autoevaluación)	44
Repuestas	47



Unidad 4. Funciones	48
Objetivo particular y temario detallado	49
Actividad diagnóstica	50
Actividades de aprendizaje	51
Actividad integradora	53
Cuestionario de reforzamiento	54
Examen parcial de la unidad (de autoevaluación)	55
Repuestas	59
Unidad 5. Tipos de datos compuestos	60
Objetivo particular y temario detallado	61
Actividad diagnóstica	62
Actividades de aprendizaje	63
Actividad integradora	65
Cuestionario de reforzamiento	66
Examen parcial de la unidad (de autoevaluación)	67
Repuestas	71
Unidad 6. Manejo de apuntadores	72
Objetivo particular y temario detallado	73
Actividad diagnóstica	74
Actividades de aprendizaje	75
Actividad integradora	76
Cuestionario de reforzamiento	77
Examen parcial de la unidad (de autoevaluación)	78
Repuestas	83



DATOS DE IDENTIFICACIÓN

Introducción a la programación		Clave: 1167
Plan: 2012 (actualizado 2016)		Créditos: 8
Licenciatura: Informática		Semestre: 2°
Área o campo de conocimiento: Desarrollo de sistemas		Horas por semana: 4
Duración del programa: semestral		Requisitos: ninguno
Tipo: Teórica Teoría: 4 Práctica: 0		
Carácter: Obligatoria () Optativa ()		
Seriación: Si (X) No () Obligatoria () Indicativa (X)		
Asignatura con seriación antecedente: Ninguna		
Asignatura con seriación subsecuente: Programación (Estructuras de datos)		



SUGERENCIAS DE APOYO

- Trata de compartir tus experiencias y comentarios sobre la asignatura con tus compañeros, a fin de formar grupos de estudio presenciales o a distancia (comunidades virtuales de aprendizaje, a través de foros de discusión y correo electrónico, etcétera), y puedan apoyarse entre sí.
- Programa un horario propicio para estudiar, en el que te encuentres menos cansado, ello facilitará tu aprendizaje.
- Dispón de periodos extensos para al estudio, con tiempos breves de descanso por lo menos entre cada hora si lo consideras necesario.
- Busca espacios adecuados donde puedas concentrarte y aprovechar al máximo el tiempo de estudio.



Instrucciones para trabajar con el cuaderno de actividades

El programa de la asignatura consta de 6 unidades. Por cada unidad encontrarás una serie de actividades, el número de las mismas varía de acuerdo a la extensión de la unidad.

Notarás que casi todas las unidades comienzan con la elaboración de un mapa conceptual ó mental, esto es con el fin de que tu primera actividad sea esquematizar el contenido total de la unidad para que tengan una mejor comprensión, y dominio total de los temas.

Te recomendamos que leas detenidamente cada actividad a fin de que te quede claro que es lo que tienes que realizar. Si al momento de hacerlo algo no queda claro, no dudes en solicitar el apoyo de tu asesor quien te indicará la mejor forma de realizar tu actividad en asesorías semipresenciales o por correo electrónico para los alumnos de la modalidad abierta, o bien para la modalidad a distancia a través de los medios proporcionados por la plataforma.

Te sugerimos (salvo la mejor opinión de tu asesor), seguir el orden de las unidades y actividades, pues ambas están organizadas para que tu aprendizaje sea gradual. En el caso de los alumnos de la modalidad a distancia, la entrega de actividades está sujeta al plan de trabajo establecido por cada asesor y el trabajo es directamente en plataforma educativa:

<http://fcaenlinea1.unam.mx/licenciaturas/>



La forma en que deberás responder a cada actividad dependerá de la instrucción dada (número de cuartillas, formatos, si hay que esquematizar etcétera).

Una vez que hayas concluido las actividades entrégalas a tu asesor si así él te lo solicita. Los alumnos de la modalidad a distancia, deberán realizar la actividad directamente en la plataforma educativa de acuerdo a la instrucción dada.

Te invitamos a que trabajes estas actividades con el mayor entusiasmo, pues fueron elaboradas considerando apoyarte en tu aprendizaje de ésta asignatura.



Indicaciones:

Notarás que tanto los cuestionarios de reforzamiento como las actividades de aprendizaje, contienen instrucciones tales como “adjuntar archivo”, “trabajo en foro”, “texto en línea”, “trabajo en wiki o en Blog”, indicaciones que aplican específicamente para los estudiantes del SUAYED de la modalidad a distancia. Los alumnos de la modalidad abierta, trabajarán las actividades de acuerdo a lo establecido por el asesor de la asignatura en su plan de trabajo, incluyendo lo que sé y lo que aprendí.



Biblioteca Digital:

Para tener acceso a otros materiales como libros electrónicos, es necesario que te des de alta a la Biblioteca Digital de la UNAM (BIDI). Puedes hacerlo desde la página principal de la FCA <http://www.fca.unam.mx/> **Alumnos >Biblioteca >Biblioteca digital >Clave para acceso remoto >Solicita tu cuenta.** Elige la opción de “Alumno” y llena los campos solicitados. Desde este sitio, también puedes tener acceso a los libros electrónicos.



OBJETIVO GENERAL

El alumno será capaz de implementar algoritmos en un lenguaje de programación.

TEMARIO OFICIAL

(64 horas)

	Horas
1. Introducción a la programación	4
2. Tipos de datos elementales (Variables, Constantes, declaraciones y expresiones y estructura de un programa)	6
3. Control de flujo	14
4. Funciones	18
5. Tipos de datos compuestos (Estructura)	14
6. Manejo de apuntadores	8
Total	64



UNIDAD 1

Introducción a la programación





OBJETIVO PARTICULAR

Será capaz de establecer la diferencia entre los paradigmas de programación e identificar los lenguajes de acuerdo a su nivel y sus principales características.

TEMARIO DETALLADO

(4 horas)

1.Introduccion a la programación

1.1. Concepto de lenguaje de programación

1.2. Paradigmas de programación

1.2.1. Paradigma imperativo

1.2.2. Paradigma orientado a objetos

1.2.3. Paradigma funcional

1.3. Lenguaje máquina

1.4. Lenguajes de bajo nivel

1.5. Lenguajes de alto nivel

1.6. Intérpretes

1.7. Compiladores

1.8. Fases de la compilación

1.9. Notación BNF

1.10. Sintaxis, léxico, semántica



ACTIVIDAD DIAGNÓSTICA

LO QUE SÉ



Texto en línea.

El siguiente cuestionario te permitirá iniciar el aprendizaje de esta unidad con tus conocimientos previos. Cabe destacar que las respuestas de este cuestionario no influyen en tu evaluación.

Con tus propias palabras define:

1. ¿Qué es un lenguaje?
2. ¿Qué es una computadora?
3. ¿Qué entiendes por lenguaje de computadora?
4. Menciona 5 lenguajes de programación que conozcas.



ACTIVIDADES DE APRENDIZAJE



Unidad 1, actividad inicial. *Adjuntar archivo.* A partir del estudio de la bibliografía específica sugerida, elabora un mapa conceptual u [organizador gráfico](#) con los temas de la unidad. Puedes auxiliarte de algunos programas como Mindjet [MindManager](#).

1. **Unidad 1, actividad 1. *Adjuntar archivo.*** Define con tus propias palabras el concepto de Lenguaje de Programación e indica cuáles son sus principales elementos.
2. **Unidad 1, actividad 2. *Adjuntar archivo.*** Investiga las características, elementos, ventajas y desventajas de las generaciones de los lenguajes de programación.
3. **Unidad 1, actividad 3. *Adjuntar archivo.*** Realiza un algoritmo que sume tres números. Consulta el documento [Ejemplos y ejercicios](#). Lee el ejercicio 3 para que comprendas cómo realizar este algoritmo.

Comunidad de programadores. (2016). *Algoritmia*. Obtenido en:
<https://pastranamoreno.files.wordpress.com/2012/05/ejercicios-resueltos.pdf>
Consultado 7 de marzo de 2017



4. **Unidad 1, actividad 4. *Adjuntar archivo*.** Modifica este programa para que sume tres números.

```
# include <stdio.h>
main()
{
    int c=0;
    int a=2;
    int b=3;
    a= a+b;
    printf("%d", c);
    return(0);
```

5. **Unidad 1, actividad 5. *Adjuntar archivo*.** Tomando como base lo explicado en el tema de Paradigma orientado a objetos, investiga los conceptos objeto, clase y herencia. Posteriormente, elabora un documento donde redactes la definición de los conceptos solicitados e indica sus características e importancia.
6. **Unidad 1, actividad 6. *Adjuntar archivo*.** Elabora un cuadro comparativo de la programación orientada a objetos y la programación imperativa.
7. **Unidad 1, actividad 7. *Adjuntar archivo*.** Realiza una búsqueda sobre:
- Lenguaje ensamblador
 - Lenguaje de bajo nivel
 - Lenguaje de alto nivel
- Identifica sus iniciadores, conceptos, principales características, diferencias entre los lenguajes, ventajas desventajas, estructura, y proporciona 2 ejemplos de cada uno.
8. **Unidad 1, actividad 8. *Adjuntar archivo*.** Elabora un programa en lenguaje ensamblador que sume dos números.
9. **Unidad 1, actividad 9. *Adjuntar archivo*.** Elabora una tabla comparativa entre los lenguajes de bajo y alto nivel, indicando aplicaciones, ventajas y desventajas.



10. **Unidad 1, actividad 10. *Adjuntar archivo.*** Elabora un cuadro comparativo de intérprete y compilador, indicando características, ventajas y desventajas.
11. **Unidad 1, actividad 11. *Adjuntar archivo.*** Investiga y elabora un cuadro comparativo de las diferencias entre el intérprete y el compilador.
12. **Unidad 1, actividad 12. *Adjuntar archivo.*** Elabora un programa en C e identifica los pasos que realiza el compilador para generar el programa ejecutable.
13. **Unidad 1, actividad complementaria. *Adjuntar archivo.*** A partir del estudio de la unidad, realiza la siguiente actividad, misma que encontraras en el foro de la asignatura. Cabe señalar que esta será colocada en el foro por tu asesor.



ACTIVIDAD INTEGRADORA

LO QUE APRENDÍ



Texto en línea.

Escribe el código fuente de un programa que obtenga el promedio de tres números, empleando cada uno de los paradigmas de programación: imperativo, orientado a objetos y funcional.



CUESTIONARIO DE REFORZAMIENTO



Adjuntar archivo. Responde las siguientes preguntas.

1. Señala qué es un lenguaje de programación.
2. ¿Qué es el código fuente?
3. ¿Qué es el código objeto?
4. ¿Qué es el código ejecutable?
5. ¿En qué nivel se clasifica al lenguaje C y por qué?
6. ¿Qué es un algoritmo?
7. ¿Qué es un programa?
8. ¿Qué es un compilador?
9. ¿Qué es un intérprete?
10. ¿Qué es un objeto?



EXAMEN PARCIAL (de autoevaluación)



I. Responde verdadero (V) o falso (F).

	V	F
1. La programación estructurada utiliza un diseño modular.	☐	☐
2. C es un lenguaje estructurado.	☐	☐
3. Un ejemplo de una estructura de control es la iteración.	☐	☐
4. <i>Printf</i> muestra un mensaje en pantalla.	☐	☐
5. <i>Scanf</i> muestra un mensaje en pantalla.	☐	☐
6. La programación estructurada utiliza objetos.	☐	☐
7. C es un lenguaje orientado a objetos.	☐	☐
8. Un ejemplo de una estructura de control es la iteración.	☐	☐
9. Una función es sinónimo de una clase.	☐	☐
10. <i>#include</i> se utiliza para acceder a una librería de funciones.	☐	☐
11. Todos los leguajes de programación usan compatibilidades.	☐	☐
12. C ++ es un lenguaje orientado a objetos.	☐	☐
13. La programación orientada a objetos divide un problema complejo en problemas más sencillos.	☐	☐
14. El lenguaje PHP utiliza un compilador.	☐	☐
15. El lenguaje C es un ejemplo de un lenguaje de bajo nivel.	☐	☐



II. Selecciona la respuesta correcta.

___ 1. Este lenguaje se refiere a un sistema de códigos directamente interpretable por un circuito micro programable.	
___ 2. Consiste en una cadena de instrucciones de lenguaje de máquina.	a) PHP
___ 3. Este lenguaje proporciona poca o ninguna abstracción del microprocesador de una computadora.	b) Ensamblador
___ 4. Se refiere a un tipo de programa informático que se encarga de traducir un archivo fuente escrito en un archivo objeto que contiene código máquina.	c) Lenguaje de máquina
___ 5. Se caracteriza por expresar los algoritmos de una manera adecuada a la capacidad cognitiva humana.	d) Lenguaje de bajo nivel
___ 6. Es ejemplo de un lenguaje de alto nivel.	e) Programa de computadoras
	f) Lenguaje de alto nivel

**III. Selecciona la respuesta correcta.**

1. El código fuente es un:

☐ a) Código de computadora	☐ c) Paradigma
☐ b) Diagrama de flujo	☐ d) Conjunto de caracteres entendibles por un ser humano

2. El código objeto es:

☐ a) Un código entendible por la computadora	☐ c) Sinónimo de un compilador
☐ b) Sinónimo de un intérprete	☐ d) Un código entendible por un ser humano

3. El código ejecutable:

☐ a) Se obtiene usando un linker	☐ c) Es un sinónimo de paradigma
☐ b) Se obtiene usando un intérprete	☐ d) Es un sinónimo de objeto

4. Una clase es:

☐ a) Sinónimo de objeto	☐ c) El conjunto de funciones con características similares
☐ b) Sinónimo de herencia	☐ d) El conjunto de objetos con características similares

5. La función tiene su equivalente, en la programación orientada a objetos, en el concepto de:

☐ a) Herencia	☐ c) Método
☐ b) Clase	☐ d) Objeto



6. Un intérprete lee línea por línea:

☐ a) El código fuente	☐ c) El código ejecutable
☐ b) El código objeto	☐ d) Las librerías

7. Es un ejemplo de una librería.

☐ a) Scanf	☐ c) Stdio.h
☐ b) Void	☐ d) Java

8. Python utiliza un:

☐ a) Enlazador	☐ c) Compilador
☐ b) Compilador y un intérprete	☐ d) Intérprete

9. En un programa, un linker:

☐ a) Agrega librerías	☐ c) Genera un código fuente
☐ b) Genera un código objeto	☐ d) Es sinónimo de clase



RESPUESTAS

EXAMEN DE AUTOEVALUACIÓN



En este apartado encontrarás las respuestas al examen por unidad.

UNIDAD 1	
I. Solución	
1. V	9. F
2. V	10. V
3. F	11. F
4. V	12. V
5. F	13. F
6. F	14. F
7. V	15. F
8. V	

UNIDAD 1
II. Solución
1. c
2. e
3. d
4. b
5. f
6. a

UNIDAD 1	
III. Solución	
1. d	6. a
2. a	7. c
3. a	8. d
4. c	9. a
5. c	



UNIDAD 2

Tipos de datos elementales

(Variables, constantes, declaraciones y expresiones y estructura de un programa.)





OBJETIVO PARTICULAR

Deberá conocer los componentes básicos de la programación y la estructura de un programa.

TEMARIO DETALLADO

(6 horas)

2. Tipos de datos elementales. (Variables, constantes, declaraciones y expresiones y estructura de un programa)

2.1. Tipos de datos

2.2. Palabras reservadas

2.3. Identificadores

2.4. Operadores

2.5. Expresiones y reglas de prioridad

2.6. Variables y constantes

2.7. Estructura de un programa



ACTIVIDAD DIAGNÓSTICA

LO QUE SÉ



Texto en línea.

Define con tus propias palabras los siguientes términos:

1. Tipo de dato.
2. Constante.
3. Variable.



ACTIVIDADES DE APRENDIZAJE



Unidad 2, actividad inicial. *Adjuntar archivo.* A partir del estudio de la bibliografía específica sugerida, elabora un mapa conceptual u [organizador gráfico](#) con los temas de la unidad. Puedes auxiliarte de algunos programas como Mindjet [MindManager](#).

1. **Unidad 2, actividad 1. *Adjuntar archivo.*** Investiga los tipos de datos que se emplean en el lenguaje C++ y compáralos con los empleados con los usados en C. Desarrolla un cuadro comparativo con los tipos de datos en C y C++.
2. **Unidad 2, actividad 2. *Adjuntar archivo.*** Investiga los tipos de datos que se emplean en el lenguaje Java. Realiza un cuadro comparativo de similitudes y diferencias de 5 tipos de datos entre los lenguajes C++ y Java, agrega la sintaxis de la forma de declararlos y utilizarlos.
3. **Unidad 2, actividad 3. *Adjuntar archivo.*** Investiga cómo se construye un identificador y cuál es su función, así como la función de la palabra reservada *enum*.
4. **Unidad 2, actividad 4. *Adjuntar archivo.*** Investiga la función de las palabras reservadas de un lenguaje de programación y cuál es su principal característica.
5. **Unidad 2, actividad 5. *Adjuntar archivo.*** Elabora un programa en C que determine si un número es par o impar.
6. **Unidad 2, actividad 6. *Adjuntar archivo.*** Elabora un programa en C que determine si dos números introducidos por el usuario son iguales.



7. **Unidad 2, actividad 7. *Adjuntar archivo.*** Investiga e indica cuál es la prioridad del operador de exponente en el lenguaje Visual Basic.
8. **Unidad 2, actividad 8. *Adjuntar archivo.*** Investiga y describe el uso de variables locales y globales en Python.
9. **Unidad 2, actividad 9. *Adjuntar archivo.*** Investiga y describe el uso de constantes en Java.
10. **Unidad 2, actividad 10. *Adjuntar archivo.*** Investiga y escribe la estructura de un programa escrito en Visual Basic.
11. **Unidad 2, actividad complementaria. *Adjuntar archivo.*** A partir del estudio de la unidad, realiza la siguiente actividad, misma que encontraras en el foro de la asignatura. Cabe señalar que esta será colocada en el foro por tu asesor.



ACTIVIDAD INTEGRADORA

LO QUE APRENDÍ



Adjuntar archivo.

Elabora un programa en donde utilices, por lo menos, una de las expresiones aritméticas (lógicas y relacionales), a la vez que cambies la prioridad de los operadores aritméticos utilizando paréntesis.



CUESTIONARIO DE REFORZAMIENTO



Adjuntar archivo. Responde las siguientes preguntas.

1. ¿Qué es una variable de tipo global?
2. ¿Qué es una variable de tipo local?
3. ¿Qué significa la palabra getch?
4. ¿Qué significa la palabra printf?
5. ¿Qué es un tipo definido por el usuario?
6. ¿Qué significa la palabra scanf?
7. ¿Qué es una variable?
8. ¿Qué es una constante?
9. Describe los operadores lógicos.
10. ¿Qué es una conversión de tipos?



EXAMEN PARCIAL

(de autoevaluación)



I. De acuerdo con lo estudiado en estos temas, lee con atención las siguientes palabras y escribe el inciso correspondiente.

Palabras reservadas	Identificadores
1.	5.
2.	6.
3.	7.
4.	8.

a. extern	e. suma_1
b. lf	f. while
c. _t	g. float
d. y2	h. largo



II. Selecciona la respuesta correcta.

1. Un tipo de dato:

☐ a) Permite usar varios elementos en una estructura	☐ c) Determina los objetos por usar
☐ b) Determina las clases por usar	☐ d) Permite usar un rango de datos

2. El tamaño del tipo char es de:

☐ a) 1 byte	☐ c) 4 bytes
☐ b) 2 bytes	☐ d) 6 bytes

3. El tipo doublé se emplea para describir:

☐ a) Números enteros	☐ c) Números con decimales
☐ b) Letra o dígito	☐ d) Números y letras

4. La siguiente es una palabra reservada

☐ a) Var	☐ c) If
☐ b) Not	☐ d) And

5. Una de las características de un identificador es que:

☐ a) Deben de comenzar por letra o por “_”	☐ c) Son símbolos cuyo significado está predefinido
☐ b) Son secuencias que se forman empleando solo letras	☐ d) Permite nuevos nombres para tipo de datos

6. Un ejemplo de operador relacional es:

☐ a) >	☐ c) &&
☐ b) AND	☐ d) %



7. Las variables locales se declaran:

☐ a) Dentro de una función	☐ c) Fuera de una función
☐ b) Después de main ()	☐ d) Antes de main ()

8. El nombre de una variable no debe sobrepasar los:

☐ a) 20 caracteres	☐ c) 31 caracteres
☐ b) 30 caracteres	☐ d) 32 caracteres

9. Las variables que se almacenan en registros del microprocesador se denominan:

☐ a) Locales	☐ c) Externas
☐ b) Globales	☐ d) De registro

10. Son variables que se declaran dentro de una función.

☐ a) Locales	☐ c) Externas
☐ b) Globales	☐ d) De registro

11. Para definir nuevos tipos de datos se utiliza la palabra reservada:

☐ a) Register	☐ c) Casting
☐ b) Struct	☐ d) Typedef

12. Una constante:

☐ a) Es sinónimo de una variable	☐ c) Tiene ámbito global
☐ b) Tiene ámbito local	☐ d) Mantiene su valor durante la ejecución del programa

13. La siguiente es una palabra reservada en C.

☐ a) Echo	☐ c) If
☐ b) Println	☐ d) Writeln



14. Un ejemplo de un operador lógico es:

☐ a) >	☐ c) &&
☐ b) And	☐ d) %

15. El operador módulo se escribe así:

☐ a) ++	☐ c) %
☐ b) --	☐ d)

16. Las variables globales se declaran:

☐ a) Dentro de una función	☐ c) Fuera de una función
☐ b) Después de las constantes	☐ d) Antes de main()

17. Un identificador puede empezar con:

☐ a) Un número	☐ c) Una letra
☐ b) Un espacio	☐ d) Un tabulador

18. El operador de negación es:

☐ a)	☐ c) %
☐ b) &&	☐ d) !

19. Es ejemplo de una biblioteca.

☐ a) Stdio.h	☐ c) Define
☐ b) Include	☐ d) Const

20. De los siguientes, el tipo de dato más grande es:

☐ a) Int	☐ c) Float
☐ b) Char	☐ d) Long int



21. Palabra reservada para devolver un valor.

☐ a) #include	☐ c) Stdio.h
☐ b) #define	☐ d) Return()

III. Observa las palabras/signos, y en base a lo que vimos en esta unidad, subraya las palabras /signos de la siguiente forma:

ROJO - las que correspondan a operadores aritméticos

AZUL - las que correspondan a asignaciones

VERDE - las que correspondan a operadores relacionales y lógicos





RESPUESTAS

EXAMEN DE AUTOEVALUACIÓN



En este apartado encontrarás las respuestas al examen por unidad.

UNIDAD 2	
I. Solución	
1.	a
2.	f
3.	g
4.	b
5.	h
6.	d
7.	c
8.	e

UNIDAD 2	
II. Solución	
1. d	12. d
2. a	13. c
3. c	14. c
4. c	15. c
5. a	16. c
6. a	17. c
7. a	18. d
8. c	19. a
9. d	20. c
10. a	21. d
11. d	

UNIDAD 2	
III. Solución	
a=2*(++c),	++
variable=variable1;	
>=	+
%	



UNIDAD 3

Control de flujo





OBJETIVO PARTICULAR

Podrá utilizar las principales estructuras de la programación.

TEMARIO DETALLADO (14 horas)

3. Control de flujo

3.1. Estructura secuencial

3.2. Estructura alternativa

3.3. Estructura repetitiva



ACTIVIDAD DIAGNÓSTICA

LO QUE SÉ



Texto en línea.

Explica qué es la programación estructurada y define cada uno de sus elementos.



ACTIVIDADES DE APRENDIZAJE



Unidad 3, actividad inicial. *Adjuntar archivo.* A partir del estudio de la bibliografía específica sugerida, elabora un mapa conceptual u [organizador gráfico](#) con los temas de la unidad. Puedes auxiliarte de algunos programas como Mindjet [MindManager](#).

1. **Unidad 3, actividad 1. *Adjuntar archivo.*** Elabora un programa en C que obtenga la edad de una persona utilizando como base, la fecha de nacimiento.
2. **Unidad 3, actividad 2. *Adjuntar archivo.*** Elabora un programa que determine la mensualidad que debe pagar una persona si pide un préstamo de \$10,000.00 pesos, tomando en cuenta una tasa de interés de 50% anual.
3. **Unidad 3, actividad 3. *Adjuntar archivo.*** Escribe un programa en lenguaje C que sume los números pares e impares dentro del rango del 1 al 100.
4. **Unidad 3, actividad 4. *Adjuntar archivo.*** Escribe un programa que valide una contraseña ingresada por el usuario. El usuario tendrá tres intentos para ingresar la contraseña correcta, de lo contrario, el programa mostrará el mensaje “usuario no autorizado”.
5. **Unidad 3, actividad 5. *Adjuntar archivo.*** Elabora un programa que obtenga la suma de los números ingresados por el usuario. Cuando el usuario ingrese un número cero, el programa se detendrá y mostrará el acumulado de la suma.



6. **Unidad 3, actividad 6.** *Adjuntar archivo.* Escribe un programa que obtenga la potencia de una base y un exponente dados, ocupando únicamente la operación de suma; no se deberá utilizar la multiplicación o la función de potencia.
7. **Unidad 3, actividad complementaria.** *Adjuntar archivo.* A partir del estudio de la unidad, realiza la siguiente actividad, misma que encontraras en el foro de la asignatura. Cabe señalar que esta será colocada en el foro por tu asesor.



ACTIVIDAD INTEGRADORA

LO QUE APRENDÍ



Adjuntar archivo.

Elabora un programa que, a través del siguiente menú, realice las operaciones de un cajero automático:

Menú Principal

- | | |
|--------------|-------------|
| 1. Depósitos | 3. Consulta |
| 2. Retiros | 4. Salir |

Restricciones:

→ El programa seguirá ejecutándose en tanto el usuario no presione la opción 4, de Salir.

→ El usuario:

- ♦ Podrá realizar el número de depósitos que desee por cualquier cantidad.
- ♦ No podrá realizar un retiro por una cantidad mayor al saldo de la cuenta.
- ♦ Podrá consultar en cualquier momento el saldo de su cuenta.

→ Utiliza las estructuras de control aprendidas en esta unidad.



CUESTIONARIO DE REFORZAMIENTO



Adjuntar archivo. Responde las siguientes preguntas.

1. Define qué es una estructura secuencial.
2. Define qué es una estructura alterativa.
3. Explica qué es la estructura repetitiva.
4. Es un ciclo con un número determinado de iteraciones. Entonces nos estamos refiriendo a:
5. Es un ciclo con un número indeterminado de iteraciones. Hacemos referencia a:
6. Es un ciclo que se ejecuta al menos una vez, independientemente de que se cumpla o no una condición. Entonces estamos hablando de:
7. ¿Cuál es la función de la instrucción *default*?
8. ¿Cuál es la función de la instrucción *switch*?
9. ¿Cuál es la función de la palabra reservada *case*?
10. ¿Cuál es la función de la palabra reservada *break*?



EXAMEN PARCIAL

(de autoevaluación)



I. Responde verdadero (V) o falso (F).

	V	F
1. La estructura alternativa permite que el programa fluya de acuerdo a una condición.	☐	☐
2. La estructura alternativa utiliza la palabra reservada <i>while</i> .	☐	☐
3. En la estructura alternativa <i>if-then</i> , si la condición no se cumple se puede utilizar la palabra <i>else</i> para cambiar el flujo.	☐	☐
4. En lugar de utilizar varios <i>if</i> , una opción es emplear la palabra reservada <i>case</i> .	☐	☐
5. Dentro del <i>case</i> , se evalúa la opción.	☐	☐

II. Elige la respuesta correcta a las siguientes preguntas.

1. Tipo de estructura que permite que todas las sentencias se ejecuten sin posibilidad de omitir alguna.

☐ a) Secuencial	☐ c) Selectiva
☐ b) Alternativa	☐ d) Repetitiva



2. Tipo de estructura que permite que el flujo de un programa se bifurque.

☐ a) Alternativa	☐ c) If
☐ b) Selectiva	☐ d) Else

3. Tipo de estructura que permite que una sentencia se ejecute un número determinado de veces.

☐ a) Do	☐ c) For
☐ b) While	☐ d) Main ()

4. Tipo de estructura que permite que una sentencia se ejecute un número indeterminado de veces.

☐ a) Default	☐ c) For
☐ b) While	☐ d) Else

5. Es un ciclo que permite que se ejecute una sentencia, al menos una vez, independientemente de que se cumpla o no una condición.

☐ a) Ciclo	☐ c) Do-while
☐ b) While	☐ d) For

6. Un ciclo de tipo for es un ciclo:

☐ a) Con un número determinado de iteraciones	☐ c) Infinito
☐ b) Con un número indeterminado de iteraciones	☐ d) De tipo do-while

7. *break* es usado para:

☐ a) Salir de un ciclo	☐ c) Determinar el flujo de un programa
☐ b) Entrar a un ciclo	☐ d) Crear una función



8. La siguiente es una palabra reservada que se usa en la función *switch* en caso de no se cumpla ningún caso.

☐ a) Break	☐ c) While
☐ b) Default	☐ d) Else

9. Es una palabra reservada que permite que el programa se ejecute de acuerdo al cumplimiento de una condición.

☐ a) If	☐ c) Case
☐ b) Where	☐ d) For

10. Es una función que permite la selección de varias opciones.

☐ a) Include	☐ c) Break
☐ b) Switch	☐ d) For



RESPUESTAS

EXAMEN DE AUTOEVALUACIÓN



En este apartado encontrarás las respuestas al examen por unidad.

UNIDAD 3
I. Solución
1. V
2. F
3. V
4. V
5. F

UNIDAD 3
II. Solución
1. a
2. a
3. c
4. b
5. c
6. a
7. a
8. b
9. a
10. b



UNIDAD 4

Funciones





OBJETIVO PARTICULAR

Utilizar las funciones preconstruidas y podrá desarrollar sus propias funciones; identificará el alcance de las variables utilizadas y aplicará la recursividad.

TEMARIO DETALLADO (18 horas)

4. Funciones

4.1. Internas

4.1.1. Funciones de caracteres y cadenas

4.1.2. Funciones matemáticas

4.1.3. Funciones de conversión

4.2. Definidas por el usuario

4.3. Ámbito de variables (locales y globales)

4.4. Recursividad



ACTIVIDAD DIAGNÓSTICA

LO QUE SÉ



Adjuntar archivo.

Con base en lo que hemos visto en las unidades anteriores y por lo que conoces de este tema, intenta realizar una función en pseudocódigo que determine el mayor de dos números introducidos por el usuario. Cualquier duda pregúntala a tu asesor.



ACTIVIDADES DE APRENDIZAJE



Unidad 4, actividad inicial. *Adjuntar archivo.* A partir del estudio de la bibliografía específica sugerida, elabora un mapa conceptual u [organizador gráfico](#) con los temas de la unidad. Puedes auxiliarte de algunos programas como Mindjet [MindManager](#).

1. **Unidad 4, actividad 1. *Adjuntar archivo.*** Elabora un programa que, a través de una función, transforme una cadena introducida por el usuario en minúsculas, a su equivalente en mayúsculas.
2. **Unidad 4, actividad 2. *Adjuntar archivo.*** Elabora un programa en C que acepte el ingreso de tres números y que, por medio de una función, los ordene en forma ascendente.
3. **Unidad 4, actividad 3. *Adjuntar archivo.*** Elabora una función que busque un número que introduzca el usuario, en un arreglo de 100 números.
4. **Unidad 4, actividad 4. *Adjuntar archivo.*** Elabora una función que determine los factores de un número introducido por el usuario; por ejemplo: si se ingresa el número 8, los factores del número serán 2, 2, 2 y 1.
5. **Unidad 4, actividad 5. *Adjuntar archivo.*** Escribe una función que obtenga las combinaciones de tres enteros introducidos por el usuario. Ejemplo: si se ingresan los números 1, 3, y 5, se imprimirán las combinaciones 1,3,5; 1,5,3; 3,1,5; 3,5,1; 5,1,3 y 5,3,1.
6. **Unidad 4, actividad 6. *Adjuntar archivo.*** Elabora una función que calcule el pago de mensualidades para una deuda de \$16,000.00, tomando en cuenta un plazo de un año, y un interés anual de 24 %; utiliza variables globales.



7. **Unidad 4, actividad 7. *Adjuntar archivo.*** Elabora una función que obtenga la factorial de un número. Ejemplo: si se ingresa el 4, el resultado es 24, porque $4 \times 3 \times 2 \times 1 = 24$
8. **Unidad 4, actividad complementaria. *Adjuntar archivo.*** A partir del estudio de la unidad, realiza la siguiente actividad, misma que encontraras en el foro de la asignatura. Cabe señalar que esta será colocada en el foro por tu asesor.



ACTIVIDAD INTEGRADORA

LO QUE APRENDÍ



Adjuntar archivo.

Elabora un programa que muestre el siguiente menú:

1. Ingresa un número
2. Incrementa un número
3. Decrementa un número
4. Salir

Cada una de las opciones del menú llamará a una función que realizará una tarea específica, como se indica a continuación:

- La opción 1 aceptará el ingreso de un número entero; si no se ocupa esta opción, se iniciará con el número 10.
- La opción 2 le sumará un 1 al número anterior; y la opción 3 le restará un 1 al número mencionado. Utiliza variables globales.



CUESTIONARIO DE REFORZAMIENTO



Adjuntar archivo. Responde las siguientes preguntas.

1. ¿Qué es una función?
2. ¿Qué es la recursividad?
3. ¿Qué significa la palabra void?
4. ¿Qué es un parámetro?
5. ¿Qué significa main?
6. ¿Qué es un parámetro por valor?
7. ¿Qué es un parámetro por referencia?
8. ¿Qué es una función desarrollada por el usuario?
9. ¿Cuál es la función de return?
10. ¿Qué es una función interna?



EXAMEN PARCIAL

(de autoevaluación)



I. Selecciona la respuesta correcta.

1. Las funciones isalpha se encuentra en la librería:

☐ a) Math.h	☐ c) Stdio.h
☐ b) String.h	☐ d) Ctype.h

2. Las funciones strcat se encuentran en la librería:

☐ a) Math.h	☐ c) Stdio.h
☐ b) String.h	☐ d) Ctype.h

3. Es una función que transforma una cadena en un valor entero:

☐ a) Atol	☐ c) Itoa
☐ b) Atoi	☐ d) Sprintf

4. La función randomize se encuentra en la librería:

☐ a) Math.h	☐ c) Stdio.h
☐ b) String.h	☐ d) Ctype.h



5. Es una función que transforma un número a su equivalente en cadena:

☐ a) Atol	☐ c) Itoa
☐ b) Atoi	☐ d) Sprintf

6. Cuando se copia el contenido del argumento al parámetro de la función se hace un paso de parámetros por:

☐ a) Valor	☐ c) Apuntadores
☐ b) Referencia	☐ d) Variables

7. Para devolver un valor se usa la palabra reservada:

☐ a) Void	☐ c) Int
☐ b) Sizeof	☐ d) Return

8. La palabra reservada void, antes de una función, indica que la función:

☐ a) Devuelve un valor	☐ c) Devuelve un entero
☐ b) No devuelve nada	☐ d) Devuelve un tipo char

9. Para desarrollar una función el primer paso es:

☐ a) Definir los parámetros	☐ c) Llamar a la función
☐ b) Definir las variables	☐ d) Crear el prototipo

10. La palabra reservada void en los parámetros indica que:

☐ a) La función carece de parámetros	☐ c) La función tiene más de un parámetro
☐ b) La función tiene un parámetro	☐ d) nN devuelve valor alguno

11. La palabra *int* antes de una función indica que:

☐ a) La función es de tipo entero	☐ c) Sus parámetros son enteros
☐ b) La función devuelve un entero	☐ d) Que la función no devuelve valores



12. El parámetro de una función puede ser:

☐ a) Un tipo de dato	☐ c) Una biblioteca
☐ b) Otra función	☐ d) Variables o constantes

13. Las variables de tipo *register*:

☐ a) Son variables globales	☐ c) Son variables locales
☐ b) Se almacenan en registros del microprocesador	☐ d) Son variables externas

14. La función *getch()* se encuentra en la biblioteca:

☐ a) String.h	☐ c) Conio.h
☐ b) Stdio.h	☐ d) Dos.h

15. La función *islower* devuelve un:

☐ a) Entero	☐ c) Carácter
☐ b) Flotante	☐ d) Long

16. Para inicializar una región de memoria se usa la función:

☐ a) Scanf	☐ c) Cin
☐ b) Memset	☐ d) Cout

17. La función *isalpha* se encuentra en la biblioteca:

☐ a) Ctype.h	☐ c) Sodio.h
☐ b) String.h	☐ d) Conio.h

18. La función *sqrt* obtiene:

☐ a) La potencia de un número	☐ c) El logaritmo de un número
☐ b) La raíz cuadrada de un número	☐ d) El coseno de un número



19. La biblioteca que contiene las funciones de conversión es:

☐ a) String	☐ c) Stdlib
☐ b) Conio	☐ d) Math

20. Una función recursiva es una función que:

☐ a) Se llama a sí misma	☐ c) No devuelve valores
☐ b) Devuelve valores	☐ d) Usa parámetros



RESPUESTAS

EXAMEN DE AUTOEVALUACIÓN



En este apartado encontrarás las respuestas al examen por unidad.

UNIDAD 4	
I. Solución	
1. d	11. b
2. b	12. b
3. b	13. b
4. d	14. c
5. a	15. a
6. a	16. b
7. d	17. a
8. b	18. b
9. d	19. c
10. a	20. a



UNIDAD 5

Tipos de datos compuestos





OBJETIVO PARTICULAR

Podrá utilizar arreglos unidimensionales, multidimensionales y estructuras, para almacenar y procesar datos para aplicaciones específicas.

TEMARIO DETALLADO (14 horas)

5. Tipos de datos compuestos

5.1. Arreglos Unidimensionales
5.2. Arreglos Multidimensionales
5.3. Arreglos y cadenas
5.4. Estructuras



ACTIVIDAD DIAGNÓSTICA

LO QUE SÉ



Adjuntar archivo.

Enlista cinco conjuntos de valores que consideres que se puedan almacenar en la memoria de una computadora, y que sirvan para realizar alguna tarea en específico.



ACTIVIDADES DE APRENDIZAJE



Unidad 5, actividad inicial. *Adjuntar archivo.* A partir del estudio de la bibliografía específica sugerida, elabora un mapa conceptual u [organizador gráfico](#) con los temas de la unidad. Puedes auxiliarte de algunos programas como Mindjet [MindManager](#).

1. **Unidad 5, actividad 1. *Adjuntar archivo.*** Escribe un programa que determine la suma de los números almacenados en la diagonal de un arreglo de tamaño 4 x 4.
2. **Unidad 5, actividad 2. *Adjuntar archivo.*** Elabora un programa que determine si el contenido de dos arreglos, son iguales.
3. **Unidad 5, actividad 3. *Adjuntar archivo.*** Elabora un programa en C que multiplique dos matrices, ambas de tamaño 3 X 3.
4. **Unidad 5, actividad 4. *Adjuntar archivo.*** Haz un programa que cuente la cantidad de letras que contenga una frase, ingresada por el usuario, almacenada en un arreglo de caracteres.
5. **Unidad 5, actividad 5. *Adjuntar archivo.*** Elabora un programa que cuente la cantidad de palabras de una frase almacenada en un arreglo de caracteres.
6. **Unidad 5, actividad 6. *Adjuntar archivo.*** Elabora un programa para almacenar los datos de un alumno, dichos datos son: número de cuenta, nombre, licenciatura, semestre en curso y promedio. Dicha información será almacenada en una estructura de registro.



7. **Unidad 5, actividad complementaria.** *Adjuntar archivo.* A partir del estudio de la unidad, realiza la siguiente actividad, misma que encontraras en el foro de la asignatura. Cabe señalar que esta será colocada en el foro por tu asesor.



ACTIVIDAD INTEGRADORA

LO QUE APRENDÍ



Adjuntar archivo.

Resuelve mediante un programa escrito en lenguaje c el siguiente problema:
una escuela nos solicita un programa para capturar las calificaciones de alumnos por materia, obtener el promedio por alumno, por materia y el promedio general del grupo.

Emplea un arreglo multidimensional para almacenar las calificaciones de 6 materias obtenidas por 5 alumnos y realizar los cálculos solicitados.



CUESTIONARIO DE REFORZAMIENTO



Adjuntar archivo. Responde las siguientes preguntas.

1. ¿Qué es un arreglo?
2. ¿Qué es un arreglo unidimensional?
3. ¿Qué es un arreglo multidimensional?
4. ¿Qué es el índice de un arreglo?
5. ¿Qué es una estructura?
6. ¿Qué es un miembro de una estructura?
7. ¿Cuántos tipos de datos puede almacenar un arreglo?
8. ¿Cuántos tipos de datos puede almacenar una estructura?
9. ¿Qué es una cadena?
10. ¿Cuál es la utilidad de la función flushall()?



EXAMEN PARCIAL

(Autoevaluación)



I. Selecciona la respuesta correcta.

1. Un arreglo es:

☐ a) Un grupo de elementos del mismo tipo	☐ c) Un tipo de dato
☐ b) Un grupo de elementos de tipos distintos	☐ d) Una variable

2. Un arreglo puede usar:

☐ a) Cualquier tipo de dato	☐ c) Solo números
☐ b) Solo cadenas	☐ d) Solo flotantes

3. Si el arreglo se inicializa con una cadena se usa:

☐ a) ´	☐ c) (
☐ b) “	☐ d) {

4. El primer elemento de un arreglo se ubica en la posición:

☐ a) 0	☐ c) -1
☐ b) 1	☐ d) 2



5. Un arreglo unidimensional es sinónimo de un:

☐ a) Grupo de elementos de distinto tipo	☐ c) Vector de vectores
☐ b) Vector	☐ d) Arreglo con tres índices

6. Un arreglo multidimensional es un:

☐ a) Grupo de elementos de distinto tipo	☐ c) Arreglo de arreglos
☐ b) Vector	☐ d) Arreglo con tres índices

7. Para acceder a un arreglo bidimensional se usa:

☐ a) Un índice	☐ c) Dos índices
☐ b) Un vector	☐ d) Tres índices

8. Si se almacena un número introducido por el usuario en un arreglo este debe ser antecedido por:

☐ a) &	☐ c) &&
☐ b) *	☐ d) ->

9. Un arreglo bidimensional es sinónimo de:

☐ a) Vector	☐ c) Estructura
☐ b) Matriz	☐ d) Enumeración

10. Si un arreglo es declarado de tipo carácter, es suficiente declarar la biblioteca:

☐ a) Stdio.h	☐ c) Conio.h
☐ b) String.h	☐ d) Stdlib.h

11. Para una cadena de 5 letras se deben asignar:

☐ a) 5 espacios en un arreglo	☐ c) 6 espacios en un arreglo
☐ b) 4 espacios en un arreglo	☐ d) 7 espacios en un arreglo



12. El tipo de dato para una cadena es:

☐ a) String	☐ c) Int
☐ b) Char	☐ d) Float

13. Una cadena termina con el carácter:

☐ a) /n	☐ c) /0
☐ b) /t	☐ d) /s

14. Para inicializar un carácter se usa:

☐ a) #	☐ c) ´
☐ b) “	☐ d) (

15. Para inicializar una cadena se utiliza la palabra:

☐ a) Nt	☐ c) Float
☐ b) Static	☐ d) String

16. El primer elemento de un arreglo se indica con el número:

☐ a) 1	☐ c) -1
☐ b) 0	☐ d) NULL

17. Un arreglo unidimensional es sinónimo de:

☐ a) Un vector	☐ c) Registro
☐ b) Una variable	☐ d) Índice

18. Para acceder al contenido de un arreglo unidimensional se necesita:

☐ a) Usar un índice	☐ c) Usar tres índices
☐ b) Usar dos índices	☐ d) Usar cuatro índices



19. Un arreglo permite:

☐ a) Almacenar un tipo de dato	☐ c) Almacenar tres tipos de datos
☐ b) Almacenar dos tipos de datos	☐ d) Almacenar varios tipos datos

20. Una estructura permite:

☐ a) Almacenar un tipo de dato	☐ c) Almacenar tres tipos de datos
☐ b) Almacenar dos tipos de datos	☐ d) Almacenar varios tipos datos

21. Una cadena termina con el carácter:

☐ a) Retorno de carro	☐ c) Nueva línea
☐ b) Tabulador	☐ d) Nulo

22. Para inicializar una cadena en un arreglo se usa:

☐ a) La comilla simple	☐ c) No se usan comillas
☐ b) La comilla doble	☐ d) Dos comillas simples

23. A los elementos de una estructura se les conoce como:

☐ a) Variables	☐ c) Constantes
☐ b) Miembros	☐ d) Índices

24. El carácter que separa la etiqueta de la estructura, con un elemento de la estructura es el:

☐ a) '.'	☐ c) '>'
☐ b) '*'	☐ d) '<'

25. El tamaño de un arreglo depende entre otras cosas de:

☐ a) El tipo de dato	☐ c) Las constantes usadas
☐ b) Las variables usadas	☐ d) El compilador



RESPUESTAS

EXAMEN DE AUTOEVALUACIÓN



En este apartado encontrarás las respuestas al examen por unidad.

UNIDAD 5	
I. Solución	
1. a	14. c
2. a	15. b
3. b	16. b
4. a	17. a
5. b	18. a
6. c	19. a
7. c	20. d
8. a	21. d
9. b	22. b
10. a	23. b
11. c	24. a
12. b	25. a
13. c	



UNIDAD 6

Manejo de apuntadores





OBJETIVO PARTICULAR

Utilizará apuntadores en aplicaciones con arreglos, estructuras y funciones y podrá hacer uso dinámico de la memoria.

TEMARIO DETALLADO (8 horas)

6. Manejo de apuntadores

6.1. Introducción a los apuntadores

6.2. Apuntadores y arreglos

6.3. Apuntadores y estructuras

6.4. Apuntadores y funciones

6.5. Manejo dinámico de memoria



ACTIVIDAD DIAGNÓSTICA

LO QUE SÉ



Adjuntar archivo.

Elabora una breve búsqueda acerca de los apuntadores, arreglos, estructuras, funciones y la memoria dinámica.



ACTIVIDADES DE APRENDIZAJE



Unidad 6, actividad inicial. *Adjuntar archivo.* A partir del estudio de la bibliografía específica sugerida, elabora un mapa conceptual u [organizador gráfico](#) con los temas de la unidad. Puedes auxiliarte de algunos programas como Mindjet [MindManager](#).

1. **Unidad 6, actividad 1. *Adjuntar archivo.*** Realiza un programa que incremente en uno la posición de memoria a la que apunta
2. **Unidad 6, actividad 2. *Adjuntar archivo.*** Realiza un programa que pase el valor de una variable entera a otra variable mediante el empleo de apuntadores.
3. **Unidad 6, actividad 3. *Adjuntar archivo.*** Elabora un programa que permita dar de alta y consultar los datos de un lote de autos usados, utiliza una estructura y apuntadores. Los datos a gestionar son: marca, año, kilometraje y precio.
4. **Unidad 6, actividad 4. *Adjuntar archivo.*** Elabora una función que sume dos números introducidos por el usuario, utiliza apuntadores.
5. **Unidad 6, actividad 5. *Adjuntar archivo.*** Elabora un programa que acepte el ingreso de cinco números, y que posteriormente imprima las direcciones de memoria de los valores correspondientes.
6. **Unidad 6, actividad complementaria. *Adjuntar archivo.*** A partir del estudio de la unidad, realiza la siguiente actividad, misma que encontraras en el foro de la asignatura. Cabe señalar que esta será colocada en el foro por tu asesor.



ACTIVIDAD INTEGRADORA

LO QUE APRENDÍ



Adjuntar archivo.

Elabora un programa que, por medio de funciones controladas por un menú, permita agregar o quitar valores de una lista enlazada.



CUESTIONARIO DE REFORZAMIENTO



Adjuntar archivo. Responde las siguientes preguntas.

1. ¿Qué contiene un apuntador?
2. ¿Qué es la memoria principal?
3. ¿La memoria dinámica es sinónimo de *heap*?
4. ¿Cuál es el operador de dirección?
5. ¿Cuál es el operador de indirección?
6. ¿Cómo se declara un apuntador?
7. ¿Cuál es la ventaja de usar apuntadores con funciones?
8. ¿Cuál es la ventaja de usar apuntadores con arreglos?
9. ¿Cuál es la ventaja de usar apuntadores con estructuras?
10. ¿Qué es la aritmética de apuntadores?



EXAMEN PARCIAL

(Autoevaluación)



I. En el espacio en blanco escribe el inciso que complete la oración.

a) aritméticos	b) comparación	e) nulo
c) & ó ampersand / * ó asterisco	d) variable / memoria	

- ___ 1. Un apuntador es una que contiene una dirección de otra variable.
- ___ 2. Los apuntadores siempre deben declararse, cuando éste no apunte a ninguna posición válida, ha de asignársele un valor
- ___ 3. El carácter devuelve la dirección de memoria de su operando y devuelve el contenido de la variable.
- ___ 4. La suma y la resta son operadores que se pueden usar como punteros.
- ___ 5. Se puede utilizar la para saber qué posición de memoria es más alta.



II. Selecciona la respuesta correcta.

1. El * devuelve:

☐ a) El contenido del operando	☐ c) Un tipo de dato
☐ b) La dirección de memoria de su operando	☐ d) Una variable

2. El & devuelve:

☐ a) El contenido del operando	☐ c) Un tipo de dato
☐ b) La dirección de memoria de su operando	☐ d) Una variable

3. Un arreglo que se declara como apuntador debe ir antecediendo por el caracter:

☐ a) &&	☐ c) *
☐ b) ->	☐ d) &

4. Para leer un número en un arreglo que usa apuntadores, se usa:

☐ a) &	☐ c) *
☐ b) %	☐ d) &&

5. Solo se necesita esta biblioteca para usar los apuntadores:

☐ a) String.h	☐ c) Conio.h
☐ b) Stdio.h	☐ d) Strdlib.h

6. Para acceder a los miembros de una estructura se usa el:

☐ a) '&'	☐ c) '->'
☐ b) '*'	☐ d) '<-'



7. Podemos definir a una estructura como un conjunto de:

☐ a) Variables de tipos distintos	☐ c) Variables del mismo tipo
☐ b) Variables	☐ d) Variables ordenadas

8. La palabra reservada para definir una estructura es:

☐ a) Define	☐ c) Array
☐ b) Struct	☐ d) Char

9. Antes de definir una estructura se utiliza la palabra:

☐ a) Define	☐ c) Typedef
☐ b) Struct	☐ d) Int

10. Para desreferenciar a un apuntador se usa el:

☐ a) '&'	☐ c) '->'
☐ b) '*'	☐ d) '<-'

11. C pasa los parámetros de las funciones por:

☐ a) Valor	☐ c) Apuntadores
☐ b) Referencia	☐ d) Constantes

12. Si se utilizan apuntadores, los parámetros de las funciones se pasan por:

☐ a) Valor	☐ c) Apuntadores
☐ b) Referencia	☐ d) Constantes

13. Si los parámetros de una función son apuntadores, se utiliza el signo de:

☐ a) '&'	☐ c) '->'
☐ b) '*'	☐ d) '<-'



14. El uso de apuntadores hace que un algoritmo de ordenación sea:

☐ a) Independientemente del tipo de dato	☐ c) Más lento
☐ b) Más rápido	☐ d) Más eficiente

15. Un apuntador es:

☐ a) Una variable que contiene una dirección de memoria	☐ c) Una variable
☐ b) Una dirección de memoria	☐ d) El signo de *

16. Para determinar el tamaño en bytes de un tipo de dato se usa:

☐ a) Free	☐ c) Get
☐ b) Sizeof	☐ d) Put

17. El * se usa para:

☐ a) Acceder a una región en disco	☐ c) Acceder al contenido de una variable
☐ b) Acceder a la dirección de memoria de una variable	☐ d) Acceder a una región en memoria

18. El & se utiliza para acceder a:

☐ a) Una región en disco	☐ c) El contenido de una variable
☐ b) La dirección de memoria de una variable	☐ d) Una región en memoria

19. Para asignar un espacio de memoria se usa la función:

☐ a) Free	☐ c) Get
☐ b) Malloc	☐ d) Put



20. Para liberar un espacio de memoria se usa:

☐ a) Free	☐ c) Get
☐ b) Malloc	☐ d) Typedef

21. Para incrementar una posición de memoria se puede usar:

☐ a) ->	☐ c) >
☐ b) ++	☐ d) *

22. Para comparar dos apuntadores se usa el operador:

☐ a) &	☐ c) *
☐ b) >	☐ d) _>

23. Para inicializar un apuntador se puede usar:

☐ a) NULL	☐ c) *
☐ b) &	☐ d) ->

24. El único entero que puede asignarse a un apuntador es:

☐ a) Sólo son tipos enteros	☐ c) Sólo son de tipo flotante
☐ b) Sólo son de tipo carácter	☐ d) Pueden no tener n tipo definido



RESPUESTAS

EXAMEN DE AUTOEVALUACIÓN



En este apartado encontrarás las respuestas al examen por unidad.

UNIDAD 6	
I. Solución	
1. d	
2. e	
3. c	
4. a	
5. b	

UNIDAD 6	
II. Solución	
1. a	13. b
2. b	14. a
3. c	15. a
4. a	16. b
5. b	17. c
6. c	18. b
7. a	19. b
8. b	20. a
9. c	21. b
10. b	22. b
11. a	23. a
12. b	24. d

Plan 2012

2016
actualizado

