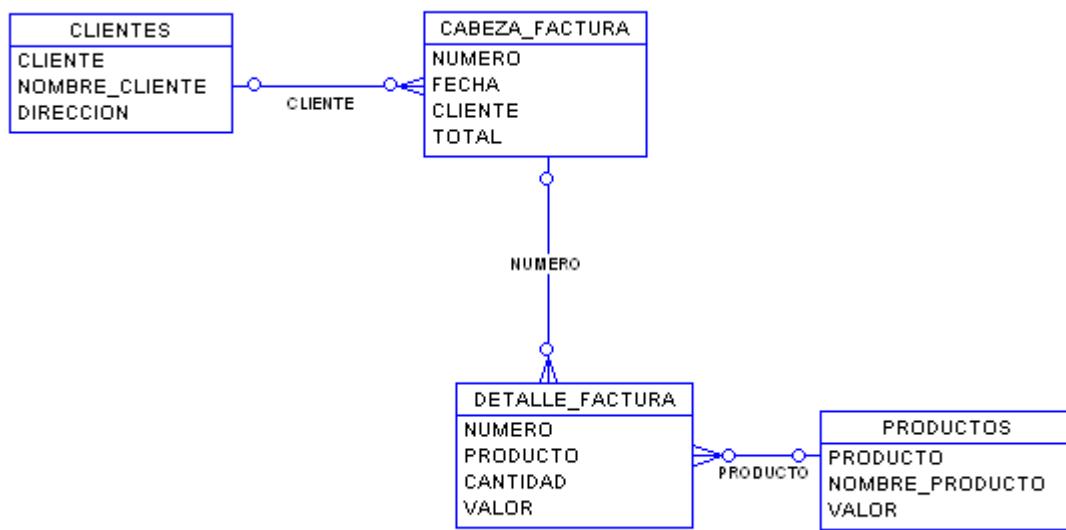


Nombre	TOMAS ALFONSO BALLESTEROS BURGOS
Fecha	11 DE OCTUBRE DE 2022
Edad	33 AÑOS

BASADO EN ESTE MODELO



Realice las siguientes consultas utilizando el SQL.

1. Seleccione Una Consulta De Los Productos Que Valen Más De \$1000. (PRODUCTO,NOMBRE_PRODUCTO, VALOR)

```

select PRODUCTO, NOMBRE_PRODUCTO, VALOR
from PRODUCTOS
where VALOR > 1000

```

2. Seleccione Una Consulta De Los Clientes Que Compraron Y Porque Valor Lo Hicieron. (CLIENTE, NOMBRE_CLIENTE, TOTAL)

```

select F.CLIENTE, C.NOMBRE_CLIENTE, F.TOTAL
from CABEZA_FACTURA F
inner join CLIENTE C on C.CLIENTE = F.CLIENTE

```

3. Seleccione Una Consulta De Las Compras Realizadas Entre Enero De 2005 Y Marzo De 2005. (NUMERO, NOMBRE_CLIENTE,FECHA,NOMBRE_PRODUCTO,CANTIDAD,VALOR)

```
select F.NUMERO, C.NOMBRE_CLIENTE, F.CLIENTE, F.FECHA, P.NOMBRE_PRODUCTO,  
D.CANTIDAD, D.VALOR  
from CABEZA_FACTURA F  
inner join CLIENTES C on C.CLIENTE = F.CLIENTE  
inner join DETALLE_FACTURA D on D.NUMERO = F.NUMERO  
inner join PRODUCTOS P on P.PRODUCTO = D.PRODUCTO  
where F.FECHA BETWEEN '2005-01-01' and '2005-03-31'
```

4. Que Clientes No Compraron Productos (CLIENTE, NOMBRE_CLIENTE)

```
select C.CLIENTE, C.NOMBRE_CLIENTE  
from CLIENTES C  
left join CABEZA_FACTURA F on F.CLIENTE = C.CLIENTE  
where F.CLIENTE is null
```

5. Realice Un Ranking De Los Productos Que Más Compraron De La Siguiente Manera:

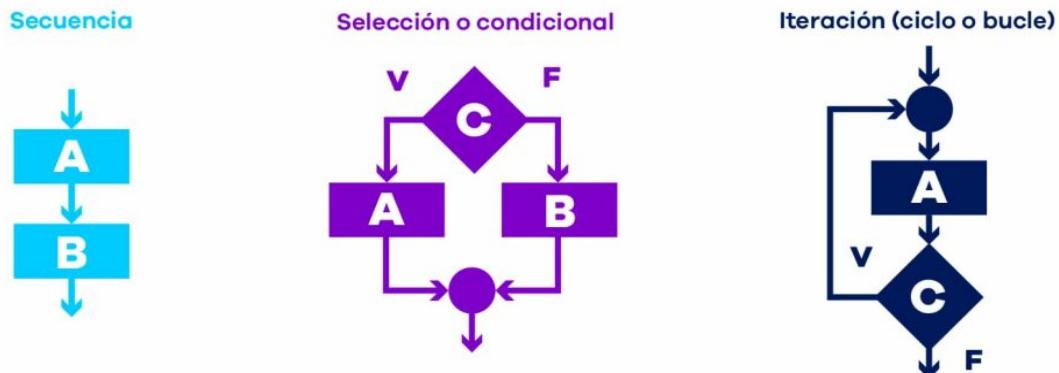
PRODUCTO	NOMBRE_PRODUCTO	TOTAL
1	PAPAYA	2
3	NARANJA	1
2	TOMATE	

```
select P.PRODUCTO, P.NOMBRE_PRODUCTO, count(P.PRODUCTO) as CONTEO,  
from CABEZA_FACTURA F  
inner join DETALLE_FACTURA D on D.NUMERO = F.NUMERO  
inner join PRODUCTOS P on P.PRODUCTO = D.PRODUCTO  
group by P.PRODUCTO  
order by CONTEO asc
```

- 1) Conceptos de lenguajes de desarrollo

- Definición de lenguajes de desarrollo estructurado
- Definición de lenguajes orientados a objetos
- Definición de clases, objetos, métodos, eventos (con ejemplos)
- Diferencia con ejemplos de lenguajes(c, java, pascal)
- Nombre algunos IDE que conozca.

- a. /El lenguaje de programación estructurado es una técnica utilizada al escribir código fuente de programación la cual sólo permite 3 sentencias o estructuras de control:



- b. /Los lenguajes orientados a objetos tratan de un paradigma de programación que trata de estructurar un programa en piezas simples y reutilizables de clases para crear instancias individuales de objetos. Es muy útil en proyectos extensos por la reutilización de código y se facilita la localización de errores.

- c. /Clases: Sirven para representar o crear un objeto, definen un conjunto de atributos y métodos para operar con dicho objeto.

- Ejemplo: Se crea una Clase llamada Moto con sus atributos Color, Modelo, Precio y sus métodos además de sus selectores y modificadores de encapsulado y constructores de clase pueden ser encender(), acelerar(), frenar(), apagar(), etc.

Objetos: Trata de un componente abstracto que permite separar los diferentes componentes de un programa, simplificando su elaboración, depuración y mejoras. Su creación se da al instanciar clases.

- Ejemplo: Creamos un objeto de la clase Moto para acceder a sus métodos y variables
`Moto moto = new Moto();`

Con el objeto moto podemos acceder a los métodos moto.encender(), moto.apagar(), etc.

Métodos: Los métodos son subrutinas que pertenecen a una clase y sirven para modificar los atributos y/o llevar a cabo operaciones variadas con los objetos.

- Ejemplo: encender() es un método de la clase Moto y se declara para hacer funcionar la Moto cuantas veces sea invocado.

Eventos: Es una señal que comunica a una aplicación que ha sucedido algo importante.

- Ejemplo: Cada que ejecutamos una acción con las entradas del equipo como un Clic en un elemento de la aplicación podemos generar un evento.

d. /Entre las principales: C es un lenguaje procedimental, no admite subprocessos, admite sugerencias, el colector de basura debe administrarse, admite llamada por valor y referencia, de nivel intermedio, no admite sobrecarga, no admite hilos.

Java es un lenguaje orientado a objetos, admite subprocessos, no admite sugerencias, el colector de basura es automático, admite llamada por valor, de alto nivel, admite sobrecarga de métodos, admite hilos.

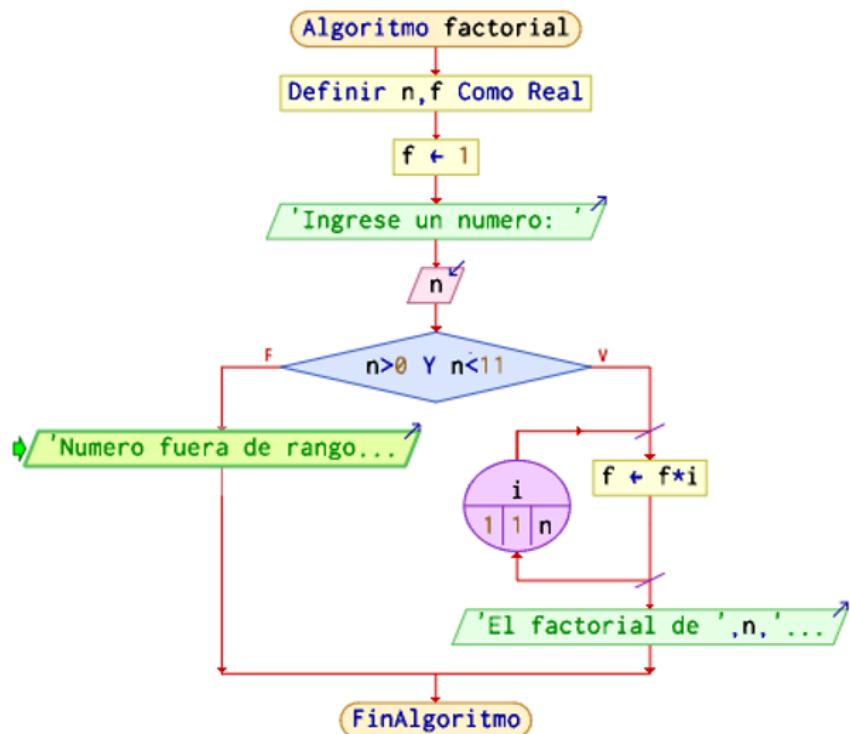
Pascal permite el paso de subprogramas como argumento de otros, el ámbito de variables y subprogramas es diferente, admite sólo tipos de datos simples, dispone de directivas de subprocessador, orientado a procedimientos, comprobación de tipos estática, lenguaje declarativo.

e. /IDE's conocidos:

- Visual Studio
- Visual Studio Code
- Eclipse
- Netbeans
- Code Blocks C++

2) Conceptos de algoritmos y lenguaje

a. Elabore un algoritmo de un factorial de 1 a 10 (diagrama de flujo y prueba de escritorio)

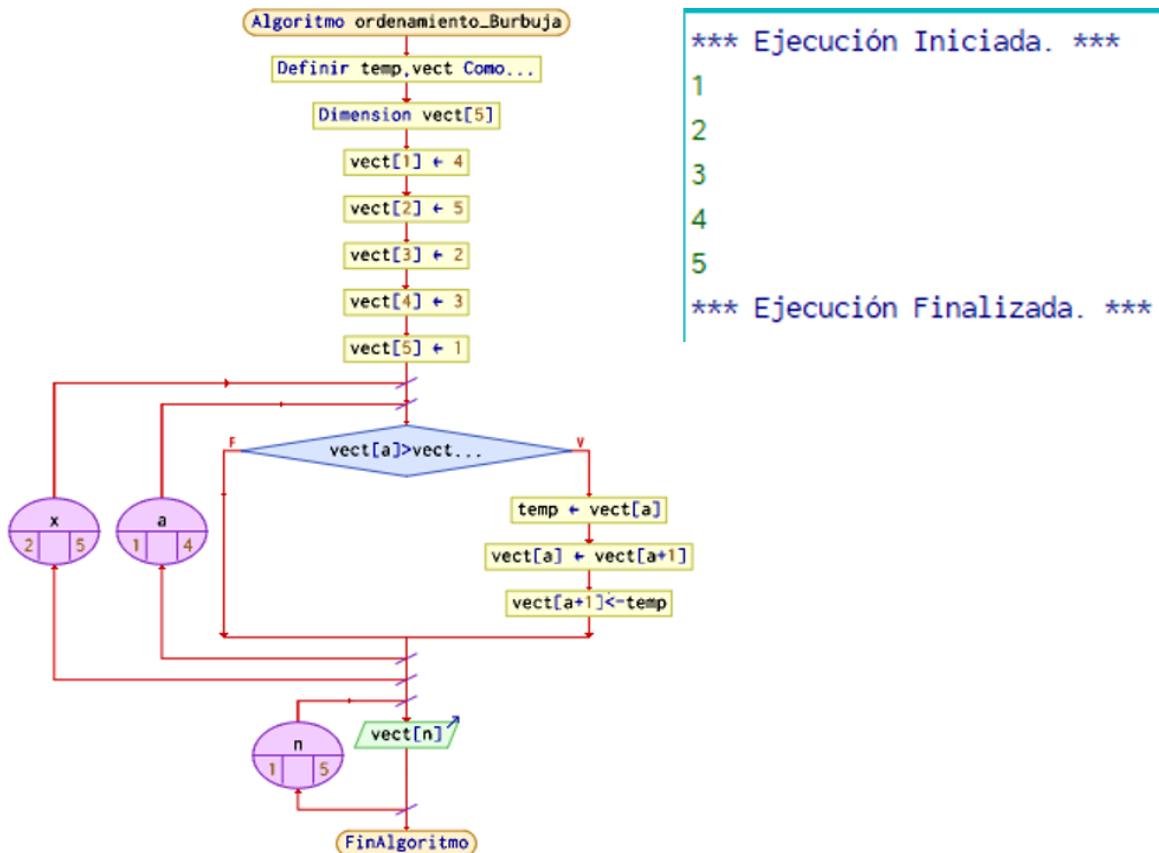


```

*** Ejecución Iniciada. ***
Ingrese un numero:
> 6
El factorial de 6 es: 720
*** Ejecución Finalizada. ***

```

- b. Elabore un algoritmo de orden burbuja (diagrama de flujo y prueba de escritorio)



- c. Elabore en pascal el orden de burbuja mediante recursividad.

```

procedure TForm2.Burbuja(vector: array of integer; cant: integer);
var
  auxi, i: integer;
begin
  if cant ≥ 0 then
  begin
    for i := 0 to cant - 1 do
    begin
      if vector[i] > vector[i + 1] then
      begin
        auxi := vector[i];
        vector[i] := vector[i + 1];
        vector[i + 1] := auxi;
      end;
    end;
    Burbuja(vector, cant - 1);
  end
  else
  begin
    for i := 0 to length(vector) - 1 do
      Memo1.Lines.Add(intToStr(vector[i]))
  end;
end;

```