



**UNIVERSIDAD AUTONOMA DE NUEVO LEON**  
**FACULTAD DE CIENCIAS FISICO MATEMATICAS**

<b>NOMBRE</b>	Marcelo Rios Quiroga
<b>MATRICULA</b>	2058125
<b>GRUPO</b>	051
<b>UNIDAD DE APRENDIZAJE</b>	Programacion basica
<b>MAESTRA</b>	LUISA ALEJANDRA MONREAL MENDOZA

08/12/2023

## ¿Que son los puntos de venta?

Los "puntos de venta" generalmente se refieren a los lugares físicos o virtuales donde se realizan transacciones comerciales, como la venta de productos o servicios. En el contexto de programación

Crear un sistema de punto de venta en implica diseñar y programar un programa que gestione la venta de productos, realice transacciones, lleve un inventario, y otras funciones relacionadas con el comercio. Aquí hay un ejemplo muy básico de cómo podrías empezar a estructurar un programa de punto de venta en C++

```
22 do {  
23     system("cls");  
24     cout<<"\n";  
25     cout<<"\n    Menu de Punto de Venta";  
26     cout<<"\n";  
27     cout<<"\n        1. Alta de Ordenes";  
28     cout<<"\n        2. Modificar Orden";  
29     cout<<"\n        3. Eliminar o Cancelar Orden";  
30     cout<<"\n        4. Lista de Ordenes";  
31     cout<<"\n        5. Limpiar Pantalla";  
32     cout<<"\n        6. Salir";  
33     cout<<"\n";  
34     cout<<"\n    Opcion: ";  
35     cin>>opcion;  
36  
37 }
```

## ¿que es la switch opcion?

la instrucción switch es una estructura de control que permite seleccionar entre varias opciones o casos basándose en el valor de una expresión. Es una alternativa a la instrucción if-else cuando se necesitan múltiples comparaciones sobre el mismo valor.

Es importante notar que después de cada case, se debe usar la palabra clave break para salir del bloque switch. Si se omite el break, la ejecución continuará hacia abajo en los casos siguientes, incluso si no coinciden con la expresión.

La etiqueta default es opcional y se utiliza para especificar un bloque de código que se ejecutará si la expresión no coincide con ninguno de los casos anteriores.

```

37
38
39 switch(opcion){
40 case 1:
41     system ("cls") ;
42     do
43     {
44         cout<<"\n Numero de Ticket:";
45         cin>>matriz[i][1];
46
47         cout<<"\n Nombre del vendedor:";
48         cin>>nomven[i];
49         cout<<"\n Fecha:";
50         cin>>matriz[i][2];
51         cout<<"\n Cuantas Pizzas Monkey:";
52         cin>>matriz[i][3];
53         cout<<"\n Cuantas Pizzas Montivirdi:";
54         cin>>matriz[i][4];
55         cout<<"\n Propina (10%, 15%, 20%):";
56         cin>>propina;
57         venta = (matriz[i][3]*99)+(matriz[i][4]*140);
58         matriz[i][5]=(venta*propina)/100;
59         cout<<"\n Descuento o Promocion:";
60         cin>>desc;
61         matriz[i][6]= (venta * desc)/100;
62         matriz[i][7]= venta*0.16;
63         matriz[i][8]= venta + matriz[i][5]- matriz[i][6]+ matriz[i][7];
64         cout<<"\n total: ";
65         cout<<matriz[i][8];
66         i=i+1;
67         cout<<"\n Desea Salir si[1] no[2]";
68         cin>>res;
69     }while(res>>1);
70
71     break;

```

### ¿Que es el System ("cls")?

system("cls") es una instrucción en C++ que se utiliza para limpiar la pantalla de la consola en sistemas basados en Windows. La función system permite ejecutar comandos del sistema operativo desde un programa C++.

En este caso, system("cls") ejecuta el comando "cls" en la consola de Windows, que significa "clear screen" (limpiar pantalla). Cuando esta instrucción se ejecuta, borra todo el contenido de la ventana de la consola, proporcionando una apariencia de pantalla limpia.

Es importante mencionar que system("cls") es específico de sistemas Windows. Si estás desarrollando un programa que se ejecutará en sistemas operativos diferentes, como Linux, puedes utilizar system("clear") en lugar de system("cls") para lograr un efecto similar.

```

132     }
133     break;
134
135     case 5:
136         cout<<"\n Limpiar pantalla";
137
138
139         system ("cls");
140
141         break;
142
143
144     case 6:
145         exit(1);
146
147         break;
148     }
149 }
150 while (res=6);
151 return 0;
152 }
153

```

### ¿Que es el Do, While?

El bucle do-while es una estructura de control que permite repetir un bloque de código mientras una condición sea verdadera. A diferencia del bucle while, el bucle do-while garantiza que el bloque de código se ejecutará al menos una vez, ya que la evaluación de la condición ocurre después de la ejecución del bloque de código.

En este caso, el bloque de código se ejecutará una vez sin importar la condición. Después de la ejecución del bloque, la condición se evalúa. Si la condición es verdadera, el bucle se repetirá; de lo contrario, el programa continuará con la siguiente instrucción después del bucle.

Es importante destacar que debido a la naturaleza del bucle do-while, el bloque de código siempre se ejecutará al menos una vez, independientemente de la condición. Esto lo hace útil en situaciones en las que deseas garantizar que un bloque de código se ejecute al menos una vez antes de verificar la condición.