

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

SECRETARÍA GENERAL

**DIRECCIÓN GENERAL DE INCORPORACIÓN Y REVALIDACIÓN DE
ESTUDIOS**

**Temario-Guía
de Cibernética y Computación II
(1604)**

Plan CCH - 1996

PRESENTACIÓN

La presente guía tiene como propósito orientarte en tu estudio para presentar con éxito el examen extraordinario de CIBERNÉTICA Y COMPUTACIÓN II.

En la guía encontrarás información acerca de los contenidos que, conforme al programa de estudio vigente, serán considerados en el examen extraordinario de la asignatura.

La guía contiene los siguientes apartados:

- 1. Temario de estudio.** Informa acerca de los contenidos que serán considerados en el examen extraordinario de la asignatura, conforme al programa de estudio vigente.
- 2. Actividades de aprendizaje.** Sugiere actividades que deberás realizar para obtener un mejor desempeño en tus aprendizajes. Te orienta acerca de la importancia relativa de cada tema en el conjunto de conocimientos del curso, así como del nivel de profundidad con que éstos serán evaluados en el examen. Asimismo, incluye ejemplos presentados en forma de reactivos con formatos similares a los que puedes encontrar en el examen y las respuestas correctas para que verifiques tus aciertos.
- 3. Bibliografía.** Proporciona información sobre los libros que puedes consultar para estudiar cada tema del programa de la asignatura.

ÍNDICE

1. Temario de estudio	6
2. Actividades de aprendizaje	9
• Respuestas selectas a actividades de aprendizaje	36
• Ejemplos de reactivos para el examen	46
• Respuesta a ejemplos de reactivos	52
3. Bibliografía	
Básica	53

INTRODUCCIÓN

Para que obtengas mejores resultados durante tu estudio es conveniente que utilices la guía de la siguiente manera:

TEMARIO DE ESTUDIO

En este listado identifica los temas que consideras que ya dominas, así como aquellos que desconoces o te resultan particularmente difíciles. Elabora un plan tentativo de trabajo, con días y cantidad de horas que dedicarás al estudio y repaso de los temas.

ACTIVIDADES DE APRENDIZAJE

Realiza cada una de las actividades sugeridas para el estudio de cada tema y elabora siempre un resumen o cuadro sinóptico destacando los aspectos más relevantes del tema. Esto te permitirá organizar tus conocimientos y ubicar los puntos principales de estudio y facilitará tus repasos. Ajusta tu plan inicial de trabajo de acuerdo a la importancia relativa de cada tema y a tus aptitudes o dificultades para su estudio.

De manera particular deberás prestar atención a lo siguiente:

Para el logro del aprendizaje de la programación con el lenguaje Turbo Pascal, es importante que contestes los cuestionarios de cada unidad, ya que la solución de ellos te proporcionarán los elementos básicos para el desarrollo de cada una de las unidades del programa. Asimismo, realiza la prueba de escritorio de los programas y fragmentos de programas para que comprendas cabalmente la estructura lógica y los elementos de programación contenidos en éstos (de ser posible realiza la codificación, edición, compilación y ejecución para que verifiques los resultados obtenidos en la prueba de escritorio, en la guía se anexa el compilador de Turbo Pascal, versión 6).

Es importante que constates tus aciertos con el listado de respuestas que se proporciona en la guía, para que detectes los aspectos que necesitas reforzar.

Recuerda que los reactivos de la guía son sólo indicativos del tipo de reactivos que puede contener el examen, y los contenidos explorados no excluyen otros (considerados en el programa de estudio vigente) no abordados en esta muestra. La información sobre la importancia relativa de cada tema en el programa de estudio, te permitirá tener una idea de la cantidad de reactivos que pudieran incluirse en el examen.

BIBLIOGRAFÍA

Consulta en cada tema los capítulos o fragmentos de los libros sugeridos en la guía. Puedes utilizar cualquier otro libro con el cual te sientas a gusto, ya que la bibliografía recomendada puede ser complementada y ampliada con libros que tú ya tengas o hayas utilizado anteriormente al estudiar estos temas.

1. TEMARIO DE ESTUDIO

El programa de la asignatura de Cibernética y Computación II, corresponde al Área de Matemáticas del Plan de Estudios vigente y contiene seis unidades temáticas, las cuales se especifican a continuación:

Primera Unidad: Sistema operativo.

1. Conceptos de sistema operativo MSDOS.
 - A) Características principales.
 - B) Componentes principales del sistema operativo.
 - a. Archivos.
 - b. Directorios y rutas.
 - c. Unidades de disco.
 - C) Iniciación y terminación de sesión.
 - a. Encendido y apagado del equipo.
 - b. Reinicialización (reset).
 - c. Proceso de carga del sistema.
 - D) Comunicación con el sistema.
 - a. Símbolo del sistema (prompt).
 - b. Cursor.
 - c. Teclado, generación de símbolos y conjunto ASCII.
 - d. Modo interactivo y de proceso en lotes (batch).
 - E) Comandos.
 - a. Identificación y ejecución.
 - b. Clasificación.
 - c. Manejo de archivos.
 - i. Copia.
 - ii. Borrado.
 - iii. Contenido.
 - iv. Búsqueda.
 - v. Cambio de nombre.
 - d. Manejo de directorios.
 - i. Creación.
 - ii. Eliminación.
 - iii. Localización.
 - e. Otros comandos.
 - i. Inicialización de discos.
 - ii. Copia de discos.

Cuarta Unidad: Procedimientos y funciones.

1. La modularidad como un herramienta para el manejo de programas extensos.
 - A. Concepto de modularidad.
 - B. Importancia de la modularidad.
2. Procedimientos y funciones en Pascal.
 - A. Concepto de procedimiento.
 - B. Concepto de función.
 - a. Valor de regreso y tipos permitidos.
 - C. Declaración y activación de procedimientos y funciones.
 - D. Parámetros.
 - a. Por valor.
 - b. Por referencia.
 - E. Alcance de identificadores.
 - a. Variables globales y locales.
 - b. Nombres de procedimientos y funciones
 - F. Recursividad.
 - a. Concepto matemático.
 - b. Implementación en el lenguaje.
 - c. Parte recursiva y parte terminal de un módulo recursivo.
 - d. Ventajas y desventajas del uso de rutinas recursivas.
 - e. Rutinas recursivas y sus equivalentes iterativas.

- iii. Definición de teclado.
- iv. Registro de fecha y hora.

Segunda Unidad: Lenguaje de programación Turbo Pascal.

1. Ambiente de programación.
 - A. Editor.
 - a. Comandos básicos.
 - B. Compilador.
 - C. Ejecución de programas y generación de códigos.
 - D. Manejo de archivos.
2. Elementos básicos del lenguaje.
 - A. Turbo Pascal como una implementación del lenguaje de programación Pascal.
 - B. Estructura de un programa en Pascal.
 - a. Encabezado.
 - b. Zona de declaraciones.
 - c. Zona de sentencias.
 - C. Identificadores, constantes, variables, palabras reservadas.
 - D. Tipos de datos primitivos.
 - a. Entero.
 - b. Caracter.
 - c. Lógico.
 - d. Real.
 - E. Sentencias.
 - a. Escritura.
 - b. Lectura.
 - c. Asignación.
 - F. Operadores, expresiones y orden de evaluación.
 - G. Funciones estándar.
 - H. Unidades en Turbo pascal.

Tercera Unidad: Estructura de control de secuencia.

1. Orden de ejecución de programa en Pascal.
2. Estructuras de control de secuencias.
 - A. Sentencias incondicionales.
 - a. Simples.
 - b. Compuestas.
 - c. La sentencia nula y el uso del

Quinta Unidad: Tipos de datos definidos por el usuario.

1. Tipos de datos estructurados.
 - A. Arreglos.
 - a. Elementos de un arreglo.
 - b. Índices y selección de elementos.
 - c. Almacenamiento de arreglos.
 - d. Dimensión de los arreglos.
 - B. Cadenas de caracteres.
 - a. Implementación en Turbo Pascal.
 - b. Funciones y procedimientos de manejo de cadenas.
 - c. Conversiones con otros tipos de datos.
 - C. Tipos enumerados y subrango.
 - D. Conjuntos.
 - E. Registros.
 - a. Elementos de un registro.
 - b. Campos y selección de elementos.
 - c. Almacenamiento de registros.
 - d. La sentencia With.
 - e. Registros variantes.
 - a. Archivos de acceso directo.
 - b. Funciones y procedimientos para el manejo de archivos de acceso directo.
2. Manejo de archivos.
 - A. Medios de almacenamiento.
 - B. Procesamiento secuencial.
 - a. Archivos de tipo texto.
 - b. Funciones y procedimientos para el manejo de archivos de tipo texto.
 - C. Procesamiento tipo Random
3. Estructuras dinámicas.
 - A. Concepto de stack y heap y el manejo de memoria de Turbo Pascal.
 - B. Apuntadores.
 - a. Localidad.
 - b. Contenido.
 - C. Listas ligadas.

terminador de sentencia “;”.

- d. Transferencia incondicional de control.
 - i. Sentencia GOTO.
- B. Sentencias condicionales.
 - a. Teorema de la programación estructurada.
 - b. Sentencias básica.
 - i. Sentencias IF THEN ELSE.
 - ii. Sentencia WHILE DO.
 - c. Otras sentencias condicionales.
 - i. Sentencia CASE.
 - ii. Sentencia FOR DO.
 - iii. Sentencia REPEAT UNTIL.

Sexta Unidad: Extensiones del lenguaje Pascal. Unidades en Turbo Pascal.

1. Fundamentos de la graficación por computadora.
 - A. Tarjetas de video.
 - B. Modos gráficos y resolución.
2. Rutinas de graficación en Turbo Pascal.
 - A. Inicialización.
 - B. Selección.
 - C. Trazo.
 - D. Inspección.
3. Rutinas de sonido.

2. ACTIVIDADES DE APRENDIZAJE

Primera Unidad: Sistema Operativo MS - DOS.

En esta unidad el estudiante reafirmará y profundizará sobre las características y componentes principales del sistema operativo, iniciación y terminación de sesión, la comunicación con la computadora y el manejo de los principales comandos internos y externos del Sistema Operativos MS – DOS.

Estrategia de aprendizaje.

- ✓ Investigación bibliográfica sobre las características y componentes principales del sistema operativo, iniciación y terminación de sesión, la comunicación con la computadora y el manejo de los principales comandos internos y externos del Sistema Operativos MS – DOS.
- ✓ Planteamiento de actividades académicas que permitan a los estudiantes la obtención de los conceptos básicos del Sistema Operativo MS – DOS y el desarrollo de habilidades sobre el manejo de comandos para un adecuado manejo del sistema de cómputo.

I. Conceptos del Sistema Operativo MS – DOS.

I.1 Describe al Sistema Operativo MS – DOS. _____

I.2 ¿Cuáles son los componentes principales del MS - DOS? _____

I.3 Describe al concepto de archivos. _____

I.4 Explica el concepto de directorio. _____

I.5 ¿Cómo se establece la comunicación con la computadora? _____

I.6 Explica que es un comando. _____

I.7 Los tipos de comandos del MS – DOS son: _____

I.8 Explica la diferencia entre comandos internos y externos. _____

I.9 Describe los comandos internos. _____

I.10 Explica los comandos externos. _____

I.11 Menciona y describe los comandos internos de la tabla 1.

Tabla 1: Comandos internos.

CLS	
COPY	
CD	
DEL	
DIR	
MD	
RD	
REN	

I.12 Menciona y describe los comandos externos de la tabla 2.

Tabla 2: Comandos externos.

AUTOEXEC	
CONFIG	
DISKCOPY	
EDIT	
FORMAT	
MEM	
TREE	
XCOPY	

I.13 Explica el proceso de carga del MS - DOS. _____

Segunda Unidad: Lenguaje de programación Turbo Pascal.

Esta unidad tiene como objetivo introducir a los estudiantes con el ambiente integrado de Turbo Pascal y con la programación mediante la metodología de resolución de problemas sencillos que involucran datos de tipo primitivos y las sentencias de lectura, escritura y asignación.

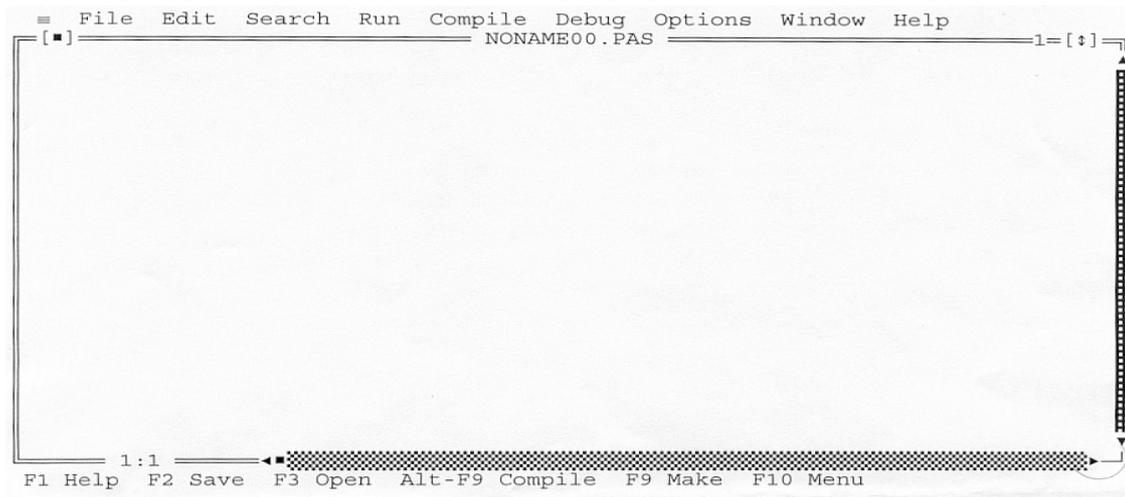
Estrategia de aprendizaje.

- ✓ Investigación bibliográfica sobre el ambiente integrado de Turbo Pascal, elementos básicos del lenguaje: estructura de programas, identificadores, constantes, variables, palabras reservadas, tipos de datos primitivos, sentencias, operadores, expresiones y orden de evaluación, funciones estándar y unidades en Turbo Pascal.
- ✓ Resolución de problemas con el apoyo de la computadora y el lenguaje de programación Turbo Pascal.
- ✓ Desarrollo de programas que involucran los tipos de datos primitivos y las sentencias de lectura, asignación y escritura.

II.1 Ambiente de programación.

II.1.1 Describe la pantalla de ambiente integrado de Turbo Pascal.

Pantalla. Ambiente de programación en Turbo Pascal.



II.2 Elementos básicos del lenguaje.

II.2.1 Describe a Turbo Pascal. _____

II.2.2 Menciona la estructura de programas en Turbo Pascal. _____

II.2.3 Explica el concepto de identificadores. _____

II.2.4. Describe el concepto de constantes. _____

II.2.5 Menciona el concepto de variable. _____

II.2.6 Explica el concepto de palabra reservada. _____

II.2.7 Enumera los tipos de datos primitivos. _____

II.2.8 Describe a los tipos de datos primitivos. _____

II.2.9 Explica el concepto se sentencia (o instrucción). _____

II.2.10 Describe la sentencia de lectura. _____

II.2.11 Explica la sentencia de asignación. _____

II.2.12 Describe la sentencia de escritura. _____

II.2.13 Menciona los operadores aritméticos. _____

II.2.14 Explica el concepto de expresión algebraica. _____

II.2.15 Menciona el orden de precedencia de los operadores aritméticos. _____

II.2.16 Describe algunas funciones estándar de Turbo Pascal. _____

II.2.17 Explica el concepto de unidades de Turbo Pascal. _____

II.3 Construcción de programas.

II.3.1 Desarrolla un programa que determine el salario bruto semanal de un empleado con base en las horas trabajadas y la tarifa por hora.

II.3.1.1 ¿Cuáles son los datos de entrada? _____

II.3.1.2 ¿Cuáles son los datos de salida? _____

II.3.1.3 ¿Cuáles son las relaciones entre los datos para determinar la salida? _____

II.3.1.4 Escribe las sentencias en el orden correcto para calcular el salario bruto de un empleado.

```
program salario_bruto;
uses crt;
var
  nombre: string[25];
  horastrab,tarifahora:integer;
  salbruto:real;
begin
  clrscr;
  writeln('INTRODUCE DATOS DEL EMPLEADO');
  write ('Digita nombre : ');
  readln(nombre);
  write ('Introduce horas trabajadas : ');
  readln (horastrab);
  write (_____);
  readln (_____);
  salbruto := _____;
  writeln('S A L A R I O   B R U T O   D E L   T R A B A J A D O R');
  writeln ('Empleado      Horas trabajadas Tarifa hora   Salbruto ');
  write(nombre, ' ');
  write(horastrab,' ');
  write(tarifahora,' ');
  write(salbruto:4:2,' ');
  readln;
end.
```

II.3.2 Escribe un programa en Turbo Pascal que calcule el promedio académico de un estudiante con base en la calificación tres exámenes y la calificación de dos tareas, asignándoles el 60 y 40%, respectivamente.

II.3.2.1 ¿Cuáles son los datos de entrada? _____

II.3.2.2 ¿Cuáles son los datos de salida? _____

II.3.2.3 ¿Cuáles son las relaciones entre los datos para determinar la salida? _____

II.3.2.4 Escribe las sentencias para calcular el promedio académico de un alumno, considerando el 60% para exámenes y el 40% para tareas, respectivamente.

```
program promedios;
uses crt;
var
  nombre:string[25];
  exa1,exa2,exa3,tarea1,tarea2:integer;
  promexa,promtar,promedio:real;

begin      { Inicia programa principal }
  clrscr;
  write('Digita el nombre: ');
  readln(nombre);
  write('Introduce calificación examen 1: ');
  readln(exa1);
  write('Introduce calificación examen 2: ');
  readln(exa2);
  write('Introduce calificación examen 3: ');
  readln(exa3);
  write('Digita tarea 1: ');
  readln(tarea1);
  write('Digita tarea 2: ');
  readln(tarea2);
  promexa:= _____;
  promtar:= _____;
  promedio:= _____;
  writeln('Nombre: ',nombre);
  writeln('Promedio = ',promedio:3:2);
  write('Oprime la tecla enter para continuar: ');
  readln;
end.
```

II.3.3 Encuentra la distancia y la pendiente entre los puntos $P_1(x_1, y_1)$ y $P_2(x_2, y_2)$.

II.3.3.1 ¿Cuáles son los datos de entrada? _____

II.3.3.2 ¿Cuáles son los datos de salida? _____

II.3.3.3 ¿Cuáles son las relaciones entre los datos para determinar la salida? _____

II.3.3.4 En el programa geometría, escribe las sentencias en forma adecuada para calcular la distancia y la pendiente que determina los puntos $P_1(x_1, y_1)$ y $P_2(x_2, y_2)$.

```
program geometria;
uses crt;
var
  x1,y1,x2,y2:integer;
  distancia, pendiente:real;
begin
  clrscr;
  write('DISTANCIA ENTRE DOS PUNTOS');
  write('Digita las coordenadas del 1er. Punto: ');
  readln(x1,y1);
  write('Digita las coordenadas del 2do. Punto: ');
  readln(x2,y2);
  distancia:= _____;
  pendiente:= _____;
  writeln('Distancia = ',distancia:4:2);

  _____
  writeln('Oprime la tecla "Enter" para continuar');
  readln;
end.
```

Tercera Unidad: Estructura de control de secuencia.

Esta unidad tiene como objetivo reforzar y profundizar los aprendizajes adquiridos por los estudiantes sobre el ambiente integrado de Turbo Pascal y en la programación mediante la metodología de resolución de problemas sencillos que involucran las sentencias de condición simple, compuesta y múltiple, if then , if then else y case, respectivamente, y las de ciclo while, repeat y for.

Estrategia de aprendizaje.

- ✓ Investigación bibliográfica sobre las sentencias de condición simple, doble y múltiple y las de ciclo while, repeat y for.
- ✓ Representación con diagrama de flujo y pseudocódigo de las sentencias de condición y de ciclo.
- ✓ Resolución de problemas con el apoyo de la computadora y el lenguaje de programación Turbo Pascal..
- ✓ Desarrollo de programas que involucran sentencias condicionales y de ciclo.

III.1 Conceptos generales de las sentencias de condición y ciclo.

III.1.1 Describe lo que es una variable lógica (booleana). _____

III.1.2 ¿Cuáles son los operadores lógicos básicos? _____

III.1.3 ¿Cuál es la representación de los operadores lógicos en Pascal? _____

III.1.4 Explica el concepto de expresión lógica (booleana). _____

III.1.5 ¿Cuál es la utilidad de las sentencias condicionales? _____

III.1.6 ¿Cuál es la utilidad de las sentencias de ciclo? _____

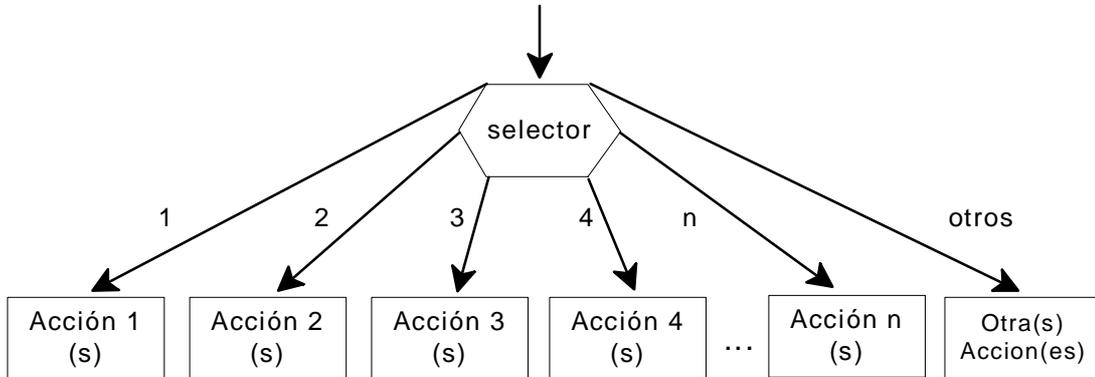
III.1.7 Describe el concepto de contador. _____

III.1.8 Describe el concepto de acumulador. _____

III.1.9 Explica el concepto de interruptor. _____

III.2.3 Con base en el diagrama de flujo de la sentencia de condición múltiple, da su descripción y codificación.

Diagrama de flujo de la sentencia de condición múltiple CASE.



Descripción	Codificación

III.2.4 Con base en el diagrama de flujo de la sentencia de ciclo while, da su descripción y codificación.

Diagrama de flujo de la sentencia while	Descripción
	<p data-bbox="1023 1675 1182 1709">Codificación</p>

III.3.1.4 El programa pago_netto calcula el salario neto de varios empleados con la sentencia de ciclo while, escribe correctamente las sentencias faltantes.

```
program pago_netto;
uses crt;
const
  imp_a = 0.05;
  imp_b = 0.10;
  imp_c = 0.15;
var
  nombre: string[25];
  horastrab,tarifahora:integer;
  salbruto, salneto, saltotal:real;
  categoria,sigue:char;
```

```

begin
    _____
while (sigue = 'S') or (sigue = 's') do
    begin
        clrscr;
        writeln('INTRODUCE DATOS DEL EMPLEADO');
        write ('Digita nombre : ');
        readln(nombre);
        write ('Introduce horas trabajadas : ');
        readln (horastrab);
        write ('Digita tarifa por hora : ');
        readln (tarifahora);
        write ('Digita categoría : ');
        readln (categoria);
        salbruto := horastrab * tarifahora;
        if (categoria = 'a') or (categoria = 'A') then
            salneto := _____
        else
            if (categoria = 'b') or (categoria = 'B') then
                salneto := _____
            else
                _____
                salneto := salbruto - salbruto * imp_c
            else
                writeln('Categoria fuera de rango, verifique');
        writeln;
        writeln('S A L A R I O   N E T O   D E L   E M P L E A D O');
        writeln ('Empleado Horas trabajadas Tarifa hora Categoría Salbruto Salneto ');
        write(nombre, ' ');
        write(horastrab, ' ');
        write(tarifahora, ' ');
        write(categoria, ' ');
        write(salbruto:4:2, ' ');
        writeln(salneto:4:2, ' ');
        saltotal := saltotal + salneto;
        write('Continuar proceso s/n : ');
        _____
    end;
    writeln('Salario total : ', saltotal:5:2);
end.

```

III.3.2 El programa promedios calcula el promedio académico de tres alumnos con base en el promedio de tres calificaciones de exámenes y con base en el promedio de dos tareas, considerando el 60% y 40 %, respectivamente. Cuando el promedio sea mayor o igual a seis, deberá aparecer la leyenda alumno “aprobado”, en caso contrario la leyenda de alumno “Reprobado”.

III.3.2.1 Completa correctamente las sentencias requeridas.

```
program promedios;
uses crt;
const
  lim = 3;
var
  nombre:string[25];
  contador, exa1,exa2,exa3,tarea1,tarea2:integer;

begin
  {Inicia programa principal}
  contador:=1;
  while contador <= lim do
    begin
      clrscr;
      write('Digita el nombre: ');
      readln(nombre);
      write('Introduce calificación examen 1: ');
      readln(exa1);
      write('Introduce calificación examen 2: ');
      readln(exa2);
      write('Introduce calificación examen 3: ');
      readln(exa3);
      write('Digita tarea 1: ');
      readln(tarea1);
      write('Digita tarea 2: ');
      readln(tarea2);

      _____
      _____
      promedio:=0.60*promexa+0.40*promtar;
      writeln('Nombre: ',nombre);
      writeln('Promedio = ',promedio:3:2);

      _____
      writeln(nombre,' Aprobado')

      _____
      writeln(nombre,' Reprobado');
      writeln('Oprime la tecla “enter” para continuar: ');

      _____
      readln;
    end;
  end.
end.
```

III.3.3 Desarrolla un programa que acepte de entrada un número entero que representa el día de la semana y dependiendo de éste deberá aparecer en pantalla el nombre del día (por ejemplo, si el día es 4, deberá aparecer “Miércoles”).

III.3.3.1 Completa las sentencias requeridas correctamente para que la ejecución sea para varias veces.

```
program día_semana;
uses crt;
var
  _____
  _____
begin
  bandera:='s';
  _____
  begin
    clrscr;
    write('Digita el día : ');
    readln(dia);
    case dia of
      1:writeln('1: Domingo');
      2:writeln('2: Lunes');
      _____
      4:writeln('4: Miércoles');
      5:writeln('5: Jueves');
      6:writeln('6: Viernes');
      _____
    else
      writeln(dia,' No es un día de la semana');
    end; {Fin del case}
    write('Mas días s/n: ');
    _____
  end;
end.
```

Cuarta Unidad: Funciones y procedimientos.

Esta unidad tiene como objetivo reforzar y profundizar los aprendizajes adquiridos por los estudiantes sobre la programación y abordar los correspondientes sobre la programación modular mediante la metodología de resolución de problemas sencillos que involucran funciones y procedimientos.

Estrategia de aprendizaje.

- ✓ Investigación bibliográfica sobre la programación modular.
- ✓ Resolución de problemas que involucran funciones y procedimientos.
- ✓ Construcción de programas que involucran la programación modular.

IV.1 Conceptos generales de la programación modular.

IV.1.1 Describe el concepto de la programación modular. _____

IV.1.2 ¿Cuáles son los elementos centrales de la programación modular? _____

IV.1.3 Explica el concepto de procedimiento. _____

IV.1.4 Menciona el concepto de función. _____

IV.1.5 Describe el concepto de parámetros. _____

IV.1.6 ¿Qué se entiende por parámetro valor? _____

IV.1.7 ¿Qué es un parámetro variable? _____

IV.1.8 Explica el concepto de variables globales. _____

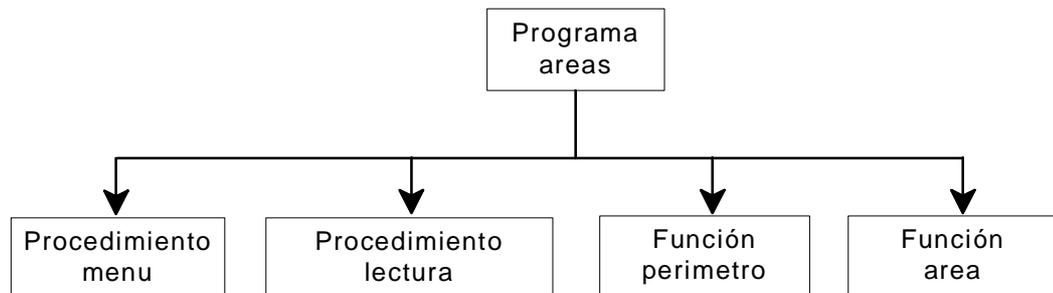
IV.1.9 Describe el concepto de variable local. _____

IV.1.10 Menciona la zona para la declaración de funciones y procedimientos. _____

IV.1.11 ¿Cómo se ejecutan las funciones y procedimientos? _____

IV.2 Desarrollo de programas que involucran la programación modular.

IV.2.1 Desarrolla un programa que calcule el perímetro y área de un rectángulo, teniendo como datos de entrada su base y altura de tipo entero. El programa debe contener los módulos que se indican en el siguiente diagrama.



IV.2.2 Contesta las preguntas siguientes:

IV.2.2.1 ¿Cuáles son los datos de entrada? _____

IV.2.2.2 ¿Cuáles son los datos de salida? _____

IV.2.2.3 ¿Cuáles son las relaciones entre los datos para determinar la salida? _____

IV.2.2.4 ¿Cuál es la palabra reservada para declarar a procedimientos? _____

IV.2.2.5 ¿Cuál es la palabra reservada para declarar a funciones? _____

IV.2.2.6 ¿Cuál es la zona para declarar a funciones y procedimientos? _____

IV.2.2.7 Describe la acción del procedimiento menu. _____

IV.2.2.8 Menciona lo que realiza el procedimiento lectura. _____

IV.2.2.9 Explica lo que realiza la función perimetro. _____

IV.2.2.10 Explica lo que realiza la función area. _____

IV.2.2.11 ¿Cuáles son las variables globales? _____

IV.2.2.12 ¿Cuáles son los parámetros? _____

IV.2.2.13 ¿Cuál es el tipo de los parámetros? _____

IV.2.2.14 Para declarar los parámetros como parámetros variable se requiere. _____

IV.2.2.15 Para la ejecución de procedimientos y funciones se requiere. _____

IV.2.2.16 Describe lo que realiza el programa principal. _____

IV.2.3 En la codificación del programa areas, escribe las sentencias faltantes en forma correcta.

```
program areas;
uses crt;
var
    _____
opcion:integer;

procedure menu;
begin
    clrscr;
    write('C A L C U L O   D E   A R E A S');
    write('1) Perímetro rectángulo');
    write('2) Area rectángulo');
    write('3) Salida');
    write('Elige opción: ');
    readln(opcion);
end;
```

```

_____ lectura;
begin
  clrscr;
  gotoxy(15,8);
  write('P R O C E D U R E  L E C T U R A');
  gotoxy(15,10);
  write('Digita base: ');
  readln(base);
  gotoxy(15,12);
  write('Digita altura: ');
  readln(altura);
end;

function _____(largo,ancho:integer):integer;
begin
  perimetro:=2*(largo+ancho);
end;

function area(largo,ancho:integer):integer;
begin
  area := _____
end;

begin          { Inicia programa principal }
  menu;        { Llamada al procedimiento menu }
  while (0 < opcion) and (opcion<3 ) do
  begin
    lectura;   { Llamada al procedimiento lectura }
    case opcion of
      1: begin
          gotoxy(15,14);          { Llamada a la función perimetro }
          write('Perímetro rectángulo = ', perimetro(base,altura));
        end;
      2: begin
          gotoxy(15,14);          { Llamada a la función area }
          write('Area del rectángulo= ',area(base,altura));
        end;
    end;
    readln;    { Pausa para presentar datos }
    menu;
  end;        { Fin del ciclo while }
end.

```

Quinta Unidad: Tipos de datos definidos por el usuario.

Esta unidad tiene como objetivo reforzar los aprendizajes adquiridos por los estudiantes sobre la programación y abordar la programación sobre tipos de datos definidos por el programador mediante la resolución de problemas sencillos que involucran a las estructuras de datos arreglos, cadenas, registros y archivos.

Estrategias de aprendizaje.

- ✓ Investigación bibliográfica sobre los tipos de datos arreglos, cadenas, registros y archivos.
- ✓ Resolución de problemas que involucran funciones y procedimientos.
- ✓ Construcción de programas que involucran la programación modular.

V.1 Conceptos generales de la estructura de tipo arreglo.

V.1.1 Describe las características de un arreglo. _____

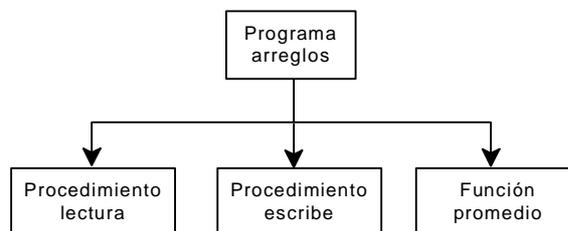
V.1.2 ¿Cuál es la dimensión de un arreglo? _____

V.1.3 ¿Cuál es la palabra reservada para declarar arreglos? _____

V.1.4 Explica cómo se accede a los elementos de un arreglo. _____

V.2 Construcción de programas con arreglos.

V.2.1 Desarrolla un programa que calcule el promedio de los diez elementos enteros contenidos en la estructura de arreglo numeros. El programa deberá considerar los módulos que se especifican en el diagrama siguiente.



V.3 Conceptos para la programación con la estructura de arreglos.

V.3.1 ¿Cuántos módulos tiene el programa arreglos? _____

V.3.2 Describe la acción del procedimiento lectura. _____

V.3.3 Describe la acción del procedimiento escribe. _____

V.3.4 Describe la acción de la función promedio. _____

V.3.5 ¿Cuál es el propósito para declarar un tipo de estructura arreglo? _____

V.3.6 Describe la acción del programa principal: _____

V.3.7 Escribe las sentencias en forma adecuada en el programa arreglos.

```
program arreglos;
uses crt;
const
    lim=5;
_____ arreglo=array[1..lim] of integer;
var
    _____: arreglo;
    i,j,acumula:integer;

procedure lectura(var numeros:arreglo);
begin
    for i:=1 to lim do
        begin
            write('Digita el elemento ',i,' ');
            readln(_____);
        end;
    end;

procedure escribe(numeros:arreglo);
begin
    write('[');
    for i:=1 to lim do
        write(_____);
    writeln(']');
end;

function promedio(numeros:arreglo):real;
begin
    for i:=1 to lim do
        acumula:= _____;
    _____ :=acumula/lim;
end;

begin
    clrscr;
    acumula:=0;
    writeln('Digita los elementos del arreglo ');
    lectura(_____);           { Llamada a lectura }
    writeln;
    writeln('Los elementos del arreglo son ');
    _____(numeros);         { llamada a escribe }
    writeln('Promedio = ',_____ (numeros):4:2); { llamada a promedio }
    readln;
end.
```

V.4 Conceptos generales de la estructura de tipo registro.

V.4.1 Describe las características de un registro. _____

V.4.2 ¿Cuál es la palabra reservada para declarar registros? _____

V.4.3 Explica cómo se accede a los campos de un registro. _____

V.4.4 ¿Cuál es la finalidad de declarar tipos de registros? _____

V.5 Conceptos generales de la estructura de tipo archivo.

V.5.1 Menciona los tipos de archivos que se manejan en Turbo Pascal. _____

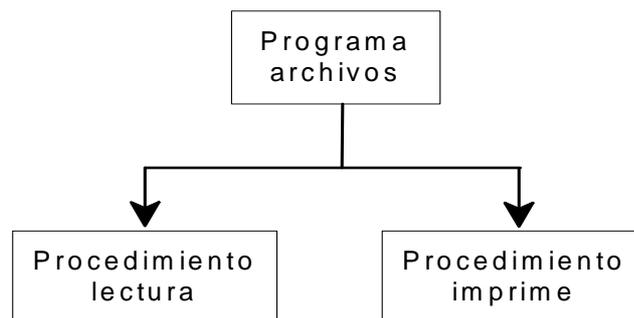
V.5.2 Describe las características de los archivos de tipo texto y tipificados. _____

V.5.3 ¿Cuál es la palabra reservada para declarar archivos? _____

V.5.4 Explica cómo se accede a los campos de un archivo. _____

V.6 Construcción de programas con registros y archivos.

V.6.1 Elabora un programa para crear un archivo con la estructura de un registro, cuyos campos sean nombre, dirección, estado_civil de tipo cadena y salario de tipo real. El programa deberá considerar los módulos que se presentan en seguida.



V.7 Conceptos para la programación con registros y archivos.

V.7.1 ¿Cuáles son los módulos que tiene el programa archivos? _____

V.7.2 Menciona la finalidad para declara tipos de registros. _____

V.7.3 Menciona la finalidad para declara tipos de archivos. _____

V.7.4 Explica el procedimiento assign. _____

V.7.5 Explica el procedimiento rewrite. _____

V.7.6 Explica el procedimiento close. _____

V.7.7 Explica el procedimiento reset. _____

V.7.8 Explica la función eof. _____

V.7.9 Explica la sentencia while not eof(nomina) do. _____

V.7.10 Describe la acción del procedimiento lectura. _____

V.7.11 Describe la acción del procedimiento imprime. _____

V.7.12 Describe la acción del programa principal. _____

V.7.13 Completa correctamente las sentencias del programa archivos _____

```
program manejo_archivos;
uses crt;
type registro = _____
                nombre:string;
                direccion: string;
                estado_civil: string;
                salario: real;
            end;
    archivo = _____
var
    sigue:char;
    nomina: _____
    empleado: _____
```

```
procedure captura(var nomina:archivo);
begin
    _____(nomina,'datos.dat');
    _____(nomina);
    sigue:='s';
    while (sigue = 's') or (sigue = 'S') do
        begin
            with _____ do
                begin
                    clrscr;
                    write('Digita nombre: ');
                    readln(nombre);
                    write('Digita dirección : ');
                    readln(direccion);
                    write('Digita Estado Civil : ');
                    readln(estado_civil);
                    write('Digita salario : ');
                    readln(salario);
                    _____(nomina,empleado);
                end;
            write('mas emplaeados s/n: ');
            readln(sigue);
        end;
    _____(nomina);
end;
```

```

procedure imprime(var nomina:archivo);
begin
    _____(nomina,'datos.dat');
    _____(nomina);
    while _____ do
        begin
            read(nomina, empleado);
            with _____ do
                begin
                    clrscr;
                    write('Nombre: ');
                    writeln(nombre);
                    write('Dirección: ');
                    writeln(direccion);
                    write('Estado Civil : ');
                    writeln(estado_civil);
                    write('salario: ');
                    writeln(salario);
                    write('Digita tecla enter para ontinuar');
                end;
            readln;
        end;
    _____(nomina);
end;

begin
    captura(nomina);
    imprime(nomina);
end.

```

Respuestas selectas a actividades de aprendizaje

Primera unidad.

- I.1 Sistema Operativo MS - DOS. Es un conjunto de programas que permiten la comunicación entre usuarios y la computadora, así como la administración y control de programas y dispositivos de la misma.
- I.2 Los componentes principales del Sistema Operativo son las archivos, directorios, las unidades de disco y las rutas para su acceso.

I.5 La comunicación entre la computadora y los usuarios se establece por medio de comandos.

I.7 Los tipos de comandos son internos y externos.

I.8 Los comandos internos se ejecutan directamente en la memoria principal, mientras que la ejecución de los comandos externos requieren la lectura desde el disco en que se encuentren.

I.9 Los comandos internos están agrupados en el archivo COMAND.COM, llamado el intérprete de comandos y son cargados a la memoria principal (RAM) al iniciar la sesión de trabajo.

I.10 Los comandos externos permanecen en el disco flexible o duro en el subdirectorio correspondiente y para ejecutarse necesitan ser leídos.

I.11 Descripción de los comandos internos de la tabla 1.

CLS	Comando para limpiar pantalla.
COPY	Comando para copiar archivos desde cualquier directorio u disco.
CD	Comando para cambiar de directorio.
DEL	Comando para borrar archivos.
DIR	Comando para desplegar la información en un directorio u disco.
MD	Comando para crear directorios.
RD	Comando para remover directorios.
REN	Comando para renombrar archivos.

I.12 Descripción de los comandos externos de la tabla 2.

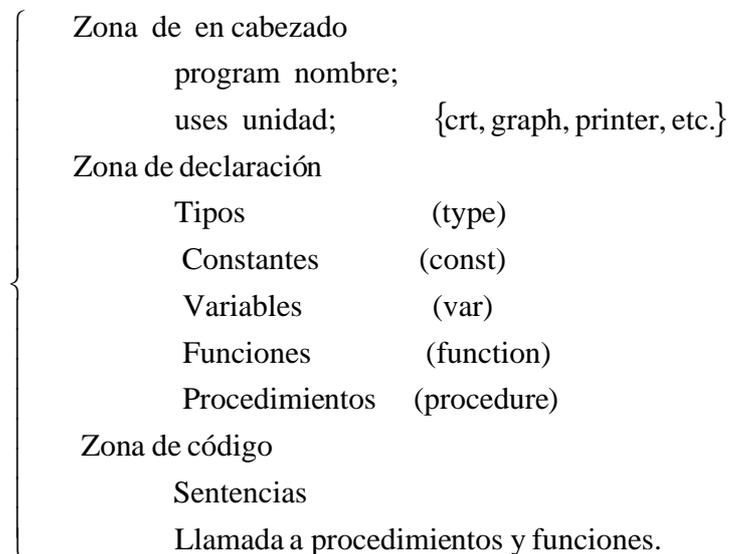
AUTOEXEC	Comando para la configuración de software y rutas de acceso.
CONFIG	Comando para la configuración de los dispositivos de hardware.
DISKCOPY	Comando para realizar copias de discos.
EDIT	Comando para edición de texto.
FORMAT	Comando para preparar discos.
MEM	Comando para verificar la memoria de la computadora.
TREE	Comando para desplegar la estructura del directorio u disco.
XCOPY	Comando para copiar subdirectorios y archivos contenidos en él.

I.13 El proceso de arranque de una computadora consiste de la ejecución de un programa de autoprueba contenida el ROM – BIOS, que checa los componentes de la computadora, para ello, busca los archivos ocultos IO.SYS y MSDOS-SYS en la unidad “A” y después en la unidad “C”, para preparar el equipo conectado para su uso, también se configuran todos los controladores de los periféricos, el procesador de comandos, los programas de aplicación y las rutas de acceso para su ejecución.

Segunda unidad.

II.1.1 La pantalla de Turbo Pascal está formada por la línea de menú que contiene las opciones File, Edit, Search, Run, Compile, Debugin, Options, Windows y Help; el área de trabajo y la línea de estado.

II.2.2 La estructura de programas en Turbo Pascal es la siguiente:



II.2.3 Identificadores son los nombre que utilizamos para nombrar a programas, procedimientos, funciones, arreglos, registros, archivos, variables, entre otros.

II.2.6 Palabras reservadas. Son aquellas que tienen un significado específico para el lenguaje, tal como se menciona en la siguiente tabla:

Palabra reservada	Significado
Begin	Inicio de un programa o subprograma.
Const	Se utiliza para declarar constantes.
Var	Se utiliza para declarar variables.
Record	Se utiliza para declarar registros.
File	Se utiliza para declarar archivos.
Type	Se utiliza para declarar tipos de datos.
Integer	Se utilizar para declara datos de tipo entero.

II.2.8 Los tipos de datos primitivos son: boolean, char, integer y real.

II.2.10 La sentencia de lectura se utiliza para introducir datos desde cualquier dispositivo de entrada, por ejemplo, READLN(dato) introduce desde el teclado el valor de dato.

II.2.11 La sentencia de asignación (:=), se utiliza para la asignación de expresiones a variables, por ejemplo, promedio := (cal1+ cal2 + cal3)/3.

II.2.12 La sentencia de escritura se utiliza para enviar datos hacia cualquier periférico de salida, por ejemplo, WRITELN(promedio), envía la salida de promedio a pantalla.

II.2.16 Algunas funciones incorporadas en Turbo Pascal se presentan en la tabla.

Función	Descripción
ABS (x)	Calcula el valor absoluto de argumento
a mod b	Calcula el resto entero de la división de dos números enteros.
a div b	Calcula el cociente entero de dos números enteros.
SQR(x)	Calcula el cuadrado del argumento.
SQRT(x)	Calcula la raíz cuadrada del argumento.
SIN(x)	Calcula la función seno del argumento.
EXP(x)	Calcula la exponencial e elevada al argumento.

II.2.17 Una unidad en Turbo Pascal es un conjunto de constantes, funciones y procedimientos. Ejemplos de unidades son CRT, PRINTER y GRAPH.

II.3 Construcción de programas.

II.3.1.4 Las sentencias en el orden correcto para calcular el salario bruto de un empleado son:

```
('Introduce tarifa por hora: ')  
tarifahora;  
horastrab * tarifahora
```

II.3.2.4 Las sentencias para calcular el promedio académico de un alumno, considerando el 60% para exámenes y el 40% para tareas son:

```
0.60*(exa1 + exa2 + exa3)/3  
0.40*(tarea1 + tarea2)/2  
Promexa + Promtar
```

II.3.3.4 Las sentencias en forma adecuada para calcular la distancia y la pendiente que determina los puntos $P_1(x_1, y_1)$ y $P_2(x_2, y_2)$ son:

```
sqrt(sqr (x2 - x1) + sqr(y2 - y1))  
(y2 - y1)/ (x2 - x1)  
writeln('Pendiente = ',pendiente:4:2);
```

III. Tercera Unidad.

III.1 Conceptos generales de las sentencias de condición y ciclo.

III.1.1 Una variable lógica (booleana) es aquella que puede tomar solamente un valor de dos posibles verdadero u falso.

III.1.2 Los operadores lógicos básicos son OR (suma lógica), AND (producto lógico) y NOT (negación).

III.1. 4 Una expresión lógica (booleana) es una combinación de operandos y operadores lógicos.

III.1.5 Las sentencias condicionales se utilizan para elegir alternativas en el proceso de ejecución de las sentencias.

III.1. 6 Las sentencias de ciclo se utilizan para la repetición de una sentencia o varias sentencias, dependiendo de la característica y condición de la sentencia de ciclo.

III.1.7 Un contador es una variable entera que incrementa su valor en forma constante.

III.1.8 Un acumulador es una variable que incrementa su valor en una cantidad variable.

III.1.9 Un interruptor es una variable que puede tomar sólo dos valores (verdadero u falso).

III.2 Representación de las sentencias condicionales y de ciclo mediante diagramas de flujo y codificación.

III.2.4 La sentencia while se utiliza para formar ciclos en el programa y de acuerdo a su diagrama de flujo tiene la característica de que primero verifica la condición de ciclo y luego ejecuta. El ciclo se repite mientras que la condición de ciclo sea verdadera, en caso contrario sale del ciclo.

Codificación	Observación
<pre>While exp_bool do bloque 1;</pre>	Cuando bloque 1, sea una sentencia compuesta, es decir, que contenga mas de una sentencia deberán delimitarse entre begin y end;

III.3 Desarrollo de programas.

III.3.1.4 Sentencias en la codificación del programa pago_netos.

```
Sigue:= 'S ';
Salneto := salbruto - salbruto * imp_a;
Salneto := salbruto - salbruto * imp_b;
if (categoria = 'c') or (categoria = 'C') then
  Readln(sigue);
```

III.3.2.1 Sentencias en la codificación del programa promedios.

```
Promexa, promtar: real;
Promexa := (exa1 + exa2 + exa3)/3;
Promtar := (tarea1 + tarea 2)/2;
If (promedio >= 6) then
Else
Contador := contador + 1;
```

III.3.3.1 Sentencias faltantes en la codificación del Programa dia_semana.

```
bandera:char;
dia:integer;
while (bandera = 'S') or (bandera = 's') do
3: writeln('3: Martes');
7: writeln('7: Sábado');
  readln(bandera);
```

IV. Cuarta Unidad.

IV.1 Conceptos generales de la programación modular.

IV.1.1 Programación modular es el proceso de dividir un programa en módulos y se le conoce como el diseño descendente.

IV.1.2 Los elementos centrales de la programación modular son las funciones y procedimientos, llamados subprogramas los cuales tienen las mismas reglas y características que los programas.

IV.1.3 Un procedimiento es un subprograma que realiza una tarea específica en el programa, se define en la zona para la declaración de variables con la palabra reservada *procedure* seguida de un nombre para identificarlo, así como la especificación de parámetros si es el caso. Para su ejecución es invocado desde el programa con el nombre asignado y con igualdad número de parámetros y tipo, si es que se definieron.

IV.1.4 Una función es un subprograma que realiza una tarea específica y regresa al llamarla un valor a diferencia del procedimiento que puede regresar cero o mas valores. Se define en la zona para la declaración de variables con la palabra reservada *function* seguida de un nombre para identificarla, así como la especificación de parámetros si es el caso y el tipo de dato que regresará. Para su ejecución es llamada desde el programa con el nombre asignado y con igual número de parámetros y tipo, si es que se definieron.

IV.1.5 Los parámetros definidos en procedimientos y funciones se utilizan para establecer comunicación entre éstos y el programa.

IV.1.6 Parámetro valor es aquel que proporciona una referencia del valor del parámetro a la función y/o procedimiento con el cual realizarán su tarea, el valor del parámetro no puede ser modificado.

IV.1.7 Parámetro variable es aquel que proporciona un valor del parámetro con el cual la función y/o procedimiento realizará su tarea, en su declaración va precedido por la palabra reservada *var*, para indicar que su valor puede ser modificado.

IV.2 Desarrollo de programas que involucra la programación modular.

IV.2.3 Codificación de las sentencias faltantes del programa areas.

```
base, altura: integer;  
procedure  
perimetro  
largo * ancho;
```

V. Quinta unidad.

V.1 Conceptos generales de la estructura de tipo arreglo.

V.1.1 Un arreglo es una estructura de datos en la que se almacena una colección de datos del mismo tipo. Por ejemplo, los identificadores enteros, nombres y letras, representan arreglos que contienen números enteros, nombre y letras, respectivamente.

enteros

67	13	9	3	17	37	43	100	89	25
1	2	3	4	5	6	7	8	9	10

nombres

María	Anabel	Juan	Marco	David
1	2	3	4	5

letras

1	e	b	p	c	a	g	m	i	d	h
2	k	o		f	r	n	s	q	l	j
	1	2	3	4	5	6	7	8	9	10

V.1.2 Se entiende por dimensión de un arreglo al producto de renglones por columnas en nuestros ejemplos las dimensiones son 1x10, 1x5 (unidimensionales) y 2x10 (bidimensional), respectivamente.

V.1.3 La palabra reservada para declarar arreglos es Array.

V.1.4 El acceso a los elementos de un arreglo se realiza con el identificador de arreglo y un índice entre corchetes, tal como se muestra en los ejemplos.

enteros [1], enteros [2], enteros [3], enteros [4], ,enteros [10]

nombres [1], nombres [2], nombres [3], nombres [4], ,nombres [10]

letras [1,1], letras [1,2], letras [1,3], letras [1,4], ,letras [1,10]

letras [2,1], letras [2,2], letras [2,3], letras [2,4], ,letras [2,10]

Acceso a los elementos de los arreglos enteros y letras con la sentencia for.

```
For i:= 1 to 10 do
  enteros [ i ];
```

```
For i:= 1 to 2 do
  For j:= 1 to 10 do
    letras [ i, j ];
```

V.3 Conceptos para la programación con la estructura de arreglos.

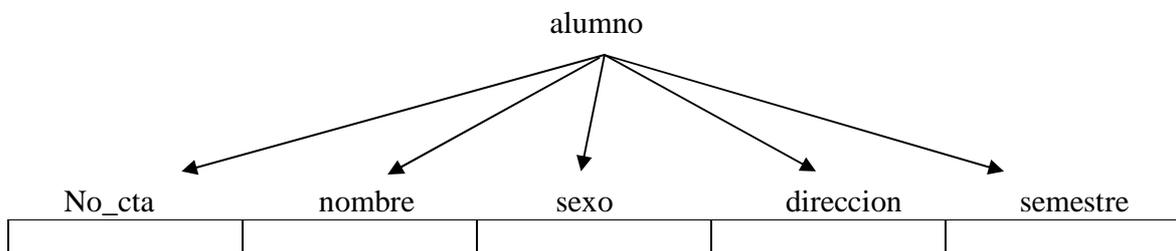
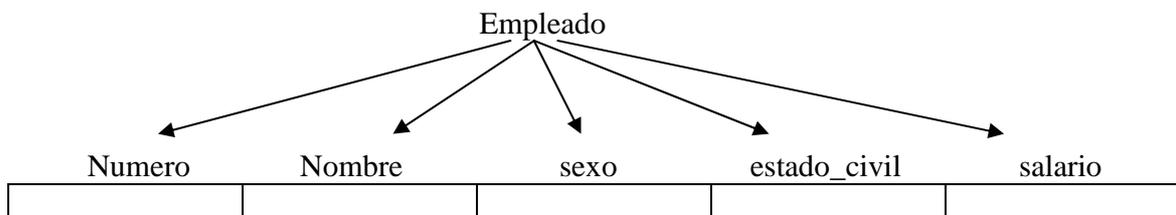
V.3.5 El propósito para crear un tipo de estructura de arreglo es para declarar variables con ese tipo de estructura que pueden ser utilizadas como parámetros en funciones y/o procedimientos.

V.3.7 Codificación del programa arreglos.

```
Type
numeros
numeros[i]
numeros[i]
acumula + numeros[i];
promedio
numeros
escribe
promedio
```

V.4 Conceptos generales de la estructura de tipo registro.

V.4.1 Un registro es un tipo de datos estructurado que consta de un conjunto de elementos que pueden ser del mismo tipo o tipo diferente. Los componentes de un registro se denominan campos, cada campo tiene un nombre de identificador de campo. Las estructuras empleado y alumno son registros.



V.4.2 La palabra reservada para declarar registros es Record.

V.4.3 El acceso a los campos de un registro se obtiene de dos formas, una de ellas consiste en separar con un punto el identificador de registro y el campo, la otra consiste en especificar la palabra reservada with seguida del identificador de registro, la palabra reservada do y la lista de campos del mismo.

Acceso a los elementos de los registros empleado y alumno.

empleado.numero, empleado.nombre, empleado.sexo, etc.

alumno.no_cta, alumno.nombre, alumno.sexo, etc.

Acceso a los elementos de los registros empleado y alumno con with.

With empleado do	with alumno do
Numero	no_cta
Nombre	nombre
Sexo	sexo
Estado_civil	direccion
salario	semestre

V.4.4 La finalidad de declarar tipos de registros es con la idea de declarar variables de ese tipo de registros para su manejo.

V.5 Conceptos generales de la estructura de tipo archivo.

V.5.1 Los tipos de archivos que se manejan en Turbo Pascal son de texto, tipificados y no tipificados.

V.5.2 Los archivos de tipo texto (text) son secuenciales, mientras que los segundos son aleatorios (file of), el acceso a los primeros es secuencial y el acceso a los segundos es aleatorio.

V.5.3 La palabra reservada para declarar archivos de tipo texto es file of text, mientras que para los archivos de tipo tipificado es file of <tipo>.

V.5.4 El acceso a los elementos de un archivo se logra con la sentencia de lectura o escritura del archivo y el registro separándolo con un punto del campo o bien con la palabra reservada with y el nombre del campo.

V.7 Conceptos para la programación con registros y archivos.

- V.7.3 La finalidad al declarar tipos de archivos es definir variables con ese tipo de archivos y facilitar su manejo.
- V.7.4 El procedimiento assign tiene la operación de asignar un archivo para establecer una correspondencia entre una variable de tipo archivo (interno) con un archivo externo situado en un disco.
- V.7.5 El procedimiento rewrite crea y abre un nuevo archivo. Si el archivo ya existe, rewrite borra su contenido, en caso contrario el archivo queda abierto para una operación de escritura.
- V.7.6 El procedimiento close tiene como función la operación de cerrar un archivo.
- V.7.7 El procedimiento reset es un procedimiento que tiene como función abrir un archivo existente para una operación de lectura.
- V.7.8 La función eof es una función lógica que devuelve el valor de verdadero cuando se ha alcanzado la marca de fin del archivo, en caso contrario devuelve falso.
- V.7.9 La sentencia while not eof(archivo) do, significa mientras no sea fin de archivo, hacer.

V.7.13 Completa correctamente las sentencias del programa archivos.

Record
File of registro;
archivo;
registro;
Assign
Rewrite
Empleado
Write
Close
Assign
Reset
Not eof (nomina)
Empleado
Close

Ejemplos de reactivos para el examen.

1. Conjunto de programas que permiten la comunicación con la computadora y la administración de sus recursos.

- A) Lenguaje de programación.
- B) Programa de aplicación.
- C) Sistema operativo.
- D) Procesamiento de datos.

2. Relaciona correctamente ambas columnas.

- | | |
|--------------------|---|
| a. Directorio | I Conjunto de información almacenada como una sola unidad. |
| b. Archivo | II Dispositivo que escribe datos en un disco magnético y lee datos del mismo. |
| c. Unidad de disco | III Parte de un disco que contiene un conjunto de archivos. |

- | | | | |
|----|----------|---------|---------|
| A) | a - I, | b - II, | c - III |
| B) | a - III, | b - I, | c - II |
| C) | a - II, | b - I, | c - III |
| D) | a - III, | b - II, | c - I |

3. Clasifica correctamente los comandos en internos y externos.

- | | | |
|--------------------|--------------------|---------|
| a. comando interno | b. comando externo | |
| I Dir | II Format | III Cls |
| IV Tree | V Copy | VI Edit |

- | | |
|---------------------|-----------------|
| A) a: I - II - III | b: IV - V - VI |
| B) a: I - III - V | b: II - IV - VI |
| C) a: II - III - IV | b: I - V - VI |
| D) a: II - IV - IV | b: I - III - V |

4. Software utilizado para la construcción de programas.

- A) Sistema operativo.
- B) Lenguaje de programación.
- C) Programa de aplicación.
- D) Lenguaje para la especificación de algoritmos.

5. Relaciona correctamente ambas columnas.

- | | |
|----------------------|--|
| a. Identificador | I Acción que debe realizar el programa. |
| b. Palabra reservada | II Es el nombre que se da a variables, constantes, programas, etc. |
| c. Sentencia | III Tiene un significado específico en el lenguaje. |

- | | | | |
|----|----------|----------|---------|
| A) | a - II, | b - III, | c - I |
| B) | a - III, | b - II, | c - I |
| C) | a - I, | b - II, | c - III |
| D) | a - II, | b - I, | c - III |

6. Proceso de traducir el programa fuente al programa objeto.

- A) Codificación
- B) Edición
- C) Compilación
- D) Ejecución

El programa viaje, calcula el costo (costo) de combustible que un automóvil consume(consumo_tot_gas) en el recorrido entre la ciudad de México y el puerto de Acapulco, con base en la distancia entre ambas ciudades que es de 400 km (dist), el rendimiento (rend) del auto (kilómetros recorridos por litro), y el costo por litro de gasolina (costo_litro_gas). Este programa te servirá para contestar los reactivos 7 y 8.

```
program viaje;
uses crt;
const
    dist = 400;
var
    rend, costo_litro_gas, costo, consumo_tot_gas:real;
begin
    clrscr;
    write('introduce el rendimiento: ');
    readln(rend);
    write('Introduce costo por litro de gasolina: ');
    readln(costo_litro_gas);
    consumo_tot_gas:= _____
    costo:= _____
    writeln('total gasolina: ',consumo_tot_gas:5:2);
    writeln('costo = ',costo:5:2);
    readln;
end.
```

7. ¿Cuáles son los datos de entrada?

- A) dist, vm
- B) rend, costo_litro_gas
- C) consumo_tot_gas, gasto
- D) rend, consumo_tot_gas

8. Elige las sentencias correctamente para calcular el consumo_tot_gas y su costo.

- A) dist*rend,
consumo_tot_gas*costo_litro_gas
- B) dist+rend,consumo_tot_gas*costo_litro_gas
- C) dist/rend, consumo_tot_gas*costo_litro_gas
- D) rend/dist, consumo_tot_gas*costo_litro_gas

9. ¿Cuál es la sentencia que se ejecuta cuando la condición de ciclo es verdadera?

- A) If then else
- B) Repeat until
- C) For do
- D) While do

El programa promedios clasifica el promedio de los estudiantes, cuando éste se encuentre en los rangos $6 \leq \text{prom} \leq 7.5$, $7.5 < \text{prom} \leq 9$, $9 < \text{prom} \leq 10$, deberá aparecer la leyenda 'alumno regular, destacado, sobresaliente o no se presentó al curso', respectivamente. Con este programa contesta los reactivos 10, 11, 12 y 13.

```
program promedios;
uses crt;
const
  alumnos = 5;
var
  nombre:string[30];
  indice,cal1,cal2,cal3:integer;
begin
  promgral:= 0;
  indice := 1;
  while (indice < alumnos) do
    begin
      clrscr;
      write('Digita el nombre: ');
      readln(nombre);
      write('digita calificación 1: ');
      readln(cal1);
      write('digita calificación 2: ');
      readln(cal2);
      write('digita calificación 3: ');
      readln(cal3);
      prom := (cal1 + cal2 + cal3)/3;
      if (prom >= 6) and (prom <=7.5) then
        writeln('Alumno Regular')
      else
        writeln('Alumno destacado')
      else
        if (prom > 9) and (prom <= 10) then
          writeln('Alumno sobresaliente')
        else
          writeln('Alumno no asistió al curso');
    end;
    readln;
    indice++;
  end;
  writeln('Promedio General = ',promgral/alumnos:3:2);
  write('Oprime la tecla enter para continuar: ');
  readln;
end.
```

10. ¿Cuántas veces se ejecuta el ciclo while?

- A) 1
- B) 4
- C) 5
- D) 6

11. ¿Qué variable representa a un acumulador?

- A) indice
- B) alumnos
- C) promgral
- D) prom

12. Elige las sentencias que completan la codificación del programa promedios en forma adecuada.

- A) prom,promgral:integer;
indice := indice +1;
promgral := promgral + prom;
- B) prom,promgral:real;
indice := indice +1;
prom := promgral + prom;
- C) prom,promgral:integer;
indice := indice +1;
prom := promgral + prom;
- D) prom,promgrl:real;
indice := indice +1;
promgral := promgral + prom;

13. ¿Cuál es la condición para clasificar al alumno como destacado?

- A) if (prom > 7.5) and (prom = 9) then
- B) if (prom > 7.5) and (prom < 9) then
- C) if (prom >= 7.5) and (prom <= 9) then
- D) if (prom > 7.5) and (prom <= 9) then

14. Relaciona ambas columnas en forma correcta.

- a. Programación modular I Llamada desde programa principal.
- b. Procedimiento II Proceso de diseño de descendente.
- c. Función III Un módulo que realiza una acción específica.
- d. Ejecución de módulos IV Un módulo que a su llamada regresa un valor.

- A) a - I, b - II, c - III, d - IV
- B) a - II, b - I, c - IV, d - III
- C) a - I, b - IV, c - III, d - II
- D) a - IV, b - III, c - II, d - I

15. Elige la opción que sea verdadera.

- A) Los arreglos son estructuras con datos dinámicas.
- B) Las cadenas son estructuras con datos dinámicas.
- C) Los registros son estructuras con datos dinámicos.
- D) Los archivos son estructuras con datos dinámicos.

16. La palabra reservada para declarar arreglos es.

- A) Array
- B) String
- C) Record
- D) File of

17. Relaciona ambas columnas de manera correcta.

- a. Assign I Procedimiento para preparar a un archivo Nuevo para escritura.
- b. Close II Procedimiento para preparar un archivo de lectura de.
- c. Reset III Procedimiento para establecer una relación entre el archivo interno y externo.
- d. Rewrite IV Procedimiento para cerrar un archivo.

- A) a - III, b - IV, c - II, d - I
- B) a - II, b - IV, c - I, d - III
- C) a - I, b - II, c - III, d - IV
- D) a - IV, b - III, c - I, d - II

El programa siguiente te servirá para contestar los reactivos 18, 19 y 20.

```
program conocidos;
uses crt;
type amigo = record
    nombre:string[25];
    direccion:string[40];
    edad: string[2];
    salario:real;
end;
amigos = file of amigo;
var
sigue:char;
persona : amigo;
archivo : amigos;
```

```

procedure graba (var archivo:amigos; persona:amigo);
begin
  assign (archivo,'amigo.dat');
  rewrite(archivo);
  sigue:='s';
  while (sigue = 's') or (sigue = 'S') do
    begin
      write('Digita nombre: ');
      readln(persona.nombre);
      write('Digita dirección : ');
      readln(persona.direccion);
      write('Digita edad : ');
      readln(persona.edad);
      write('Digita salario : ');
      readln(persona.salario);
      write(archivo,persona);
      Write('Mas amigos s/n: ');
      readln(sigue);
    end;
  close(archivo);
end; {fin procedimiento reporte}

procedure imprime (var archivo:amigos;persona:amigo);
begin
  clrscr;
  assign (archivo,'amigo.dat');
  reset(archivo);
  while not eof(archivo) do
    begin
      read(archivo,persona);
      writeln('Registros del archivo');
      writeln('Nombre = ',persona.nombre);
      writeln('Dirección = ',persona.direccion);
      writeln('Edad = ',persona.edad);
      writeln('Salario = ',persona.salario:8:2);
      delay(2000); { Retardo en milesegundos}
    end;
  close(archivo);
end; {fin procedimiento reporte}

begin
  graba(archivo,persona);
  imprime(archivo,persona);
end.

```

18. ¿Cuál es la variable del tipo de registro amigo?

- A) Amigo
- B) Persona
- C) Amigos
- D) Archivos

19. ¿Cuál es el archivo externo que genera el programa conocidos?

- A) Amigo.dat
- B) Persona.dat
- C) Archivo.dat
- D) Amigos.dat

20. ¿Cuál es la secuencia de comandos, para hacer la lectura de los registros del archivo.

- A) reset, assign, read
- B) read, rewrite, assign
- C) assign, rewrite, read
- D) assign, reset, read

Respuesta a ejemplos de reactivos.

- | |
|--------|
| 1 - C |
| 2 - B |
| 3 - B |
| 4 - B |
| 5 - A |
| 6 - C |
| 7 - B |
| 8 - C |
| 9 - D |
| 10 - B |
| 11 - C |
| 12 - D |
| 13 - D |
| 14 - A |
| 15 - D |
| 16 - A |
| 17 - A |
| 18 - B |
| 19 - A |
| 20 - D |

BÁSICA

Gonzalo Ferreira Cortés.
Informática para cuersos de Bachillerato.
Alfaomega.
México, 2000.

Contenido	Capítulo	Páginas	Observaciones
Primera Unidad	1	1 – 50	Contiene antecedentes de la Informática.
	2	65 – 91	Estructura física de la computadora.
	3	95 - 150	Estructura lógica de la computadora.

Gilberto Fuentes Romero.
Propuesta Educativa para la Materia de Cibernética y Computación II.
CCH SUR.
México, 2000.

Contenido	Capítulo	Páginas	Observaciones
Sistema Operativo	Primera Unidad	3 - 27	Contiene teoría, ejercicios y prácticas sobre MS – DOS.
Lenguaje de Programación Turbo Pascal.	Segunda Unidad	29 - 54	Contiene programas que involucran las sentencias de lectura, asignación y escritura.
Estructura de control de secuencia.	Tercera Unidad	55 - 82	Contiene programas que involucran las sentencias de condición y de ciclo.
Procedimientos y funciones.	Cuarta Unidad	83 – 107	Contiene programas que involucran la programación modular.
Tipos de datos definidos por el usuario	Quinta unidad	109 - 185	Contiene programas que involucran las estructuras de datos arreglos, registros y archivos.

Joyanes Aguilar, Luis.
Programación en Turbo Pascal, versiones 4.0,5.0, y 5.5.
McGraw - Hill.
México, 1992

Contenido	Capítulo	Páginas	Observaciones
Unidad 2	2 - 3	32 - 99	Contiene en entorno de programación en Turbo Pascal.
Unidad 3	4 - 5	108 - 174	Contiene sentencias condicionales y de ciclo.
Unidad 4	6	186 - 229	Contiene programación modular
Unida 5	11	365 – 399	Tratamiento de arreglos.
	13	441 – 482	Tratamiento de registros.
	14	490 - 542	Tratamiento de archivos.