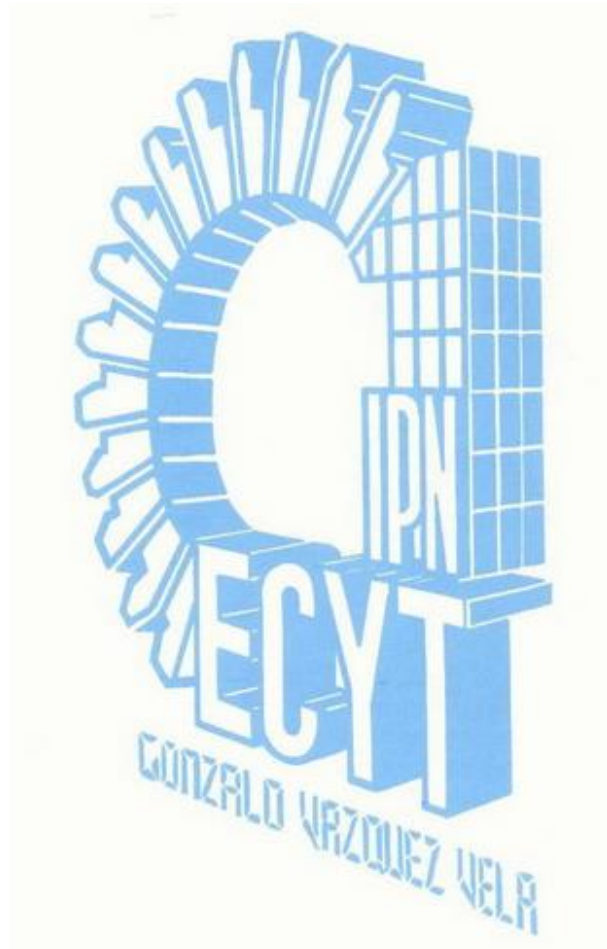




**Centro de Estudios Científicos y
Tecnológicos No. 1
"Gonzalo Vázquez Vela"**



Nombre	_____
Grupo	_____
Profesor	_____

LENGUAJE DE PROGRAMACIÓN

Cuadernillo de Prácticas

Academia de Sistemas Digitales
Turno Vespertino

M. en C. Ranulfo Dimitri Cab Cordero

Tabla de contenido

Red de Competencias (General y Particulares)	I
Programa Sintético	III
Práctica 1 Conceptos Básicos	1
Práctica 2 Como programar en el lenguaje	5
Práctica 3 Operadores y Expresiones	7
Práctica 4 Sentencias de Control Secuenciales	9
Práctica 5 Sentencias de Control de Decisión	11
Práctica 6 Sentencias de Control Anidadas	13
Práctica 7 Sentencias de Control por Casos	15
Práctica 8 Sentencias de Control: Condicionamiento Alto	17
Práctica 9 Sentencias de Control: Condicionamiento Bajo	19
Práctica 10 Arreglos Unidimensionales y Multidimensionales	21
Práctica 11 Cadenas	27
Práctica 12 Funciones Matemáticas y Gráficas	29
Práctica 13 Funciones de Manejo de Puertos	33
Práctica 14 Funciones sin Parámetros	37
Práctica 15 Funciones con Parámetros	39
Código ASCII	41
Referencias Documentales	43



CARRERA: TÉCNICO EN SISTEMAS DIGITALES

UNIDAD DE APRENDIZAJE: LENGUAJE DE PROGRAMACIÓN

RED DE COMPETENCIAS (GENERAL Y PARTICULARES)

Desarrolla programas en un lenguaje de alto nivel que resuelven problemáticas planteadas relacionadas con su entorno, empleando las herramientas propias del lenguaje, estructuras de control y diseñando funciones especializadas.

COMPETENCIA PARTICULAR 1

Realiza programas con funciones básicas de entrada – salida de datos, declarando variables de diversos tipos y realizando operaciones matemáticas simples basándose en algoritmos y utilizando el entorno integrado del lenguaje.

RAP 1:
Conoce el concepto de programa y cuáles son sus elementos, así como su estructura general.

RAP 2:
Aplica las funciones básicas de entrada y salida de datos.

RAP 3:
Emplea los distintos operadores disponibles en el lenguaje aplicando sus reglas de prioridad.

COMPETENCIA PARTICULAR 2

Diseña programas empleando las estructuras de control condicionales y repetitivas basadas en el modelo de la programación estructurada.

RAP 1:
Conoce los principios de la programación estructurada, así como las sentencias de control secuenciales.

RAP 2:
Aplica las distintas estructuras de control condicional simples y anidadas.

RAP 3:
Emplea las diversas estructuras de control repetitivas simples y anidadas.

COMPETENCIA PARTICULAR 3

Realiza programas que implementan arreglos como herramienta de solución para resolver las problemáticas planteadas.

RAP 1:
Utiliza los arreglos como herramienta auxiliar en la simplificación de variables del mismo tipo.

RAP 2:
Emplea los arreglos como herramienta para implementar cadenas de caracteres.

COMPETENCIA PARTICULAR 4

Desarrolla programas utilizando funciones matemáticas, gráficas y de entrada – salida de datos mediante puerto, así como las diseñadas para uso específico de la problemática planteada.

RAP 1:
Utiliza las funciones predefinidas en el lenguaje.

RAP 2:
Diseña funciones específicas para resolver la problemática planteada.





CARRERA: TÉCNICO EN SISTEMAS DIGITALES

UNIDAD DE APRENDIZAJE: LENGUAJE DE PROGRAMACIÓN

PROGRAMA SINTÉTICO

COMPETENCIA GENERAL:

Desarrolla programas en un lenguaje de alto nivel que resuelven problemáticas planteadas relacionadas con su entorno, empleando las herramientas propias del lenguaje, estructuras de control y diseñando funciones especializadas.

COMPETENCIA PARTICULAR

COMPETENCIA PARTICULAR 1

Realiza programas con funciones básicas de entrada – salida de datos, declarando variables de diversos tipos y realizando operaciones matemáticas simples basándose en algoritmos y utilizando el entorno integrado del lenguaje.

RAP

RAP 1:

Conoce el concepto de programa y cuáles son sus elementos, así como su estructura general.

RAP 2:

Aplica las funciones básicas de entrada y salida de datos.

RAP 3:

Emplea los distintos operadores disponibles en el lenguaje aplicando sus reglas de prioridad.

CONTENIDOS

CONCEPTUALES

- Elementos de un programa
 - Datos y tipos de datos
 - Datos numéricos
 - Datos tipo carácter y cadena
 - Datos lógicos
 - Identificadores, constantes y variables
- Palabras reservadas
- Declaraciones
- Conversión de tipos
- Comentarios
- Instrucciones
- Estructura de un programa en el lenguaje
- Funciones de entrada y salida de datos
 - Salida de datos
 - Caracteres de conversión
 - Banderas
 - Entrada de datos
 - Caracteres de conversión
 - Operadores
 - De asignación
 - Aritméticos
 - De asignación compuesta
 - De incremento y decremento
 - Relacionales
 - Operadores lógicos
 - Con operandos
 - Con expresión

PROCEDIMENTALES

- Práctica 1: Conceptos básicos
- Práctica 2: Cómo programar en el lenguaje
- Práctica 3: Operadores y expresiones



CARRERA: TÉCNICO EN SISTEMAS DIGITALES

UNIDAD DE APRENDIZAJE: LENGUAJE DE PROGRAMACIÓN

<p>COMPETENCIA PARTICULAR 2 Diseña programas empleando las estructuras de control condicionales y repetitivas basadas en el modelo de la programación estructurada.</p>	<p>RAP 1: Conoce los principios de la programación estructurada, así como las sentencias de control secuenciales.</p> <p>RAP 2: Aplica las distintas estructuras de control condicional simples y anidadas.</p> <p>RAP 3: Emplea las diversas estructuras de control repetitivas simples y anidadas.</p>	<p>CONCEPTUALES</p> <ul style="list-style-type: none"> • Modelo de la programación estructurada <ul style="list-style-type: none"> • Reglas • Técnicas • Sentencias de control <ul style="list-style-type: none"> • Secuenciales • Condicionales • Repetitivas <ul style="list-style-type: none"> • Condicionamiento alto • Condicionamiento bajo • Anidación <p>PROCEDIMENTALES</p> <p>Práctica 4: Sentencias de control secuenciales Práctica 5: Sentencias de control de decisión Práctica 6: Sentencias de control anidadas Práctica 7: Sentencias de control por casos Práctica 8: Sentencias de control: condicionamiento alto Práctica 8: Sentencias de control: condicionamiento bajo</p>
<p>COMPETENCIA PARTICULAR 3 Realiza programas que implementan arreglos como herramienta de solución para resolver las problemáticas planteadas.</p>	<p>RAP 1: Utiliza los arreglos como herramienta auxiliar en la simplificación de variables del mismo tipo.</p> <p>RAP 2: Emplea los arreglos como herramienta para implementar cadenas de caracteres.</p>	<p>CONCEPTUALES</p> <ul style="list-style-type: none"> • Definición • Declaración • Inicialización • Tipos de arreglos <ul style="list-style-type: none"> • Unidimensionales • Multidimensionales • Acceso a datos <ul style="list-style-type: none"> • Directo • Automatizado • Cadenas <ul style="list-style-type: none"> • Representación • Operaciones básicas <p>PROCEDIMENTALES</p> <p>Práctica 10: Arreglos unidimensionales y multidimensionales Práctica 11: Cadenas</p>





CARRERA: TÉCNICO EN SISTEMAS DIGITALES

COMPETENCIA PARTICULAR 4

Desarrolla programas utilizando funciones matemáticas, gráficas y de entrada – salida de datos mediante puerto, así como las diseñadas para uso específico de la problemática planteada.

RAP 1:

Utiliza las funciones predefinidas en el lenguaje.

RAP 2:

Diseña funciones específicas para resolver la problemática planteada.

UNIDAD DE APRENDIZAJE: LENGUAJE DE PROGRAMACIÓN

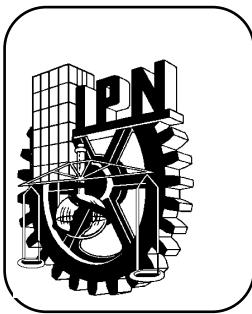
CONCEPTUALES

- Funciones predefinidas en el lenguaje de programación
- Definición
- Declaración y llamada
 - Matemáticas
 - Gráficas
 - Entrada – salida de puerto
- Funciones definidas por el programador
 - Declaración
 - Prototipo
 - Cuerpo
- Tipo de función
 - Con Parámetros , sin Parámetros
 - No regresan valor, regresan valor
 - Llamadas

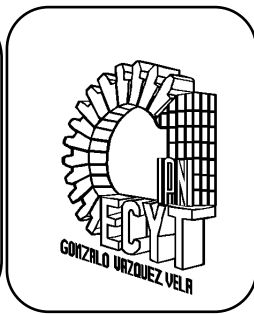
PROCEDIMENTALES

- Practica 12: Funciones matemáticas y graficas
- Practica 13: Funciones manejo de puertos
- Practica 14: Funciones sin parámetros
- Practica 15: Funciones con parámetros





INSTITUTO POLITÉCNICO NACIONAL
 Centro de Estudios Científicos y Tecnológicos N° 1
 “Gonzalo Vázquez Vela”
 Academia de Sistemas Digitales T. V.
 Prácticas: Lenguaje de Programación



NOMBRE DEL ALUMNO: _____

 Apellido Paterno

 Apellido Materno

 Nombre

N° DE BOLETA: _____ GRUPO: _____

ASIGNATURA: **Lenguaje de Programación**

HOJA	DE	FECHA	EVALUACION
1	4		

Profesor Titular: _____
 Profesor Adjunto: _____

Práctica 1 Conceptos Básicos

Objetivo de la Unidad: UNIDAD 1

Realizar programas con funciones básicas de entrada – salida de datos, declarando variables de diversos tipos y realizando operaciones matemáticas simples basándose en algoritmos y utilizando el entorno integrado del lenguaje.

Resultado de Aprendizaje Propuesto (RAP):

Asociado con esta Práctica: (1.1)	Conoce el concepto de programa y cuáles son sus elementos, así como su estructura general.
--	--

Objetivos de la Práctica:

1. Manejar el ambiente de C por medio del uso del ambiente.

Equipo Necesario

1. PC
2. Borland C
3. Cañón de proyección

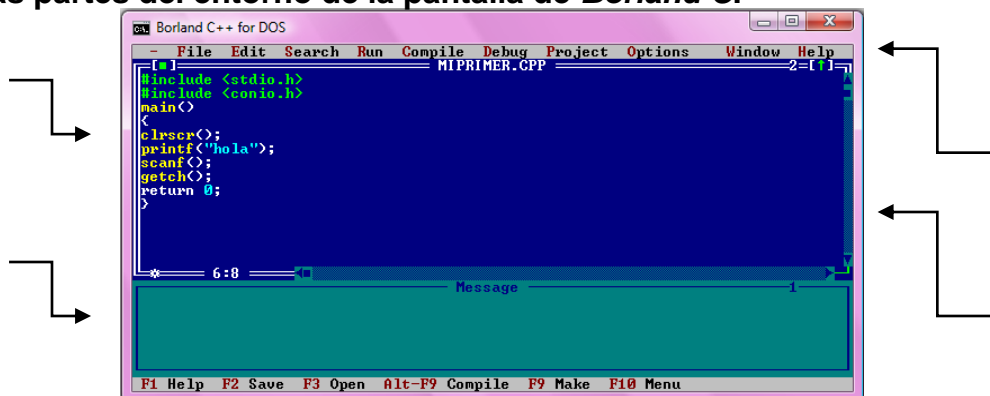
Material

1. Apuntes
2. Información Impresa relacionada

Instrucciones:

Siga las instrucciones del maestro en su exposición y responda lo que se le indica.

a) Escribe las partes del entorno de la pantalla de **Borland C.**



b) Describe cada Menú.

- 1 File: _____
- 2 Edit: _____
- 3 Search: _____
- 4 Run: _____
- 5 Compile: _____
- 6 Debug: _____
- 7 Project: _____
- 8 Options: _____
- 9 Window: _____
- 10 Help: _____

c) Describe los submenús de FILE.

- 1 New: _____
- 2 Open: _____
- 3 Save: _____
- 4 Save As: _____
- 5 Save All: _____
- 6 Change Dir: _____
- 7 Print: _____
- 8 Dos Shell: _____
- 9 Quit: _____

d) Describe los submenús de EDIT.

- 1 Undo: _____
- 2 Redo: _____
- 3 Cut: _____
- 4 Copy: _____
- 5 Paste: _____
- 6 Clear: _____
- 7 Copy Example _____
- 8 Clipboard: _____

e) Describe los submenús de SEARCH.

- 1 Find: _____
- 2 Replace: _____
- 3 Search Again: _____
- 4 Go to Line Number _____
- 5 Previous Error: _____
- 6 Next Error: _____
- 7 Locate Function _____

f) Describe los submenús de RUN.

- 1 Run: _____
- 2 Program Reset:
 - a. . _____
 - b. . _____
 - c. . _____
- 3 Go to Cursor: _____
- 4 Trace Into: _____
- 5 Step Over: _____

g) Describe los submenús de COMPILE.

- 1 Compile _____
- 2 Make: _____
- 3 Link: _____
- 4 Remove Message: _____

h) ¿Para qué sirve el menú DEBUG?

i) ¿Para qué sirve el menú PROJECT?

j) ¿Para qué sirve el menú OPTIONS?

k) ¿Para qué sirve el menú WINDOW?

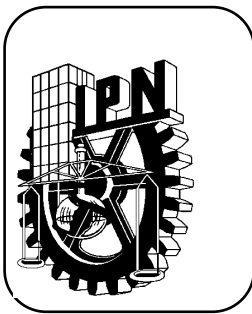
l) ¿Para qué sirve el menú HELP?

m) Escribe la función para cada tecla.

F1		F10	
F2		ALT+ X	
F3		ALT+ F9	
F4		CTRL+ F1	
F5		CTRL+ F9	
F6		CTRL+ INSERT	
F7		SHIFT+ SUPR	
F8		SHIFT+ INSERT	
F9			

Conclusiones:

1. ¿Qué diferencia hay entre un lenguaje de alto nivel y otro de bajo nivel?
2. ¿Cuándo se dice que se está programando en lenguaje de Maquina?
3. Menciona 5 lenguajes de alto nivel.
4. ¿Qué significa compilar un programa?
5. Dibuja los símbolos usados en los diagramas de flujo
6. Codifica el programa que aparece en la imagen anterior y observa lo que sucede, anota tus conclusiones.



INSTITUTO POLITÉCNICO NACIONAL
 Centro de Estudios Científicos y Tecnológicos N° 1
 “Gonzalo Vázquez Vela”
 Academia de Sistemas Digitales T. V.
Prácticas: Lenguaje de Programación



NOMBRE DEL ALUMNO: _____
 _____ Apellido Paterno
 _____ Apellido Paterno Nombre
 N° DE BOLETA _____ GRUPO: _____

ASIGNATURA: **Lenguaje de Programación**

HOJA	DE	FECHA	EVALUACION
1	2		

Profesor Titular: _____
 Profesor Adjunto: _____

Práctica 2

Como programar en el lenguaje

Objetivo de la unidad: UNIDAD 1

Realizar programas con funciones básicas de entrada – salida de datos, declarando variables de diversos tipos y realizando operaciones matemáticas simples basándose en algoritmos y utilizando el entorno integrado del lenguaje.

Resultado de Aprendizaje Propuesto (RAP):

Asociado con esta Práctica: (1.2)	Aplica las funciones básicas de entrada y salida de datos.
--	--

Objetivos de la Práctica:

2. El alumno desarrollara programas básicos en C

Equipo necesario: <ol style="list-style-type: none"> 1. PC 2. Borland C 3. Cañón de proyección 	Material: <ol style="list-style-type: none"> 1. Apuntes 2. Información impresa relacionada
--	---

Cuestionario Previo:

1. Escribe las partes básicas de un programa en lenguaje C
2. ¿Cómo se llama la función principal de un programa en C?
3. Escribe las instrucciones necesarias que correspondan a las siguientes acciones:
 - a. Limpiar pantalla: _____
 - b. Poner un texto en pantalla: _____
 - c. Limpiar pantalla: _____
 - d. Detener la ejecución de un programa: _____
 - e. Incluir librerías: _____

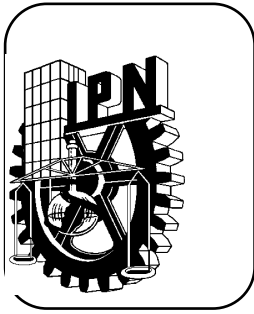
4. Menciona 5 librerías básicas de Lenguaje C.
5. Menciona algunas secuencias de escape
6. Menciona los operadores aritméticos de C
7. ¿Qué diferencias existe entre una variable local y una global?
8. Escribe la instrucción para limpiar pantalla.
9. Escribe la sintaxis de la instrucción *printf*
10. ¿Cuál es la instrucción de inclusión de librerías en un programa en C?

Actividades Prácticas:

1. Diseña un programa que de cómo salida una línea de texto
2. Realiza un programa que de cómo salida un texto con varias líneas en lenguaje C.
3. Implementa un programa en lenguaje C que determine la corriente y la potencia de un circuito a partir de su corriente y voltaje.
4. Realiza un programa que determine la resistencia total de un circuito de dos resistores en paralelo.
5. Implementa un programa que realice la conversión entre grados centígrados y grados Fahrenheit.
6. Realiza el programa que determine el promedio de 5 números.
7. Haz un programa que determine el cubo de un número ingresado por el usuario
8. Realiza un programa que determine en mili Amperes de un valor ingresado en Amperes.
9. Diseña un programa que lea un nombre y un teléfono y los imprima.
10. Usando *printf* realiza un programa que muestre el calendario del mes de agosto.

Conclusiones:

1. Describe como se puede usar la ayuda de *Borland C*
2. De los programas implementados realiza:
 - a. Diagrama de flujo
 - b. Codificación
 - c. Prueba de escritorio



INSTITUTO POLITÉCNICO NACIONAL
Centro de Estudios Científicos y Tecnológicos N° 1
“Gonzalo Vázquez Vela”
Academia de Sistemas Digitales T. V.
Prácticas: Lenguaje de Programación



NOMBRE DEL ALUMNO: _____
Apellido Paterno

Apellido Paterno Nombre

N° DE BOLETA _____ GRUPO: _____

ASIGNATURA: **Lenguaje de Programación**

HOJA	DE	FECHA	EVALUACION
1	2		

Profesor Titular: _____
 Profesor Adjunto: _____

Práctica 3 Operadores y Expresiones

Objetivo de la unidad: UNIDAD I

Realizar programas con funciones básicas de entrada – salida de datos, declarando variables de diversos tipos y realizando operaciones matemáticas simples basándose en algoritmos y utilizando el entorno integrado del lenguaje.

Resultado de Aprendizaje Propuesto (RAP):

Asociado con esta Práctica: (1.3)	Emplea los distintos operadores disponibles en el lenguaje aplicando sus reglas de prioridad.
--	---

Objetivos de la Práctica:

3. El alumno desarrollara programas básicos en C, aplicando los diferentes tipos de datos y variables.

Equipo Necesario <ol style="list-style-type: none"> 1. PC 2. BorlandC 3. Cañón de proyección 	Material <ol style="list-style-type: none"> 1. Apuntes 2. Información impresa relacionada
--	--

Cuestionario Previo

1. Escriba las funciones de una librería *math.h*.

2. ¿Cuál es la función para calcular la raíz cuadrada?

3. ¿Cuál es la función para calcular el senoidal?

4. Menciona en que librería se encuentra cada una de las siguientes instrucciones:

- a. *printf* _____
- b. *scanf* _____
- c. *getch()* _____
- d. *clrscr()* _____
- e. *return* _____
- f. *sin()* _____

5. Explica que función realizan las siguientes funciones:

- a. *floor* _____
- b. *fabs* _____
- c. *sinh* _____
- d. *log()* _____
- e. *pow10* _____

6. Enumere los operadores aritméticos de C.

7. Encuentre los valores que daría “C” para las siguientes expresiones:

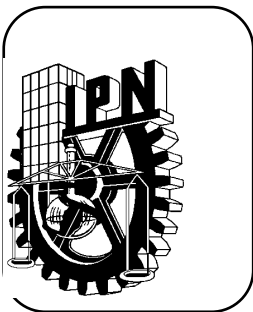
- a. $(4+3-5)\%2+6*5/2-7=$ _____
- b. $20/10+5*6\%3=$ _____
- c. $(4*6-3+1)-(4*2-4)+6+2*3=$ _____
- d. $6+1-7+8\%3\%2+1=$ _____
- e. $100*(20+5)+3\%10=$ _____

Actividades Prácticas

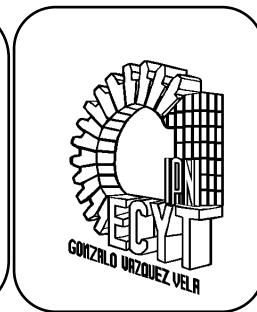
1. Realiza un programa que te permita resolver una ecuación de segundo grado a partir del ingreso de sus coeficientes *a*, *b* y *c* (use la fórmula general).
2. Realizar un programa que encuentre los valores de “*x*” y “*y*” para un sistema de ecuaciones de segundo grado y en donde el usuario ingresará los coeficientes *a1*, *b1* y *c1*, así como *a2*, *b2* y *c2*.(use método de determinantes).
3. Haga un programa que determine el valor de la hipotenusa y los ángulos de un triángulo rectángulo (sug. Use teorema de Pitágoras, funciones trigonométrías y leyes de los ángulos de los triangulo)

Conclusiones

1. De los programas implementados realiza:
 - a. Diagrama de flujo
 - b. Codificación
 - c. Prueba de escritorio.
2. Utilizando los operadores aritméticos y la librería *math.h*, escriba la sintaxis de las expresiones matemáticas.



INSTITUTO POLITÉCNICO NACIONAL
 Centro de Estudios Científicos y Tecnológicos N° 1
 “Gonzalo Vázquez Vela”
 Academia de Sistemas Digitales T. V.
 Prácticas: Lenguaje de Programación



NOMBRE DEL ALUMNO: _____
 _____ Apellido Paterno
 _____ Apellido Materno _____ Nombre
 N° DE BOLETA _____ GRUPO: _____

ASIGNATURA: **Lenguaje de Programación**

HOJA	DE	FECHA		EVALUACION
1	2			

Profesor Titular: _____
 Profesor Adjunto: _____

Práctica 4 Sentencias de Control Secuenciales

Objetivo de la Unidad: UNIDAD II

Diseña programas empleando las estructuras de control condicionales y repetitivas basadas en el modelo de la programación estructurada.

Resultado de Aprendizaje Propuesto (RAP):

Asociado con esta Práctica: (2.1)

Conoce los principios de la programación estructurada, así como las sentencias de control secuenciales.

Objetivos de la Práctica:

1. Identificar la sintaxis para la creación de ciclos en Lenguaje C
2. Diseñar programas básicos que presenten diferentes opciones al usuario

Equipo Necesario 1. PC 2. Borland C 3. Cañón de proyección	Material 1. Apuntes 2. Información Impresa relacionada
--	---

Cuestionario Previo

1. ¿Cuál es la sintaxis de la sentencia *for*?

2. ¿Por qué es importante la sentencia *for* para la implementación de contadores?

3. Escribe como se escriben en C las siguientes expresiones
 - a. $A=A+3$ _____
 - b. $A+1$ _____
 - c. $Z=Z-5$ _____
 - d. $C-1$ _____
 - e. $X= X*3$ _____

4. ¿A qué se le conoce como *foranidados*?

Actividades Prácticas

Implementa en laboratorio los programas que se te indiquen incluyendo diagrama de flujo y prueba de escritorio. Ya una vez probados deberán tener recurrencia.

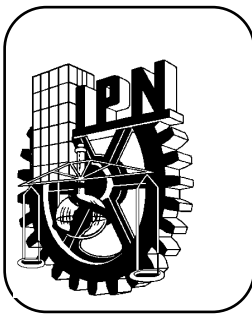
1. Realiza un programa que de cómo salda un contador ascendente en el que el usuario ingrese el valor inicial, valor final e intervalo de conteo.
2. Realiza un programa que despliegue las tablas de multiplicar
3. Realice un programa que valúe la función $y=2x^2 + 3x - 6$ en una rango de -10 a 10
4. Diseña un programa que imprima en pantalla la siguiente figura carácter por carácter.

```
*****  
  
*****  
  
*****  
  
*****
```

Modifíquelo para que se el usuario pueda cambiar la base y la altura de la figura.

5. Haga un programa que por medio de un menú de la opción de imprimir un contador ascendente o un contador descendente cuyos variables deberán ser ingresadas por el usuario.
6. Implemente un programa que calcula el factorial de n numero usando la sentencia *for*
7. Haga un programa que sea capaz de realizar la suma de N números ingresados por el usuario.
8. Use la función *random* para realizar un contador de 0 a un valor aleatorio.
9. Une la sentencia *for* para encontrar el valor de la función $f= \sum_{x=1}^{x=5} (x^2/x)$
10. Por ayuda de la sentencia *exit* hacer recurrente un programa con la sentencia *for*

Conclusiones:



INSTITUTO POLITÉCNICO NACIONAL
 Centro de Estudios Científicos y Tecnológicos N° 1
 “Gonzalo Vázquez Vela”
 Academia de Sistemas Digitales T. V.
 Prácticas: Lenguaje de Programación



NOMBRE DEL ALUMNO: _____
 Apellido Paterno _____
 Apellido Paterno _____ Nombre _____
 N° DE BOLETA _____ GRUPO: _____

ASIGNATURA: *Lenguaje de Programación*

HOJA	DE	FECHA		EVALUACION
1	2			

Profesor Titular: _____
 Profesor Adjunto: _____

Práctica 5

Sentencias de Control de Decisión

Objetivo de la Unidad: UNIDAD II

Diseña programas empleando las estructuras de control condicionales y repetitivas basadas en el modelo de la programación estructurada.

Resultado de Aprendizaje Propuesto (RAP):

Asociado con esta Práctica: (2.2)	Aplica las distintas estructuras de control de condicional simples y anidadas.
--	--

Objetivos de la Práctica:

- El alumno dará solución a diversos problemas mediante el uso la sentencia de control *IF*.

<p>Equipo Necesario</p> <ol style="list-style-type: none"> PC BorlandC Cañón de proyección 	<p>Material</p> <ol style="list-style-type: none"> Apuntes Información Impresa relacionada
--	---

Questionario previo

- Menciona que función tienen las sentencias de control dentro de un programa en lenguaje C
- Convierte mediante el uso de operadores condicionales las siguientes sentencias:
 - Si “a” es mayor que 5 _____
 - Si la suma de “a” y “b” es mayor o igual que 0 _____
 - Si “a” es mayor que “b” o bien “c” es igual a 5 _____
 - Si “X” es igual a “Y” _____
 - Si “d” es diferente de “t” _____
 - Si “a” es mayor a 5 pero menor a 200. _____
 - Si “z” es igual a 4 y “w” es mayor a 6 _____
 - Si el producto entre “a” y “b” es menor a la diferencia entre “c” y “b” _____
 - Si el módulo de 5 y 3 es igual o menor a 5 _____
 - Si “R” es mayor a 1000 _____

3. Describe la sintaxis de la sentencia de control “*if*”
4. Explica que aplicación tiene la sentencias “*else if*” con respecto a una sentencia “*if*”

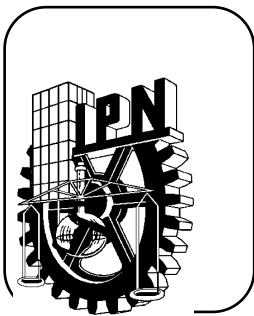
Actividades prácticas

Desarrolla los programas que se te indiquen en el laboratorio los cuales deberán ser guardados en extensión *c* y ejecutable.

1. Realice un programa que acepte dos valores: el de la calificación del extraordinario y el de la calificación de promedio final. La salida del programa será el valor más alto de los dos ingresados.
2. Haga un programa que capte tres valores “*a*”, “*b*”, y “*c*” La salida del programa deberá ser el valor menor de los tres ingresados.
3. Mediante un programa al cual se le ingresen 5 calificaciones, al salida deberá ser el promedio de dichos números si este es mayor o igual a 6 de lo contrario deberá dar como salida un mensaje que diga “*reprobado*”.
4. Realice un programa que determine la resistencia total de un circuito con dos resistores en paralelo, cuando uno o los dos valores ingresados sea “0” el programa deberá ingresar un mensaje de error. La salida del programa deberá ser en Ohms cuando el valor sea menor a 1000 y cuando sea mayor a 1000 la salida deberá ser en Kilo ohms
5. Implementa un programa que acepte del usuario un carácter y cuya salida será que si se trata de un carácter en mayúscula o minúscula.

Conclusiones:

1. De los programas implementados realiza:
 - a. Diagrama de flujo
 - b. Codificación
 - c. Prueba de escritorio.



INSTITUTO POLITÉCNICO NACIONAL
 Centro de Estudios Científicos y Tecnológicos N° 1
 “Gonzalo Vázquez Vela”
 Academia de Sistemas Digitales T. V.
 Prácticas: Lenguaje de Programación



NOMBRE DEL ALUMNO: _____
 Apellido Paterno _____
 Apellido Paterno _____ Nombre _____
 N° DE BOLETA _____ GRUPO: _____

ASIGNATURA: *Lenguaje de Programación*

HOJA	DE	FECHA		EVALUACION
1	2			

Profesor Titular: _____
 Profesor Adjunto: _____

Práctica 6 Sentencias de Control Anidadas

Objetivo de la Unidad: UNIDAD II

Diseña programas empleando las estructuras de control condicionales y repetitivas basadas en el modelo de la programación estructurada.

Resultado de Aprendizaje Propuesto (RAP):

**Asociado con esta
 Práctica: (2.2)**

Aplica las distintas estructuras de control de condicional simples y anidadas.

Objetivos de la Práctica:

- El alumno dará solución a diversos problemas mediante el uso la sentencia de control *IF*.

Equipo Necesario

- PC
- BorlandC
- Cañón de proyección

Material

- Apuntes
- Información Impresa relacionada

Questionario previo

- Menciona por medio de que se hace un “anidamiento de sentencias *if*”

Actividades prácticas

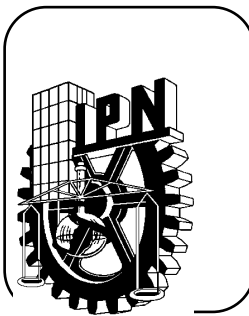
Desarrolla los programas que se te indiquen en el laboratorio los cuales deberán ser guardados en extensión c y ejecutable.

- Haga un programa que determine el total a pagar a partir del número de artículos comprados, cada artículo tiene un valor de \$12.85, Si la compra total es mayor \$1000, el programa deberá de aplicar un 10% de descuento al total.
- Diseña un programa que determina si un número ingresado por el usuario es divisible entre 5 y entre 3.
- Realiza un programa que al ingresar un 1 deberá dar como salida un texto relacionado con información personal de uno de los integrantes y para cualquier otro número los datos relacionados con el otro integrante del equipo.

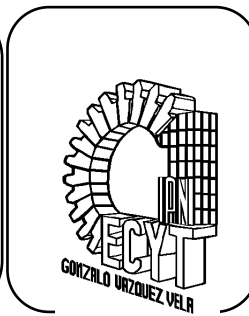
4. Haga un programa que imprima en la pantalla un texto relacionado con la sentencia *If*, pero solo cuando el usuario ingrese el código “2020” de lo contrario deberá mandar un mensaje de error “número no valido”
5. Implemente un programa que imprima en pantalla el sueldo total de un trabajador; el usuario deberá de ingresar el número de horas trabajadas, seguidas de la letra “a” si se trata de un trabajador cuyo sueldo es de \$16 la hora y pulsar otra tecla para el caso en el que el sueldo sea de \$14.50

Conclusiones:

1. De los programas implementados realiza:
 - a. Diagrama de flujo
 - b. Codificación
 - c. Prueba de escritorio.



INSTITUTO POLITÉCNICO NACIONAL
 Centro de Estudios Científicos y Tecnológicos N° 1
 “Gonzalo Vázquez Vela”
 Academia de Sistemas Digitales T. V.
 Prácticas: Lenguaje de Programación



NOMBRE DEL ALUMNO: _____
 Apellido Paterno _____
 Apellido Materno _____
 Nombre _____
 N° DE BOLETA _____ GRUPO: _____

ASIGNATURA: **Lenguaje de Programación**

HOJA	DE	FECHA	EVALUACION
1	2		

Profesor Titular: _____
 Profesor Adjunto: _____

Práctica 7

Sentencias de Control por Casos

Objetivo de la Unidad: UNIDAD II

Diseña programas empleando las estructuras de control condicionales y repetitivas basadas en el modelo de la programación estructurada.

Resultado de Aprendizaje Propuesto (RAP):

Asociado con esta Práctica: (2.2)	Aplica las distintas estructuras de control de condicional simples y anidadas.
--	--

Objetivos de la Práctica:

- 1 Identificar la sintaxis para la creación de menús en Lenguaje C
- 2 Diseñar programas básicos que presenten diferentes opciones al usuario

<p>Equipo Necesario</p> <ol style="list-style-type: none"> 1. PC 2. BorlandC 3. Cañón de proyección 	<p>Material</p> <ol style="list-style-type: none"> 1. Apuntes 2. Información Impresa relacionada
---	---

Cuestionario previo

1. ¿Explica la función de la sentencia *switch*?
2. ¿Cuál es la sintaxis de la sentencia *switch*?
3. ¿Qué aplicaciones le darías a la sentencia *switch*?

Actividades prácticas

1. Busca en la ayuda de *BorlandC* un programa de ejemplo de la sentencia *switch*. y ejecútalo.
 - a. 1a.- Modifícalo con el fin de aumentar una opción.
 - b. 1b.- Modifícalo con el fin de cambiar las letras necesarias para acceder a cada opción.

2. Implementa los programas que se te indican.
 - a. Realiza un programa que tenga un opción para que el usuario seleccione las unidades de aprendizaje que se imparten de tercero a sexto semestre de la carrera.
 - b. Haga un programa que de la opción de determinar:
 - i. La resistencia total de un circuito con tres resistores en serie al oprimir una “s”
 - ii. La resistencia total de un circuito de tres resistores en paralelo al oprimir una “p”
 - iii. Un mensaje de error para cualquier otra opción no valida.
 - c. Realiza un programa de de la opción de calcular cualquiera de la tres variables de la ley de ohm.
 - d. Implementa un programa cuya función sea calcular de °F a °C o bien de °C a °F.
 - e. Realiza por medio de un menú un programa que de la opción de realizar alguna las operaciones aritméticas básicas entre dos números.

Conclusiones de la práctica:

1. Elabora los diagramas de flujo y registra la codificación de los programas realizados.



INSTITUTO POLITÉCNICO NACIONAL
 Centro de Estudios Científicos y Tecnológicos N° 1
 “Gonzalo Vázquez Vela”
 Academia de Sistemas Digitales T. V.
 Prácticas: Lenguaje de Programación



NOMBRE DEL ALUMNO: _____

 Apellido Paterno

 Apellido Materno

 Nombre

N° DE BOLETA _____ GRUPO _____

ASIGNATURA: *Lenguaje de Programación*

HOJA	DE	FECHA		EVALUACION
1	2			

Profesor Titular: _____
 Profesor Adjunto: _____

Práctica 8

Sentencias de Control: Condicionamiento Alto

Objetivo de la Unidad: UNIDAD II

Diseña programas empleando las estructuras de control condicionales y repetitivas basadas en el modelo de la programación estructurada.

Resultado de Aprendizaje Propuesto (RAP):

Asociado con esta Práctica: (2.3)	Emplea las diversas estructuras de control repetitivas
--	--

Objetivos de la Práctica:

1. La sintaxis para la creación de menús en *Lenguaje C*
2. Diseñar programas básicos que presenten diferentes opciones al usuario

<p>Equipo Necesario</p> <ol style="list-style-type: none"> 1. PC 2. <i>BorlandC</i> 3. Cañón de proyección 	<p>Material</p> <ol style="list-style-type: none"> 1. Apuntes 2. Información Impresa relacionada
--	---

Questionario previo

1. ¿Qué es una sentencia *while*?
2. Escribe la sintaxis de la sentencia *while*:
3. ¿Qué pasa cuando se ejecuta una sentencia *while*?

Actividades prácticas

1. Realiza un programa que despliega todas las tablas de multiplicar.
2. Realiza un programa que calcula la suma de 1 hasta n números.
3. Realiza un programa que calcula el factorial de un número.
4. Realiza un programa que muestre los números pares del 0 al 100.
5. Realiza un programa que solicite una

Conclusiones:

1. Realiza el diagrama de flujo, codificación y prueba de escritorio de los programas realizados.



INSTITUTO POLITÉCNICO NACIONAL
 Centro de Estudios Científicos y Tecnológicos N° 1
 “Gonzalo Vázquez Vela”
 Academia de Sistemas Digitales T.V.
 Prácticas: *Introducción al Lenguaje de Programación*



NOMBRE DEL ALUMNO: _____

 Apellido Paterno

 Apellido Materno

 Nombre

N° DE BOLETA _____ GRUPO _____

ASIGNATURA: *Lenguaje de Programación*

HOJA	DE	FECHA		EVALUACION
1	2			

Profesor Titular: _____
 Profesor Adjunto: _____

Práctica 9

Sentencias de Control: Condicionamiento Bajo

Objetivo de la Unidad: UNIDAD II

Diseña programas empleando las estructuras de control condicionales y repetitivas basadas en el modelo de la programación estructurada.

Resultado de Aprendizaje Propuesto (RAP):

Asociado con esta Práctica: (2.3)	Emplea las diversas estructuras de control repetitivas
--	--

Objetivos de la Práctica:

1. La sintaxis para la creación de menús en *Lenguaje C*
2. Diseñar programas básicos que presenten diferentes opciones al usuario

Equipo Necesario <ol style="list-style-type: none"> 1. PC 2. <i>BorlandC</i> 3. Cañón de proyección 	Material <ol style="list-style-type: none"> 1. Apuntes 2. Información Impresa relacionada
---	--

Questionario previo

1. Explica la sentencia *do while*?

2. Escribe la sintaxis de la sentencia *do while*:

Actividades prácticas

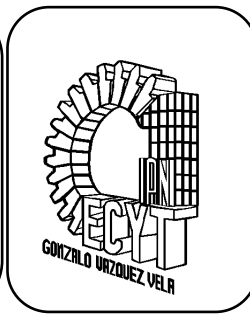
1. Realiza un programa que muestre los números pares del 1 al 999.
2. Realiza un programa con un con una sentencia *for* un contador de 0 100 de 5 en 5 y hágalo recurrente por medio de un sentencia *while* y *do while*.
3. Realice un programa que calcule la suma de “n” números ingresados por el usuario.
4. Haga recurrente un programa realizado en una práctica anterior con sentencia *do - while*.
5. Haga un programa que solicite una contraseña y de no ser ingresada correctamente deberá continuar solicitándolo.

Conclusiones:

1. Realiza el diagrama de flujo, codificación y prueba de escritorio de los programas realizados.



INSTITUTO POLITÉCNICO NACIONAL
 Centro de Estudios Científicos y Tecnológicos N° 1
 “Gonzalo Vázquez Vela”
 Academia de Sistemas Digitales T.V.
 Prácticas De Lenguaje de programación



NOMBRE DEL ALUMNO: _____
 _____ Apellido Paterno
 _____ Apellido Paterno Nombre
 N° DE BOLETA _____ GRUPO: _____

ASIGNATURA: *Lenguaje de Programación*

HOJA	DE	FECHA		EVALUACION
1	5			

Profesor Titular: _____
 Profesor Adjunto: _____

Práctica 10

Arreglos Unidimensionales y multidimensionales

Objetivo de la Unidad: UNIDAD III

Diseña programas empleando las estructuras de control condicionales y repetitivas basadas en el modelo de la programación estructurada.

Resultado de Aprendizaje Propuesto (RAP):

**Asociado con esta
Práctica: (3.1)**

Utiliza los arreglos como herramienta auxiliar en simplificación de variables del mismo tipo

Objetivos de la Práctica:

1. Definir los parámetros que requiere un arreglo en *lenguaje C*
2. Utilizar tipos de arreglos para el desarrollo de programas en *lenguaje C*

Equipo Necesario

1. PC
2. BorlandC
3. Cañón de proyección

Material

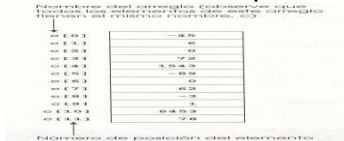
1. Apuntes
2. Información Impresa relacionada

Arreglos

Los arreglos son estructuras de datos que consisten de elementos de información del mismo tipo que están relacionados.

Definición.

Un arreglo es un grupo consecutivo de localidades de memoria que tienen el mismo nombre y el mismo tipo de datos. Para referirnos a una localidad o elemento particular de un arreglo, especificamos el nombre del arreglo y el número de posición del elemento particular dentro de este.



Sintaxis (Forma de declararlo).

Tipo_de_datos nombre_del_arreglo[numero_de_elementos];

Donde:

Tipo_de_datos : Se refiere al tipo de datos que vas a guardar en el arreglo, recuerda todos los datos tienen que ser del mismo tipo (int, float, char, double, etc.).

nombre_del_arreglo : Es el nombre con el que te vas a referir al arreglo durante todo el programa, elige un nombre no mayor a 8 caracteres y que sea relacionado con el uso que le vas a dar al arreglo.

numero_de_elementos : Es la cantidad de elementos que vas a guardar en el arreglo.

Ejemplos de declaración de arreglos:

```
int c[12];           /* Arreglo de 12 datos enteros */
char cadena[30];    /* Arreglo de 30 caracteres */
float numeros[100]; /* Arreglo de 100 numeros flotantes*/
```

El primer elemento de cualquier arreglo es el elemento cero. Por lo tanto, nos referimos al primer elemento del arreglo *nombre[0]*; Al número de posición entre los corchetes se le conoce como subíndice. Los subíndices deben ser enteros o expresiones enteras.

Ejemplo

El siguiente programa es un ejemplo de limpiar o poner a ceros todas las localidades de un arreglo, además de imprimirlo en la pantalla.

```
/* Programa que inicializa en cero e imprime un arreglo */
#include <stdio.h>
#include <conio.h>

void main(void)
{
    int i, n[10];

    clrscr();
    for(i = 0; i < 10; i++)           /* Inicializa el arreglo en ceros */
    {
        n[i] = 0;
    }
    printf("Elemento \t Valor \n");
    for(i = 0; i < 10; i++)          /* imprime el arreglo */
    {
        printf("\t %d \t\t %d \n",i,n[i]);
    }
    getch();
}
```


El siguiente programa es un ejemplo de cómo introducir valores al arreglo desde el inicio de la ejecución del programa.

```
/* Programa que inicializa con valores predeterminados e imprime un arreglo */
#include <stdio.h>
#include <conio.h>

void main(void)
{
    int i, a[6]={32,26,64,18,95,14}; /* Inicializacion de un arreglo */
                                   /* con valores predeterminados */
    clrscr();
    printf("Elemento \t valor \n");
    for(i = 0; i < 6; i++)          /* imprime el arreglo */
    {
        printf("\t %d \t\t %d \n",i,a[i]);
    }
    getch();
}
```

Los siguientes programas son aplicaciones del uso de los arreglos.

```
/* Programa que calcula la suma de los elementos de un arreglo */
#include <stdio.h>
#include <conio.h>

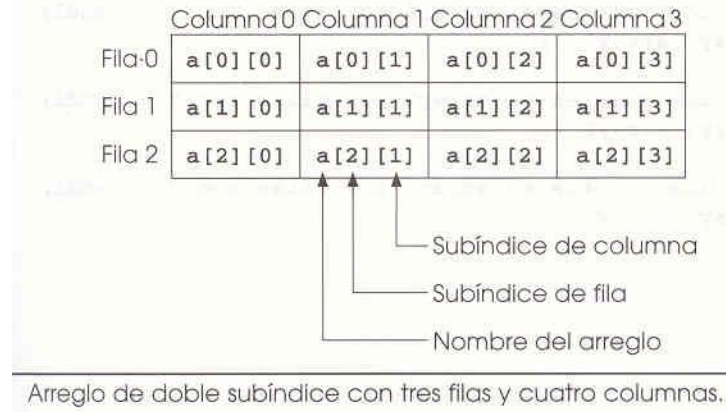
void main(void)
{
    const int datos = 6;
    int i, total = 0, n[datos]={1,6,4,8,5,4};/*Inicializacion del arreglo*/
                                           /* con valores predeterminados */
    clrscr();
    printf("Elemento \t valor \n");
    for(i = 0; i < datos; i++)          /* imprime el arreglo */
    {
        printf("\t %d \t\t %d \n",i,n[i]);
    }
    for(i = 0; i < datos; i++)          /* suma de los elementos */
    {
        total += n[i];
    }
    printf("La suma de todos los elementos del arreglo es %d \n", total);
    getch();
}
```

Arreglos de múltiples subíndices.

Un uso común de los arreglos de múltiples subíndices es la representación de tablas de valores que consisten en información ordenada en filas y columnas. Para identificar un elemento en particular de una tabla, debemos especificar dos subíndices: el primero identifica la fila del elemento y el segundo la columna.

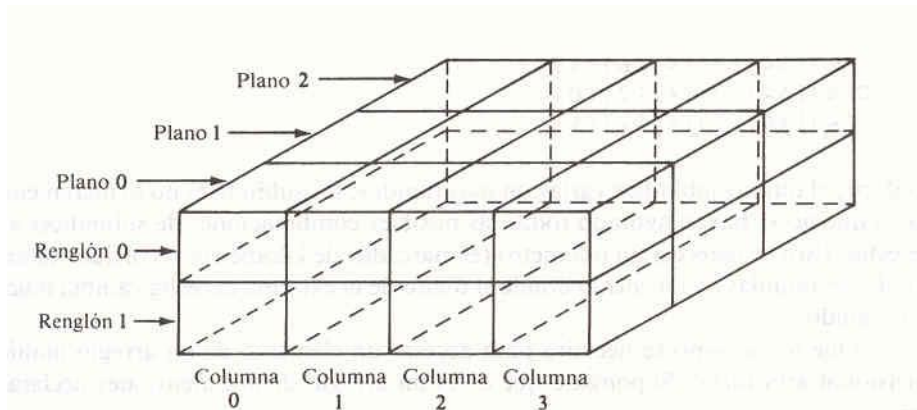
Ejemplo:

```
int a[3][4];      /* arreglo de tres filas por cuatro columnas */
```



Los arreglos de dos subíndices se conocen como arreglos bidimensionales. Los arreglos pueden tener más de dos subíndices, llamándose multidimensionales.

```
Ejemplo: int múltiple [2][4][3]; /* Arreglo tridimensional */
```



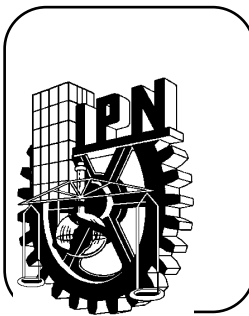
Questionario previo

1. Define un arreglo.
2. ¿Cuántos tipos de arreglos existen de acuerdo a su tamaño?
3. ¿Qué tipos de datos pueden guardar los arreglos?
4. ¿Existe una cantidad limitada de arreglos usados en *lenguaje C* dentro de un programa?
5. ¿Cuándo se considera conveniente usar arreglos en un programa en *lenguaje C*?

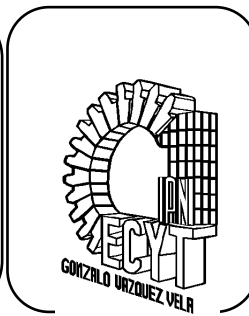
Actividades prácticas

1. Haga un programa que permita ingresar los datos a un arreglo unidimensional de 10 elementos.
2. Implementa un programa que lea del usuario los 9 elementos de un arreglo de 3x3 elementos, limpie la pantalla y los imprima en pantalla.
3. Realiza un programa que lea un arreglo de cinco elementos y guarde e imprima en otro arreglo el doble de los datos ingresados en el primer arreglo.

Conclusiones



INSTITUTO POLITÉCNICO NACIONAL
 Centro de Estudios Científicos y Tecnológicos N° 1
 “Gonzalo Vázquez Vela”
 Academia de Sistemas Digitales T.V.
 Prácticas De Lenguaje de programación



NOMBRE DEL ALUMNO: _____
 Apellido Paterno _____
 Apellido Paterno _____ Nombre _____
 N° DE BOLETA _____ GRUPO: _____

ASIGNATURA: *Lenguaje de Programación*

HOJA	DE	FECHA	EVALUACION
1	3		

Profesor Titular: _____
 Profesor Adjunto: _____

Práctica 11 Cadenas

Objetivo de la Unidad: UNIDAD III

Diseña programas empleando las estructuras de control condicionales y repetitivas basadas en el modelo de la programación estructurada.

Resultado de Aprendizaje Propuesto (RAP):

Asociado con esta Práctica: (3.2)	Emplea los arreglos como herramientas para implementar cadenas de caracteres
--	--

Objetivos de la Práctica:

1. El alumno desarrollara programas básicos donde manipule cadenas de caracteres.

Material 1. Computadora 2. BorlandC 3. Cañón Electrónico	Material 1. Apuntes 2. Información Impresa relacionada
---	--

Questionario Previo

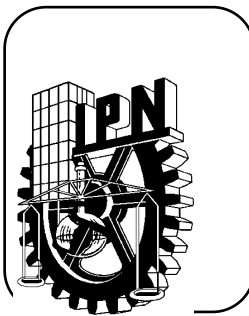
1. ¿Qué es una cadena de caracteres?
2. ¿Qué relación existe entre una cadena de caracteres y los arreglos?
3. ¿Qué aplicaciones tienen los arreglos de caracteres?

Actividades Prácticas

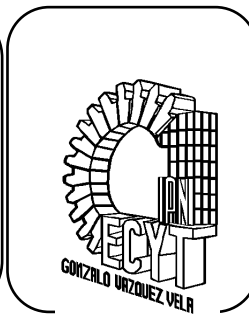
1. Realiza un programa que permita el ingreso de una cadena de caracteres de 10 elementos y de cómo salida la cantidad de letras A que contiene.
2. Realice un programa donde se ingrese una cadena de caracteres de al menos 6 caracteres y convierta mayúsculas en minúsculas y viceversa, el programa debe imprimir ambas cadenas en pantalla
3. Realice un programa donde se ingresen datos de 3 personas tales como *Nombre, Edad, Sexo, etc.*, y después los despliegue en pantalla

Conclusiones:

1. Realiza el diagrama de flujo, codificación y prueba de escritorio de los programas realizados.



INSTITUTO POLITÉCNICO NACIONAL
 Centro de Estudios Científicos y Tecnológicos N° 1
 “Gonzalo Vázquez Vela”
 Academia de Sistemas Digitales T.V.
 Prácticas De Lenguaje de programación



NOMBRE DEL ALUMNO: _____
 Apellido Paterno _____
 Apellido Paterno _____ Nombre _____
 N° DE BOLETA _____ GRUPO: _____

ASIGNATURA: *Lenguaje de Programación*

HOJA	DE	FECHA		EVALUACION
1	3			

Profesor Titular: _____
 Profesor Adjunto: _____

Práctica 12

Funciones Matemáticas y Gráficas

Objetivo de la Unidad: UNIDAD IV

Desarrolla programas utilizando funciones matemáticas, gráficas y de entrada – salida mediante el puerto, así como las diseñadas para uso específico de la problemática planteada.

Resultado de Aprendizaje Propuesto (RAP):

Asociado con esta Práctica: (4.1)	Utiliza las funciones predefinidas en el lenguaje de programación
--	---

Objetivos de la Práctica:

- El alumno desarrollara programas básicos que muestren texto y gráficos a color en pantalla.

Material 4. Computadora 5. BorlandC 6. Cañón Electrónico	Material 1. Apuntes 2. Información Impresa relacionada
---	--

Questionario previo

- Anote la librería que incluye los comandos de gráficos en *Lenguaje C*
- Mencione tres Modos de Video con su respectiva resolución
- Escriba la ruta para ingresar a la carpeta *BGI* donde se localizan los modos de video.
- Mencione los comandos para graficar una circunferencia y un rectángulo
- Diga qué función realiza el comando *DELAY*

Actividades prácticas

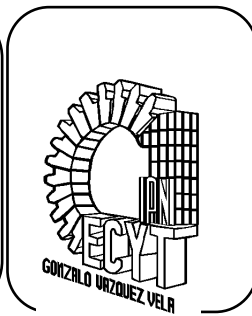
1. Diseña un programa que muestre en pantalla un texto en color.
2. Realiza un programa que despliegue un texto en pantalla parpadeando.
3. Desarrolle un programa que muestre el fondo de un texto en color.
4. Elabore un programa mediante el cual cambie de color la pantalla.
5. Elabore un programa que muestre en pantalla una caratula de datos del o los alumnos con efectos de color en ella.
6. Realice un programa mediante el cual cambie de color un texto.
7. Realice un programa mediante el cual cambie de color el fondo de un texto.
8. Elabore un programa que permita cambiar automáticamente el color de la pantalla.
9. Desarrolle un programa que dibuje una línea comenzando por la parte superior izquierda a cualquier punto de la pantalla.
10. Diseñe un programa mediante el cual aparezca en pantalla una línea recta con coordenadas aleatorias de inicio y fin.
11. Desarrollando un programa elabore una línea en pantalla con un color específico.
12. Realice un programa que muestre en pantalla un rectángulo en color.
13. Elabore un programa que muestre en pantalla una circunferencia en color.
14. Proponga un programa que rellene un rectángulo con un color específico en pantalla.
15. Proponga un programa que rellene una circunferencia con un color específico en pantalla.
16. Elabore un programa que haga se desplace un circulo de color moviéndose de izquierda a derecha.
17. Mediante un programa elabore un carrito que en pantalla avance de izquierda a derecha.
18. Desarrolle en pantalla un programa que dibuje un brazo robot de 4 articulaciones y un elemento terminal de pinza.

Conclusiones:

1. 1.- Mencione que finalidad tiene presentar textos y pantallas en color.
2. 2.- Diga que procedimientos y comandos utilizo para mezclar texto y gráficos.
3. 3.- Explique el procedimiento para generar animaciones o dibujos en movimiento basándose en gráficos.
4. 4.- Comente porque es importante la sentencia de control *For* en la implementación de gráficos en C.
5. 5.- Mencione la importancia de la graficación en la simulación de circuitos y proyectos electrónicos.



INSTITUTO POLITÉCNICO NACIONAL
 Centro de Estudios Científicos y Tecnológicos N° 1
 “Gonzalo Vázquez Vela”
 Academia de Sistemas Digitales T.V.
 Prácticas De Lenguaje de programación



NOMBRE DEL ALUMNO: _____
 Apellido Paterno _____
 Apellido Paterno _____
 Nombre _____
 N° DE BOLETA _____ GRUPO: _____

ASIGNATURA: *Lenguaje de Programación*

HOJA	DE	FECHA		EVALUACION
1	3			

Profesor Titular: _____
 Profesor Adjunto: _____

Práctica 13

Funciones de Manejo de Puertos

Objetivo de la Unidad: UNIDAD IV

Desarrolla programas utilizando funciones matemáticas, gráficas y de entrada – salida mediante el puerto, así como las diseñadas para uso específico de la problemática planteada.

Resultado de Aprendizaje Propuesto (RAP):

Asociado con esta Práctica: (4.1)	Utiliza las funciones predefinidas en el lenguaje de programación
--	---

Objetivos de la Práctica:

1. El alumno desarrollara programas básicos usando el puerto paralelo de la computadora en su opción de salida de datos.

Material 1. Computadora 2. BorlandC 3. Cañón Electrónico	Material 1. Apuntes 2. Información Impresa relacionada
--	---

Questionario previo

1. Diga qué función realiza un puerto en la computadora.
2. Mencione brevemente la diferencia entre la transmisión serie y la paralelo.
3. Dibuje los conectores DB-25 hembra y macho anotando los nombres respectivos para cada pin.
4. Anote que comando se utiliza para enviar datos de salida al exterior en *Lenguaje C* y que características presenta dicho comando.
5. Diga que precauciones deben tomarse para no dañar el puerto paralelo en la computadora.

Actividades prácticas

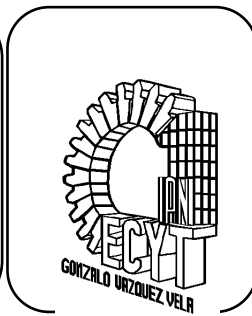
1. En base a lo desarrollado en clase elabore un cable con conector DB-25 teniendo mucho cuidado que las terminales en ambos extremos no hagan contacto ya que esto puede dañar el puerto en la computadora.
2. En un protoboard coloque para terminal de datos del puerto paralelo un LED y una resistencia en serie con valores específicos de corriente para tecnología TTL.
3. Una vez desarrollado lo anterior utilice compuertas lógicas entre el cable y los LED's que servirán de protección al puerto de la computadora.
4. Es importante verifique la alimentación a las compuertas lógicas y todo el alambrado en general del circuito electrónico ya que puede ocasionar un corto que dañe el puerto de la computadora.
5. Por el momento no conecte el DB-25 hasta haber compilado correctamente los siguientes programas.
6. Realice un programa que mande el Dato AA_H por un periodo de tiempo de 5 segundos y posteriormente se apaguen todos los LED's.
7. Desarrolle un programa que muestre en los 8 LED's un conteo en binario de 0 a 255.
8. Diseñe un programa que muestre en los 8 LED's un efecto de luces secuenciales de izquierda a derecha y viceversa.
9. Desarrolle un programa que muestre en los LED's el control de dos motores y dos pares de luces (un par blancas y un par rojas independientes). El primer motor moverá un carrito en sentido Adelante-Atrás. El segundo motor moverá el carrito en sentido Izquierda-Derecha. El par de luces blancas serán las luces delanteras del carrito y encenderán simultáneamente. El par de luces rojas encenderán de forma individual e indicaran cuando el carrito gire a la derecha o izquierda respectivamente. En base a lo anterior, proponga los movimientos necesarios de forma automática para que mediante un programa en C ingrese el carrito a un estacionamiento de noche y se estacione en un cajón del mismo.
10. Una vez elaborado todo lo anterior correctamente, proceda a conectar el DB-25 y observe si efectivamente se desarrollan en los LED'S lo que se propuso en los programas.
11. Finalmente añada al circuito electrónico anterior una interfase con dispositivos de potencia para conectar dos motores de C.D. que simulen el problema anterior del carrito.

Conclusiones:

1. Mencione qué importancia tiene el comando DELAY en el manejo de puertos.
2. Explique de forma breve el procedimiento para enviar un dato al puerto.
3. Mencione porque es importante colocar compuertas lógicas entre el cable del conector DB-25 y la aplicación.
4. Mencione algunas aplicaciones del uso de los puertos en la vida real.
5. Mencione que dificultades presento en el desarrollo de la presente práctica.



INSTITUTO POLITÉCNICO NACIONAL
 Centro de Estudios Científicos y Tecnológicos N° 1
 “Gonzalo Vázquez Vela”
 Academia de Sistemas Digitales T.V.
 Prácticas De Lenguaje de programación



NOMBRE DEL ALUMNO: _____
 Apellido Paterno _____
 Apellido Materno _____
 Nombre _____
 N° DE BOLETA _____ GRUPO: _____

ASIGNATURA: **Lenguaje de Programación**

HOJA	DE	FECHA		EVALUACION
1	2			

Profesor Titular: _____
 Profesor Adjunto: _____

Práctica 14

Funciones sin Parámetros

Objetivo de la Unidad: UNIDAD IV

Desarrolla programas utilizando funciones matemáticas, gráficas y de entrada – salida mediante el puerto, así como las diseñadas para uso específico de la problemática planteada.

Resultado de Aprendizaje Propuesto (RAP):

Asociado con esta Práctica: (4.2)	Diseñe funciones específicas para resolver las problemáticas planteadas.
--	--

Objetivos de la Práctica:

1. Definir los parámetros que requiere una función en *lenguaje C*
2. Utilizar funciones para el desarrollo de programas en *lenguaje C*

Material 1. Computadora 2. <i>BorlandC</i> 3. Cañón Electrónico	Material 1. Apuntes 2. Información Impresa relacionada
---	---

Cuestionario previo

1. ¿Cuál es el objetivo de programar mediante el uso de funciones?
2. ¿Cuántos tipos de funciones existen?
3. Menciona el nombre de la función o programa principal en *lenguaje C*
4. ¿Existe una cantidad limitada de funciones a usar en *lenguaje C*?
5. ¿Cuándo se considera conveniente usar funciones en un programa en *lenguaje C*?

Actividades prácticas

1. Escribir en el editor de C el siguiente código de programa:

```
#include <stdio.h>
#include <math.h>

void mediageo();

void main()
{
printf("\n Programa Principal que llama a una ");
printf("\n\n función de nombre mediageo");
mediageo();
printf("\n\n\n fin de programa principal");
}

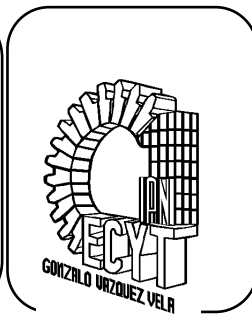
void mediageo()
{
float a,b;
printf("\n Introduce los dos números: \n");
scanf("%f %f",&a,&b);
printf("\n\n La Media Geométrica de %f y %f es %f",a,b,sqrt(a*b));
}
```

Al realizar la ejecución se deben observar los detalles siguientes:

- La sentencia *void mediageo();* indica al compilador, que el programa contiene una función definida por nosotros de nombre *mediageo* (llamado también subrutina o subprograma).
 - El programa principal (*voidmain()*), llama al subprograma en la línea: *mediageo();*
 - Observa de qué forma construimos el subprograma: *voidmediageo()* sin punto y coma, y a continuación, entre llaves, las líneas o sentencias del programa.
 - Observa también que cuando se acaba la ejecución del subprograma, el control pasa al programa principal, en la línea siguiente a la llamada del subprograma. En nuestro caso se ejecuta la línea: *printf("\n\n\n fin de programa principal");*
2. Utilizando funciones con la estructura del ejemplo anterior realizar un programa que permita calcular una suma, una raíz cuadrada, un logaritmo y una potencia (diseñar una función para cada caso y todas las funciones deben ser utilizadas en el mismo programa)



INSTITUTO POLITÉCNICO NACIONAL
 Centro de Estudios Científicos y Tecnológicos N° 1
 “Gonzalo Vázquez Vela”
 Academia de Sistemas Digitales T.V.
 Prácticas De Lenguaje de programación



NOMBRE DEL ALUMNO: _____
 Apellido Paterno _____
 Apellido Paterno _____ Nombre _____
 N° DE BOLETA _____ GRUPO: _____

ASIGNATURA: *Lenguaje de Programación*

HOJA	DE	FECHA		EVALUACION
1	2			

Profesor Titular: _____
 Profesor Adjunto: _____

Práctica 15 Funciones con Parámetros

Objetivo de la Unidad: UNIDAD IV

Desarrolla programas utilizando funciones matemáticas, gráficas y de entrada – salida mediante el puerto, así como las diseñadas para uso específico de la problemática planteada.

Resultado de Aprendizaje Propuesto (RAP):

Asociado con esta Práctica: (4.2)	Diseñe funciones específicas para resolver las problemáticas planteadas.
--	--

Objetivos de la Práctica:

- Definir los parámetros que requiere una función en *lenguaje C*
- Utilizar funciones para el desarrollo de programas en *lenguaje C*

Material 4. Computadora 5. BorlandC 6. Cañón Electrónico	Material 1. Apuntes 2. Información Impresa relacionada
---	--

Actividades prácticas

- Escribir en el editor de C el siguiente código de programa:

```
#include <stdio.h>

int cuenta(char nom[25]);

void main()
{
  int longit;
  char palabra[25];
  printf("\n Escribe una palabra de 25 caracteres como máximo: ");
  scanf("%s", palabra);
  longit=cuenta(palabra);
  printf("\n\n Longitud de %s = %d", palabra, longit);
}

int cuenta(char nom[25])
{
  int i;
  i=0;
```

```
while(nom[i] != '\0') i++;  
return i;  
}
```

Al realizar la ejecución se deben observar los detalles siguientes:

- La línea de programa `int cuenta(char nom[25]);` indica al compilador que además del programa principal (`void main`), hay una función de nombre `cuenta()`, que utilizará una variable tipo `string` con un máximo de 25 caracteres y además dicho subprograma retornará un valor tipo `int`, a diferencia de las funciones anteriores que no retornaban nada (`void`).
 - La línea de programa: `longit= cuenta(palabra);`, llama al subprograma `cuenta()` y en número entero que retorna, se asigna a la variable entera `longit`
 - Observa de qué forma contamos los caracteres ocupados por la variable `nom`: `nom[i] != '\0'`
 - Observa que la variable utilizada por la función `cuenta()`, no tiene porqué ser la misma: En el programa principal es `palabra` y en la función es `nom`. De todas formas es imprescindible que sea del mismo tipo: en los dos casos es `char` de 25 caracteres.
 - El valor retornado por la función, no tiene porqué corresponder a la misma variable: en el programa principal es `longit` y en la función es `i`. De todas formas es imprescindible que sean del mismo tipo: en los dos casos es `int`.
2. Diseñar un programa que por medio de una función que retorne parámetros, permita calcular las funciones seno y coseno de un argumento dado. (Los parámetros de la función deben ser: valor inicial, valor final y paso).
 3. Proponer un programa que mediante el uso de funciones muestre en número de días que tiene cada mes del año (por ejemplo: febrero tiene 28 días, agosto 31 días y septiembre 30).

ASCII

Hex	Símbolo	Hex	Símbolo	Hex	Símbolo	Hex	Símbolo	Hex	Símbolo	
0	0	NUL	16	10	DLE	32	20 (space)	48	30	0
1	1	SOH	17	11	DC1	33	21 !	49	31	1
2	2	STX	18	12	DC2	34	22 "	50	32	2
3	3	ETX	19	13	DC3	35	23 #	51	33	3
4	4	EOT	20	14	DC4	36	24 \$	52	34	4
5	5	ENQ	21	15	NAK	37	25 %	53	35	5
6	6	ACK	22	16	SYN	38	26 &	54	36	6
7	7	BEL	23	17	ETB	39	27 '	55	37	7
8	8	BS	24	18	CAN	40	28 (56	38	8
9	9	TAB	25	19	EM	41	29)	57	39	9
10	A	LF	26	1A	SUB	42	2A *	58	3A	:
11	B	VT	27	1B	ESC	43	2B +	59	3B	;
12	C	FF	28	1C	FS	44	2C ,	60	3C	<
13	D	CR	29	1D	GS	45	2D -	61	3D	=
14	E	SO	30	1E	RS	46	2E .	62	3E	>
15	F	SI	31	1F	US	47	2F /	63	3F	?

ASCII

Hex	Símbolo	Hex	Símbolo	Hex	Símbolo	Hex	Símbolo	Hex	Símbolo	
64	40	@	80	50	P	96	60 `	112	70	p
65	41	A	81	51	Q	97	61 a	113	71	q
66	42	B	82	52	R	98	62 b	114	72	r
67	43	C	83	53	S	99	63 c	115	73	s
68	44	D	84	54	T	100	64 d	116	74	t
69	45	E	85	55	U	101	65 e	117	75	u
70	46	F	86	56	V	102	66 f	118	76	v
71	47	G	87	57	W	103	67 g	119	77	w
72	48	H	88	58	X	104	68 h	120	78	x
73	49	I	89	59	Y	105	69 i	121	79	y
74	4A	J	90	5A	Z	106	6A j	122	7A	z
75	4B	K	91	5B	[107	6B k	123	7B	{
76	4C	L	92	5C	\	108	6C l	124	7C	
77	4D	M	93	5D]	109	6D m	125	7D	}
78	4E	N	94	5E	^	110	6E n	126	7E	~
79	4F	O	95	5F	_	111	6F o	127	7F	•

Código ASCII

Referencias Documentales

Fundamentos de Programación, Santos Manuel, Patiño Ismael, Carrasco Raúl, Editorial Alfa – Omega, RA Ma 2006

Enciclopedia, Lenguaje C++, 2ª Edición, Ceballos Sierra, Francisco Javier, Ra Ma Editorial, 2009

C/C++ Curso de Programación. 3ª Edición Ceballos Sierra, Francisco Javier, Editorial Ra Ma, 2007

Programación En C++: Algoritmos, Estructuras de Datos y Objetos, Joyanes Aguilar Luis, McGraw Hill, 2006

Prpgramación en C/C++, Alfonseca, Manuel y Sierra Urrecho, Alejandro, Anaya Multimedia, 2005

Fundamentos de Programación en Lenguaje C, Menchaca García, Felipe Rolando, Editorial IPN, 2010