

**UNIVERSIDAD NACIONAL DE SAN
AGUSTÍN DE AREQUIPA**

**FACULTAD DE INGENIERIA DE PRODUCCION Y
SERVICIOS**

**ESCUELA PROFESIONAL DE CIENCIAS DE LA
COMPUTACIÓN**



LABORATORIO 14 – Punteros de Punteros.

DOCENTE:

Enzo Edir Velásquez Lobatón

ALUMNO:

Owen Haziél Roque Sosa.

FECHA:

30/06/2022

Arequipa – Perú

Implemente un programa utilizando puntero a punteros que gestione los datos de stock de una tienda de ropa, la información a recoger será: código, nombre y precio del producto, asimismo la cantidad en stock. La tienda dispone de 10 productos distintos. El programa debe ser capaz de:

- Métodos adicionales implementados:

```
void menu(string ** accesorios, int tamano){
    char opt;
    while (opt != 'h'){
        void (*fptr)(string**, int) = nullptr;
        cout << "Slots disponibles = " << tamano - idxfiller << endl;
        cout << "*****ACCIONES DISPONIBLES*****\n";
        cout << " - fillItems          (a)\n";
        if (accesorios_was_deleted == 0){
            cout << " - deleteItems      (b)\n";
        }
        cout << " - DarDeAlta          (c)\n";
        cout << " - DarDeBaja         (d)\n";
        cout << " - searchItem        (e)\n";
        cout << " - modifyItem        (f)\n";
        cout << " - showAllItems      (g)\n";
        cout << " - Exit              (h)\n";
        cin >> opt;
        switch (opt) {
            case 'a':
                fptr = fillItems;
                fptr(accesorios, tamano);
                break;
            case 'b':
                fptr = deleteItems;
                fptr(accesorios, tamano);
                break;
            case 'c':
                fptr = DarDeAlta;
                fptr(accesorios, tamano);
                break;
            case 'd':
                fptr = DarDeBaja;
                fptr(accesorios, tamano);
                break;
            case 'e':
                fptr = searchItem;
                fptr(accesorios, tamano);
                break;
            case 'f':
                fptr = modifyItem;
                fptr(accesorios, tamano);
                break;
            case 'g':
                fptr = showAllItems;
                fptr(accesorios, tamano);
                break;
            case 'h':
                // en caso cree los items y salga directamente del programa
                // sin liberar memoria (no elegi la opcion b antes)
                for (int i = 0; i < idxfiller; ++i)
                    delete [] accesorios[i];
                delete [] accesorios;
                cout << "Saliendo del programa...\n";
                break;
            default:
                cout << "Opcion no valida!\n";
                system("pause");
                system("cls");
        }
    }
    cout << "Fin del Programa.\n";
}
```

```

void fillItems(string ** accesorios, int tamaño){
    string codigo ,nombre;
    float precio;
    int stock, estado = 0; // 1 = Activo | 0 = Nulo
    int n;
    int rangoDisponible = tamaño-idxfiller;
    if (rangoDisponible == 0) {
        cout << "Lista de articulos LLENA!\n";
        system("pause");
        system("cls");
        return;
    }
    cout << "Cant. articulos a crear (<= a " << rangoDisponible << "): ";
    cin >> n;
    if (n >= rangoDisponible){
        cout << "Cant. debe ser <= a " << rangoDisponible << "!\n";
        cin.clear();
        system("pause");
        system("cls");
        return;
    }
    cout << "Por favor ingrese la siguiente información de los accesorios: \n";
    for(int i = 0; i < n; i++){
        // ncolumnas = nAtributos
        // codigo, nombre, precio
        // stock, estado
        const int cols = 5;
        accesorios[i] = new string[cols];
        cout << "\n***** Item #" << i + 1 << "*****:\n";
        cin.ignore();
        cout << "Codigo: ";
        getline(cin,codigo);
        cout << "Nombre: ";
        getline(cin,nombre);
        cout << "Precio (S/): ";
        while(!(cin >> precio)){ // validar precio
            cout << "Stock: ";
            while(!(cin >> stock)){ // validar stock
                do { // validar estado
                    //cin.ignore();
                    string _precio = std::to_string(precio);
                    string _stock = std::to_string(stock);
                    string _estado = std::to_string(estado);
                    (*(accesorios+i)+0) = codigo;
                    (*(accesorios+i)+1) = nombre;
                    (*(accesorios+i)+2) = _precio;
                    (*(accesorios+i)+3) = _stock;
                    (*(accesorios+i)+4) = _estado;
                }
                idxfiller+=n;
                accesorios_was_deleted = 0;
                cout << "HECHO!" << endl;
                system("pause");
                system("cls");
            }
        }
    }
}

```

```

void deleteItems(string ** accesorios, int tamaño){
    // verificar
    if (accesorios_was_deleted == 1){
        cout << "La lista de articulos YA SE ELIMINO.\n";
        system("pause");
        system("cls");
        return;
    }
    for (int i = 0; i < idxfiller; ++i)
        delete [] accesorios[i];
    accesorios_was_deleted = 1;
    idxfiller = 0;
    cout << "HECHO!" << endl;
    system("pause");
    system("cls");
}

```

```

// var. global para saber el idx/limite del array de articulos (<= a 10)
int idxfiller = 0;
// 0 = no se borro | 1 = se borro/está vacia
int accesorios_was_deleted = 1;

```

```

int main() {
    // matriz (punteros de punteros)
    string** accesorios;
    const int tamaño = 10; // 10 tamaño maximo
    // tamaño = nfilas = nItems
    // se crea accesorios por default
    // se eliminará de todas formas al
    // terminar la funcion menu()
    accesorios = new string*[tamaño];
    menu(accesorios, tamaño);
    return 0;
}

```

```

void showAllItems(string ** accesorios, int tamaño){
    if (idxfiller == 0){
        cout << "Lista de articulos VACIA!\n";
    }
    else {
        for (int i = 0; i < idxfiller; ++i){
            // mostrar informacion del articulo
            cout << "Articulo # " << i+1 << ":\n";
            cout << "Codigo: " << (*(accesorios+i)+0) << endl;
            cout << "Nombre: " << (*(accesorios+i)+1) << endl;
            cout << "Precio: S/ " << (*(accesorios+i)+2) << endl;
            cout << "Stock: " << (*(accesorios+i)+3) << endl;
            // comparacion con op. ternario
            string _estado = (*(accesorios+i)+4) == "0" ? "Anulado" : "Activo";
            cout << "Estado: " << _estado << endl;
            cout << "-----\n\n";
        }
        cout << "HECHO!\n";
    }
    system("pause");
    system("cls");
}

```

1. Dar de alta un producto nuevo.

```
void DarDeAlta(string ** accesorios, int tamano){
    if (idxfiller == 10){
        cout << "Lista de articulos LLENA!\n";
        system("pause");
        system("cls");
        return;
    }
    else {
        string codigo ,nombre;
        float precio;
        int stock, estado = 0; // 1 = Activo | 0 = Nulo
        const int cols = 5;
        accesorios[idxfiller] = new string[cols];
        cout << "\n***** Item #" << idxfiller + 1 << "*****\n";
        cin.ignore();
        cout << "Codigo: ";
        getline(cin,codigo);
        cout << "Nombre: ";
        getline(cin,nombre);
        cout << "Precio (S/): ";
        while(!(cin >> precio)){
            cout << "Formato incorrecto!\n";
            cin.clear();
            cin.ignore(numeric_limits<std::streamsize>::max(), '\n');
            cout << "Precio: ";
        }
        cout << "Stock: ";
        while(!(cin >> stock)){
            cout << "Formato incorrecto!\n";
            cin.clear();
            cin.ignore(numeric_limits<std::streamsize>::max(), '\n');
            cout << "Stock: ";
        }
        do {
            cout << "Estado (0 = anulado | 1 = activo): ";
            cin >> estado;
        } while (estado != 0 && estado != 1);
        //cin.ignore();
        string _precio = std::to_string(precio);
        string _stock = std::to_string(stock);
        string _estado = std::to_string(estado);
        *(*accesorios+idxfiller)+0) = codigo;
        *(*accesorios+idxfiller)+1) = nombre;
        *(*accesorios+idxfiller)+2) = _precio;
        *(*accesorios+idxfiller)+3) = _stock;
        *(*accesorios+idxfiller)+4) = _estado;
        idxfiller++;
        accesorios_was_deleted = 0;
        cout << "HECHO!" << endl;
        system("pause");
        system("cls");
    }
}
```

```
Slots disponibles = 10
*****ACCIONES DISPONIBLES*****
- fillItems      (a)
- DarDeAlta      (c)
- DarDeBaja      (d)
- searchItem     (e)
- modifyItem     (f)
- showAllItems   (g)
- Exit           (h)
c

***** Item #1*****:
Codigo: lkjasfha45
Nombre: Bufanda Azul
Precio (S/): 35
Stock: 12
Estado (0 = anulado | 1 = activo): 1
HECHO!
Presione una tecla para continuar . . . █
```

```
Slots disponibles = 9
*****ACCIONES DISPONIBLES*****
- fillItems      (a)
- deleteItems    (b)
- DarDeAlta      (c)
- DarDeBaja      (d)
- searchItem     (e)
- modifyItem     (f)
- showAllItems   (g)
- Exit           (h)
g
Articulo # 1:
Codigo: lkjasfha45
Nombre: Bufanda Azul
Precio: S/ 35.000000
Stock: 12
Estado: Activo
-----

HECHO!
Presione una tecla para continuar . . . █
```

2. Dar de baja a un producto. (No hay stock).

```
void DarDeBaja(string ** accesorios, int tamano){
    string _codigo; int search_status = 0;
    cout << "Ingrese codigo del articulo a dar de baja: ";
    cin.ignore();
    getline(cin, _codigo);
    for (int i = 0; i < idxfiller; ++i){
        if (*(accesorios+i)+0) == _codigo){
            (*(accesorios+i)+3) = "0"; // no hay stock
            (*(accesorios+i)+4) = "0"; // anulado
            cout << "HECHO!";
            search_status = 1;
            break;
        }
    }
    if (search_status == 0){
        cout << "No se encontro el codigo ingresado.\n";
        cout << "Intente Nuevamente.\n";
    }
    system("pause");
    system("cls");
}
```

Continuando con el registro creado anteriormente:

```
Slots disponibles = 9
*****ACCIONES DISPONIBLES*****
- fillItems      (a)
- deleteItems    (b)
- DarDeAlta      (c)
- DarDeBaja      (d)
- searchItem     (e)
- modifyItem     (f)
- showAllItems   (g)
- Exit           (h)
d
Ingrese codigo del articulo a dar de baja: lkjasfha45
HECHO!Presione una tecla para continuar . . .
```

```
Slots disponibles = 9
*****ACCIONES DISPONIBLES*****
- fillItems      (a)
- deleteItems    (b)
- DarDeAlta      (c)
- DarDeBaja      (d)
- searchItem     (e)
- modifyItem     (f)
- showAllItems   (g)
- Exit           (h)
g
Articulo # 1:
Codigo: lkjasfha45
Nombre: Bufanda Azul
Precio: S/ 35.000000
Stock: 0
Estado: Anulado
-----
HECHO!
Presione una tecla para continuar . . . _
```


3. Buscar un producto por su nombre o código.

```
void searchItem(string ** accesorios, int tamaño){
    string _valor; int search_status = 0;
    cout << "Ingrese codigo o nombre del articulo a buscar: ";
    cin.ignore();
    getline(cin, _valor);
    for (int i = 0; i < idxfiller; ++i){
        if ((*(*accesorios+i)+0) == _valor) || ((*(*accesorios+i)+1) == _valor)){
            // mostrar informacion del articulo
            cout << "Articulo # " << i+1 << ":\n";
            cout << "Nombre: " << *(*(*accesorios+i)+0) << endl;
            cout << "Codigo: " << *(*(*accesorios+i)+1) << endl;
            cout << "Precio: S/ " << *(*(*accesorios+i)+2) << endl;
            cout << "Stock: " << *(*(*accesorios+i)+3) << endl;
            // comparacion con op. ternario
            string _estado = ((*(*(*accesorios+i)+4) == "0") ? "Anulado" : "Activo");
            cout << "Estado: " << _estado << endl;
            search_status = 1;
            break;
        }
    }
    if (search_status == 0){
        cout << "No se encontro ningun resultado.\n";
        cout << "Intente Nuevamente.\n";
    }
    cout << "HECHO!\n";
    system("pause");
    system("cls");
}
```

Búsqueda por código:

```
Slots disponibles = 9
*****ACCIONES DISPONIBLES*****
- fillItems      (a)
- deleteItems    (b)
- DarDeAlta      (c)
- DarDeBaja      (d)
- searchItem     (e)
- modifyItem     (f)
- showAllItems   (g)
- Exit           (h)
e
Ingrese codigo o nombre del articulo a buscar: lkjasfha45
Articulo # 1:
Nombre: lkjasfha45
Codigo: Bufanda Azul
Precio: S/ 35.000000
Stock: 0
Estado: Anulado
HECHO!
Presione una tecla para continuar . . . _
```

Búsqueda por nombre:

```
Slots disponibles = 9
*****ACCIONES DISPONIBLES*****
- fillItems      (a)
- deleteItems    (b)
- DarDeAlta      (c)
- DarDeBaja      (d)
- searchItem     (e)
- modifyItem     (f)
- showAllItems   (g)
- Exit           (h)
e
Ingrese codigo o nombre del articulo a buscar: Bufanda Azul
Articulo # 1:
Nombre: lkjasfha45
Codigo: Bufanda Azul
Precio: S/ 35.000000
Stock: 0
Estado: Anulado
HECHO!
Presione una tecla para continuar . . .
```

4. Modificar el nombre, precio o cantidad de un producto dado.

```
void modifyItem(string ** accesorios, int tamano){
    string _valor; int search_status = 0; int was_modified = 0; int was_cancelled = 0;
    cout << "Ingrese codigo o nombre del articulo a buscar: ";
    cin.ignore();
    getline(cin, _valor);
    for (int i = 0; i < idxfiller; ++i){
        if ((*(*accesorios+i)+0) == _valor) || ((*(*accesorios+i)+1) == _valor){
            // modificar campos
            cout << "\n***** Articulo a modificar => Item #" << i + 1 << "*****:\n";
            cout << "Nombre: " << *(*(*accesorios+i)+0) << endl;
            cout << "Codigo: " << *(*(*accesorios+i)+1) << endl;
            cout << "Precio: $/ " << *(*(*accesorios+i)+2) << endl;
            cout << "Stock: " << *(*(*accesorios+i)+3) << endl;
            // comparacion con op. ternario
            string _estado = ((*(*accesorios+i)+4) == "0") ? "Anulado" : "Activo";
            cout << "Estado: " << _estado << endl;
            cout << endl;
            search_status = 1;
            cout << "Escoja los campos que desea modificar: \n";
            cout << "(Solo puede escoger uno a la vez) \n";
            cout << " ->codigo (c)\n";
            cout << " ->nombre (n)\n";
            cout << " ->precio (p)\n";
            cout << " ->stock (s)\n";
            cout << " ->estado (e)\n";
            cout << " ->cancelar (x)\n";
            char opt;
            cout << "Ingrese una opcion: ";
            cin >> opt;
            switch (opt){
                case 'c':
                    { // modifica codigo
                        string ncodigo;
                        cin.ignore();
                        cout << "Nuevo Codigo: ";
                        getline(cin, ncodigo);
                        *(*(*accesorios+i)+0) = ncodigo;
                        was_modified = 1;
                        break;
                    }
                case 'n':
                    { // modifica nombre
                        string nnombre;
                        cin.ignore();
                        cout << "Nuevo Nombre: ";
                        getline(cin, nnombre);
                        *(*(*accesorios+i)+1) = nnombre;
                        was_modified = 1;
                        break;
                    }
            }
        }
    }
}
```

Continuación del código a la siguiente pagina...

```

case 'p':
{ // modifica precio
    float nprecio;
    cout << "Nuevo Precio: ";
    while(!(cin >> nprecio)){
        cout << "Formato incorrecto!\n";
        cin.clear();
        cin.ignore(numeric_limits<std::streamsize>::max(), '\n');
        cout << "Nuevo Precio: ";
    }
    string n_precio = std::to_string(nprecio);
    (*(accesorios+i)+2) = n_precio;
    was_modified = 1;
    break;
}

case 's':
{ // modifica stock
    int nstock;
    cout << "Nuevo Stock: ";
    while(!(cin >> nstock)){
        cout << "Formato incorrecto!\n";
        cin.clear();
        cin.ignore(numeric_limits<std::streamsize>::max(), '\n');
        cout << "Nuevo Stock: ";
    }
    string n_stock = std::to_string(nstock);
    (*(accesorios+i)+3) = n_stock;
    was_modified = 1;
    break;
}

case 'e':
{ // modifica estado
    int nestado;
    do {
        cout << "Nuevo Estado (0 = anulado | 1 = activo): ";
        cin >> nestado;
    } while (nestado != 0 && nestado != 1);
    //cin.ignore();
    string n_estado = std::to_string(nestado);
    (*(accesorios+i)+4) = n_estado;
    was_modified = 1;
    break;
}

case 'x':
{ // cancela
    was_cancelled = 1;
    break;
}

default:
{ // ninguna de las anteriores
    cout << "Campo no Validado!\n";
}
}

```

```

    }
}

if (search_status == 0){ // si no encuentro el articulo
    cout << "No se encontro ningun resultado.\n";
    cout << "Intente Nuevamente.\n";
    system("pause");
    system("cls");
    return;
}

else if (was_modified == 0){ // si el articulo fue modificado
    if (was_cancelled == 1){ // si fue cancelado
        cout << "Modificacion Cancelada!\n";
    } else { // sino, no llego a concretarse la modificacion
        cout << "No se logró modificar el articulo.\n";
        cout << "Intente Nuevamente.\n";
    }

    system("pause");
    system("cls");
    return;
}

else { // modificacion exitosa
    cout << "HECHO!\n";
    system("pause");
    system("cls");
    return;
}
}

```

```
Slots disponibles = 9
*****ACCIONES DISPONIBLES*****
- fillItems      (a)
- deleteItems    (b)
- DarDeAlta      (c)
- DarDeBaja      (d)
- searchItem     (e)
- modifyItem     (f)
- showAllItems   (g)
- Exit           (h)
f
Ingrese codigo o nombre del articulo a buscar: Bufanda Azul

***** Articulo a modificar => Item #1*****:
Nombre: lkjasfha45
Codigo: Bufanda Azul
Precio: S/ 35.000000
Stock: 0
Estado: Anulado

Escoja los campos que desea modificar:
(Solo puede escoger uno a la vez)
->codigo (c)
->nombre (n)
->precio (p)
->stock (s)
->estado (e)
->cancelar (x)
Ingrese una opcion:
```

```
Ingrese una opcion: e
Nuevo Estado (0 = anulado | 1 = activo): 1
HECHO!
Presione una tecla para continuar . . . _
```

```

Slots disponibles = 9
*****ACCIONES DISPONIBLES*****
- fillItems      (a)
- deleteItems    (b)
- DarDeAlta      (c)
- DarDeBaja      (d)
- searchItem     (e)
- modifyItem     (f)
- showAllItems   (g)
- Exit           (h)

g
Articulo # 1:
Codigo: lkjasfha45
Nombre: Bufanda Azul
Precio: S/ 35.000000
Stock: 0
Estado: Activo
-----

HECHO!
Presione una tecla para continuar . . . █

```

Fin programa (elimina la matriz **accesorios):

```

Slots disponibles = 9
*****ACCIONES DISPONIBLES*****
- fillItems      (a)
- deleteItems    (b)
- DarDeAlta      (c)
- DarDeBaja      (d)
- searchItem     (e)
- modifyItem     (f)
- showAllItems   (g)
- Exit           (h)

h
Saliendo del programa...
Fin del Programa.

<< El programa ha finalizado: codigo de salida: 0 >>
<< Presione enter para cerrar esta ventana >>

```