

# **Universidad Antonio Nariño**



## **Asignatura: Lógica computacional**

### **Proyecto de curso**

**Docente: Juan Martínez.**

### **Grupo 1**

#### **Integrantes:**

- **Jennifer Rivera.**
- **Shirley Mora.**
- **Juan José Martínez**

**Programa: Ingeniería Biomédica – Primer semestre.**

**2025**

### **3. WORKFLOW DE IMPLEMENTACIÓN:**

3. Por cada requerimiento funcional desarrollado en el numeral 1.2, programa en un lenguaje de programación (consulta en referencia las pág. 34-39)

## Requerimiento funcional 1

Nombre: visualizar Productos	<pre>{#include &lt;iostream&gt;  using namespace std;  struct Producto {      string nombre;      string tipo;      int cantidad;      int minimo;      float precioBase;      float precioFinal;      void asignarDatosConIVA();      void mostrar();  };  void Producto::asignarDatosConIVA() {      cout &lt;&lt; "Ingresa la cantidad disponible: ";      cin &gt;&gt; cantidad;      cout &lt;&lt; "Ingresa la cantidad minima para abastecer: ";      cin &gt;&gt; minimo;      cout &lt;&lt; "Ingresa el precio base del producto: ";</pre>
Parámetros:	
Ninguno (la función utiliza el array productos definido en main)	
Retorno:	
void (la función no devuelve ningún valor, sino que imprime la información en la consola)	
Descripción:	
Esta función muestra en la consola la información detallada de todos los productos disponibles en el inventario (nombre, tipo, cantidad actual, cantidad mínima para abastecimiento, precio base y precio final).	

```
cin >> precioBase;
```

```
if (tipo == "papeleria") {
```

```
    precioFinal = precioBase + (precioBase * 0.16);
```

```
} else if (tipo == "drogueria") {
```

```
    precioFinal = precioBase + (precioBase * 0.12);
```

```
} else if (tipo == "supermercado") {
```

```
    precioFinal = precioBase + (precioBase * 0.04);
```

```
}
```

```
}
```

```
void Producto::mostrar() {
```

```
    cout << "\n--- DATOS DEL PRODUCTO ---\n";
```

```
    cout << "Nombre: " << nombre << endl;
```

```
    cout << "Tipo: " << tipo << endl;
```

```
    cout << "Cantidad en bodega: " << cantidad << endl;
```

```
    cout << "Cantidad minima: " << minimo << endl;
```

```
    cout << "Precio base: $" << precioBase << endl;
```

```
    cout << "Precio final con IVA: $" << precioFinal << endl;
```

```
    cout << "\n";
```

```
}
```

```
int main() {
```

```
Producto productos[9];
```

```
string categoria;
```

```
// Productos de Drogueria
```

```
productos[0].tipo = "drogueria";
```

```
productos[0].nombre = "Acetaminofen";
```

```
productos[0].asignarDatosConIVA();
```

```
productos[1].tipo = "drogueria";
```

```
productos[1].nombre = "Fenilefrina";
```

```
productos[1].asignarDatosConIVA();
```

```
productos[2].tipo = "drogueria";
```

```
productos[2].nombre = "Condomes";
```

```
productos[2].asignarDatosConIVA();
```

```
// Productos de Papeleria
```

```
productos[3].tipo = "papeleria";
```

```
productos[3].nombre = "Cuadernos";
```

```
productos[3].asignarDatosConIVA();
```

```
productos[4].tipo = "papeleria";
```

```
productos[4].nombre = "Colores";
```

```
productos[4].asignarDatosConIVA();
```

```
productos[5].tipo = "papeleria";
```

```
productos[5].nombre = "Carpetas";
```

```
productos[5].asignarDatosConIVA();
```

```
// Productos de Supermercado
```

```
productos[6].tipo = "supermercado";
```

```
productos[6].nombre = "Arroz";
```

```
productos[6].asignarDatosConIVA();
```

```
productos[7].tipo = "supermercado";
```

```
productos[7].nombre = "Leche";
```

```
productos[7].asignarDatosConIVA();
```

```
productos[8].tipo = "supermercado";
```

```
productos[8].nombre = "Azucar";
```

```
productos[8].asignarDatosConIVA();
```

```
// Mostramos la informacion de todos los productos
```

```
cout << "--- INFORMACION DE PRODUCTOS ---" << endl;
```

```
for (int i = 0; i < 9; i++) {
```

```
    productos[i].mostrar();
```

	<pre>         }          return 0;      } </pre>
--	--

## Requerimiento funcional 2

Nombre: vender Producto	<pre> #include &lt;iostream&gt;  #include &lt;vector&gt;  #include &lt;string&gt;  using namespace std;  struct Producto {      string nombre;      string tipo;      int cantidad;      float precio_base;      int vendidos;  };  vector&lt;Producto&gt; inventario = {      {"pan", "supermercado", 100, 1500.0f, 0}  }; </pre>
Parámetros: <p>Ninguno (la función no recibe ningún parámetro explícito; opera sobre la variable global inventario y obtiene la entrada del usuario a través de cin).</p>	
Retorno: <p>void (la función no devuelve ningún valor; modifica directamente el inventario y muestra la información en la consola).</p>	
Descripción: <p>Esta función permite al usuario registrar la venta de un producto. Toma el nombre del producto y la cantidad a vender como entrada del usuario, busca el producto en el inventario, actualiza el inventario, calcula el total de la venta, y muestra un mensaje de confirmación.</p>	

```
float total_ventas = 0.0f;
```

```
float calcularPrecioFinal(const Producto& p) {
```

```
    float iva = 0.0f;
```

```
    if (p.tipo == "papeleria") iva = 0.16f;
```

```
    else if (p.tipo == "supermercado") iva = 0.04f;
```

```
    else if (p.tipo == "drogueria") iva = 0.12f;
```

```
    return p.precio_base * (1 + iva);
```

```
}
```

```
void venderProducto() {
```

```
    string nombre;
```

```
    int cantidad;
```

```
    cout << "Nombre del producto: ";
```

```
    cin >> nombre;
```

```
    cout << "Cantidad a vender: ";
```

```
    cin >> cantidad;
```

```
    for (auto& p : inventario) {
```

```
        if (p.nombre == nombre && p.cantidad >= cantidad) {
```

```
            float venta = calcularPrecioFinal(p) * cantidad;
```

```
            p.cantidad -= cantidad;
```

```
            p.vendidos += cantidad;
```

	<pre> total_ventas += venta;  cout &lt;&lt; "Venta realizada: \$" &lt;&lt; venta &lt;&lt; endl;  return;  }  }  cout &lt;&lt; "Producto no encontrado o sin stock.\n";  }  int main() {      venderProducto();      return 0;  } </pre>
--	---

### Requerimiento funcional 3

Nombre: abastecer Producto	<pre> #include &lt;iostream&gt;  #include &lt;vector&gt;  #include &lt;string&gt;  using namespace std;  struct Producto {      string nombre;      int cantidad;  }; </pre>
Parámetros: Ninguno (la función no recibe ningún parámetro explícito; opera sobre la variable global inventario y obtiene la entrada del usuario a través de cin).	
Retorno: void (la función no devuelve ningún valor; modifica directamente el inventario y muestra la información en la consola).	
Descripción: Esta función permite al usuario agregar unidades de un producto al inventario. Toma el nombre del producto y la cantidad a agregar como entrada del usuario, busca el producto en	



el inventario, actualiza la cantidad del producto y muestra un mensaje de confirmación con el nuevo stock.

```
vector<Producto> inventario = {  
  
    {"lapicero", 10}  
  
};  
  
void abastecerProducto() {  
  
    string nombre;  
  
    int cantidad;  
  
    cout << "Producto a abastecer: ";  
  
    cin >> nombre;  
  
    cout << "Cantidad a agregar: ";  
  
    cin >> cantidad;  
  
  
    for (auto& p : inventario) {  
  
        if (p.nombre == nombre) {  
  
            p.cantidad += cantidad;  
  
            cout << "Nuevo stock: " << p.cantidad << endl;  
  
            Return;  
  
        }  
  
    }  
  
    cout << "Producto no encontrado.\n";  
}  
  
int main() {  
  
    abastecerProducto();  
  
    return 0;  
}
```

	}
--	---

## Requerimiento funcional 4

Nombre: cambiar Producto	<pre> #include &lt;iostream&gt;  #include &lt;vector&gt;  #include &lt;string&gt;  using namespace std;  struct Producto {      string nombre;      string tipo;      int cantidad_minima;      float precio_base;  };  vector&lt;Producto&gt; inventario = {      {"pan", "supermercado", 20, 1500.0f}  };  void cambiarProducto() {      string nombre;      cout &lt;&lt; "Nombre del producto a cambiar: ";      cin &gt;&gt; nombre; </pre>
Parámetros: Ninguno	
Retorno: void	
<p>Descripción:</p> <p>Esta función permite al usuario modificar la información de un producto existente en el inventario. Toma el nombre del producto a cambiar como entrada del usuario, busca el producto en el inventario, y luego solicita y actualiza el nombre, tipo, cantidad mínima y precio base del producto.</p>	

```
for (auto& p : inventario) {  
  
    if (p.nombre == nombre) {  
  
        cout << "Nuevo nombre: ";  
  
        cin >> p.nombre;  
  
        cout << "Nuevo tipo: ";  
  
        cin >> p.tipo;  
  
        cout << "Nueva cantidad mínima: ";  
  
        cin >> p.cantidad_minima;  
  
        cout << "Nuevo precio base: ";  
  
        cin >> p.precio_base;  
  
        cout << "Producto actualizado.\n";  
  
        return;  
  
    }  
  
}  
  
cout << "Producto no encontrado.\n";  
  
}  
  
int main() {  
  
    cambiarProducto();  
  
    return 0;  
  
}
```

## Requerimiento funcional 5

Nombre: estadísticas de Ventas	<pre> #include #include  #include using namespace std;  struct Producto { string nombre; int vendidos; };  vector inventario = { {"pan", 10}, {"lapicero", 5}, {"aspirina", 0} };  float total_ventas = 30000.0f;  void estadisticasVentas() { if (inventario.empty()) { cout &lt;&lt; "No hay productos.\n"; return; }  Producto* mas = &amp;inventario[0]; Producto* menos = &amp;inventario[0]; int total_unidades = 0;  for (auto&amp; p : inventario) {     if (p.vendidos &gt; mas-&gt;vendidos) mas = &amp;p;     if (p.vendidos &lt; menos-&gt;vendidos) menos = &amp;p;     total_unidades += p.vendidos; }  cout &lt;&lt; "\n--- Estadísticas ---\n"; cout &lt;&lt; "Más vendido: " &lt;&lt; mas-&gt;nombre &lt;&lt; " (" &lt;&lt; mas-&gt;vendidos &lt;&lt; ")\n"; cout &lt;&lt; "Menos vendido: " &lt;&lt; menos-&gt;nombre &lt;&lt; " (" &lt;&lt; menos-&gt;vendidos &lt;&lt; ")\n"; cout &lt;&lt; "Ingresos: \$" &lt;&lt; total_ventas &lt;&lt; endl;  if (total_unidades &gt; 0)     cout &lt;&lt; "Promedio por unidad: \$" &lt;&lt; total_ventas / total_unidades &lt;&lt; endl; else     cout &lt;&lt; "No se han vendido productos.\n"; </pre>
Parámetros: Ninguno	
Retorno: Void	
<p>Descripción:</p> <p>Esta función calcula y muestra las estadísticas de ventas. Verifica si el inventario está vacío, encuentra el producto más y menos vendido, calcula el total de unidades vendidas, y muestra el nombre del producto más y menos vendido, el total de ingresos, y el promedio de ingresos por unidad (si se han vendido productos).</p>	

	<pre>}  int main() { estadisticasVentas(); return 0;  }</pre>
--	---