

Laboratorio 15

Patrones de diseño: Builder

Nombre: Gabriel Fernando Rodriguez Cutimbo

CUI: 20212157

Grupo: B

Repositorio GitHub:

https://github.com/gaco123/EPCC_CCII.git

1. Ejercicios

Resolver los siguientes ejercicios planteados:

1. El alumno deberá de implementar un conjunto de clases que permita seleccionar las piezas de un automóvil, es decir, se podrán tener componentes a disposición del cliente (puertas, llantas, timón, asientos, motor, espejos, vidrios, etc.). Del cual el cliente puede indicar que características de color puede tener cada pieza. Al final mostrar opciones al Cliente o permitirle que él pueda escoger las piezas e indicar el color. Utilizar el patrón Builder.

*Pista, en lugar de trabajar el producto con una lista de componentes, se puede alojar una estructura o clase.

Código “Lab_15.cpp”:

```
//Laboratorio N°15
```

```
//Alumno: Gabriel Fernando Rodriguez Cutimbo
```

```
/*
```

```
Ejercicio:
```

1. El alumno deberá de implementar un conjunto de clases que permita seleccionar las piezas de un automóvil, es decir, se podrán tener componentes a disposición del cliente (puertas, llantas, timón, asientos, motor, espejos, vidrios, etc.). Del cual el cliente puede indicar que características de color puede tener cada pieza. Al final mostrar opciones al Cliente o permitirle que él pueda escoger las piezas e indicar el color. Utilizar el patrón Builder.

*Pista, en lugar de trabajar el producto con una lista de componentes, se puede alojar una estructura o clase.

```
*/
```

```
#include <iostream>
#include <vector>
using namespace std;

//Este es el producto realizado por la clase "BuilderEspecifico"
class Automovil{
public:
    vector<string> componentes;
    void ListaComp()const {
        cout << "Componentes: ";
        for (size_t i = 0; i < componentes.size(); i++) {
            if (componentes[i] == componentes.back()) {
                cout << componentes[i];
            }
            else {
                cout << componentes[i] << ", ";
            }
        }
        cout << "\n\n";
    }
};

/*Nombres de las funciones que tendrán que describir cada
clase "BuilderEspecifico"*/
class IBUILDER{
public:
    virtual ~IBUILDER() {}
    virtual void ProducirPuerta(string) const = 0;
    virtual void ProducirLlanta(string) const = 0;
    virtual void ProducirTimon(string) const = 0;
    virtual void ProducirAsiento(string) const = 0;
    virtual void ProducirMotor(string) const = 0;
    virtual void ProducirEspejo(string) const = 0;
    virtual void ProducirGuantera(string) const = 0;
    virtual void ProducirCajaCambios(string) const = 0;
    virtual void ProducirVentana(string) const = 0;
    virtual void ProducirBateria(string) const = 0;
};

/*Descripción de cada función de "IBUILDER" (en este caso solo
hay una clase "BuilderEspecifico", pero, pueden a ver más con otras
instrucciones para las funciones nombradas por "IBUILDER"*/
class BuilderEspecifico : public IBUILDER{
private:
    Automovil* product;
public:
    BuilderEspecifico() {
        this->Reset();
    }
    ~BuilderEspecifico() {
```

```
        delete product;
    }
    void Reset() {
        this->product = new Automovil();
    }

    void ProducirPuerta(string color)const override{
        string temp = "Puerta["+color+"]";
        this->product->componentes.push_back(temp);
    }
    void ProducirLlanta(string color)const override{
        string temp = "Llanta["+color+"]";
        this->product->componentes.push_back(temp);
    }
    void ProducirTimon(string color)const override{
        string temp = "Timon["+color+"]";
        this->product->componentes.push_back(temp);
    }
    void ProducirAsiento(string color)const override{
        string temp = "Asiento["+color+"]";
        this->product->componentes.push_back(temp);
    }
    void ProducirMotor(string color)const override{
        string temp = "Motor["+color+"]";
        this->product->componentes.push_back(temp);
    }
    void ProducirEspejo(string color)const override{
        string temp = "Espejo["+color+"]";
        this->product->componentes.push_back(temp);
    }
    void ProducirGuantera(string color)const override{
        string temp = "Guantera["+color+"]";
        this->product->componentes.push_back(temp);
    }
    void ProducirCajaCambios(string color)const override{
        string temp = "Caja_de_Cambios["+color+"]";
        this->product->componentes.push_back(temp);
    }
    void ProducirVentana(string color)const override{
        string temp = "Ventana["+color+"]";
        this->product->componentes.push_back(temp);
    }
    void ProducirBateria(string color)const override{
        string temp = "Batería["+color+"]";
        this->product->componentes.push_back(temp);
    }
}

Automovil* GetProducto(){
    Automovil* resultado = this->product;
    this->Reset();
    return resultado;
}
```

```
    }  
};  
  
/*No existe la clase "Director", ya que, en este caso no se necesita un producto  
prefabricado*/  
  
//Programa en sí  
int main(){  
    //Memoria Basica  
    BuilderEspecifico* builder = new BuilderEspecifico(); //IBuilder => BuilderEspecifico  
    Automovil* p; //Producto  
    int op_p; //Se encarga de las opciones del menú principal  
  
    //Se encargan de las opciones de los diferentes submenús de colores  
    int op_ps1;  
    int op_ps2;  
    int op_ps3;  
    int op_ps4;  
    int op_ps5;  
    int op_ps6;  
    int op_ps7;  
    int op_ps8;  
    int op_ps9;  
    int op_ps10;  
  
    bool comp_ps11 = true; //Para comprobar que al menos una vez se ha construido el vehiculo  
  
    //Esta va a ser el "Cliente" ya que con este menú se va a armar un auto desde cero  
    while(true){  
        cout<<"MENU PARA FABRICAR UN AUTOMOVIL\n";  
        cout<<"1. Agregar puerta\n";  
        cout<<"2. Agregar llanta\n";  
        cout<<"3. Agregar timón\n";  
        cout<<"4. Agregar asiento\n";  
        cout<<"5. Agregar motor\n";  
        cout<<"6. Agregar espejo\n";  
        cout<<"7. Agregar guantera\n";  
        cout<<"8. Agregar caja de cambios\n";  
        cout<<"9. Agregar ventana\n";  
        cout<<"10. Agregar batería\n";  
        cout<<"11. Construir vehiculo(Solo usar una vez terminado de agregar todo)\n";  
        cout<<"12. Mostrar componentes actuales(Usé primero la opción 11)\n";  
        cout<<"13. Salir\n";  
        do{  
            cout<<"Ingrese el número de la opción: ";  
            cin>>op_p;  
  
            if(op_p==1||op_p==2||op_p==3||op_p==4||op_p==5||op_p==6||op_p==7||op_p==8||op_p==9||op_p==10||  
            op_p==11||op_p==12||op_p==13){  
                break;  
            }  
        }  
    }  
}
```

```
        else{
            cout<<"Error, ingrese nuevamente el número de opción...\n\n";
        }
    }while(true);
    cout<<endl;
    //Puerta
    if(op_p==1){
        cout<<"MENU PARA ELEGIR EL COLOR DE LA PUERTA\n";
        cout<<"1. Azul Metalico\n";
        cout<<"2. Rojo Metalico\n";
        cout<<"3. Blanco\n";
        do{
            cout<<"Ingrese el número de la opción: ";
            cin>>op_ps1;
            if(op_ps1==1||op_ps1==2||op_ps1==3){
                break;
            }
            else{
                cout<<"Error, ingrese nuevamente el número de opción...\n\n";
            }
        }while(true);
        cout<<endl;
        if(op_ps1==1){
            builder->ProducirPuerta("Azul Metalico");
            cout<<"Se agrego una puerta color \"Azul Metalico\"\n\n";
        }
        else if(op_ps1==2){
            builder->ProducirPuerta("Rojo Metalico");
            cout<<"Se agrego una puerta color \"Rojo Metalico\"\n\n";
        }
        else if(op_ps1==3){
            builder->ProducirPuerta("Blanco");
            cout<<"Se agrego una puerta color \"Blanco\"\n\n";
        }
    }
    //Llanta
    if(op_p==2){
        cout<<"MENU PARA ELEGIR EL COLOR DE LA LLANTA\n";
        cout<<"1. Negro\n";
        cout<<"2. Gris\n";
        cout<<"3. Amarillo\n";
        cout<<"4. Blanco\n";
        do{
            cout<<"Ingrese el número de la opción: ";
            cin>>op_ps2;
            if(op_ps2==1||op_ps2==2||op_ps2==3||op_ps2==4){
                break;
            }
            else{
                cout<<"Error, ingrese nuevamente el número de opción...\n\n";
            }
        }
```

```
}while(true);
cout<<endl;
if(op_ps2==1){
    builder->ProducirLlanta("Negro");
    cout<<"Se agrego una llanta color \"Negro\\\"\\n\\n";
}
else if(op_ps2==2){
    builder->ProducirLlanta("Gris");
    cout<<"Se agrego una llanta color \"Gris\\\"\\n\\n";
}
else if(op_ps2==3){
    builder->ProducirLlanta("Amarillo");
    cout<<"Se agrego una llanta color \"Amarillo\\\"\\n\\n";
}
else if(op_ps2==4){
    builder->ProducirLlanta("Blanco");
    cout<<"Se agrego una llanta color \"Blanco\\\"\\n\\n";
}
}
//Timon
if(op_p==3){
    cout<<"MENU PARA ELEGIR EL COLOR DEL TIMON\\n";
    cout<<"1. Café\\n";
    cout<<"2. Negro\\n";
    do{
        cout<<"Ingrese el número de la opción: ";
        cin>>op_ps3;
        if(op_ps3==1||op_ps3==2){
            break;
        }
        else{
            cout<<"Error, ingrese nuevamente el número de opción...\\n\\n";
        }
    }while(true);
    cout<<endl;
    if(op_ps3==1){
        builder->ProducirTimon("Café");
        cout<<"Se agrego un timón color \"Café\\\"\\n\\n";
    }
    else if(op_ps3==2){
        builder->ProducirTimon("Negro");
        cout<<"Se agrego un timón color \"Negro\\\"\\n\\n";
    }
}
//Asiento
if(op_p==4){
    cout<<"MENU PARA ELEGIR EL COLOR DEL ASIENTO\\n";
    cout<<"1. Café\\n";
    cout<<"2. Negro\\n";
    cout<<"3. Blanco\\n";
    do{
```

```
        cout<<"Ingrese el número de la opción: ";
        cin>>op_ps4;
        if(op_ps4==1||op_ps4==2||op_ps4==3){
            break;
        }
        else{
            cout<<"Error, ingrese nuevamente el número de opción...\n\n";
        }
    } while(true);
    cout<<endl;
    if(op_ps4==1){
        builder->ProducirAsiento("Café");
        cout<<"Se agrego un asiento color \"Café\"\n\n";
    }
    else if(op_ps4==2){
        builder->ProducirAsiento("Negro");
        cout<<"Se agrego un asiento color \"Negro\"\n\n";
    }
    else if(op_ps4==2){
        builder->ProducirAsiento("Blanco");
        cout<<"Se agrego un asiento color \"Blanco\"\n\n";
    }
}
//Motor
if(op_p==5){
    cout<<"MENU PARA ELEGIR EL COLOR DEL MOTOR\n";
    cout<<"1. Negro\n";
    cout<<"2. Blanco\n";
    do{
        cout<<"Ingrese el número de la opción: ";
        cin>>op_ps5;
        if(op_ps5==1||op_ps5==2){
            break;
        }
        else{
            cout<<"Error, ingrese nuevamente el número de opción...\n\n";
        }
    } while(true);
    cout<<endl;
    if(op_ps5==1){
        builder->ProducirMotor("Negro");
        cout<<"Se agrego un motor color \"Negro\"\n\n";
    }
    else if(op_ps5==2){
        builder->ProducirMotor("Blanco");
        cout<<"Se agrego un motor color \"Blanco\"\n\n";
    }
}
//Espejo
if(op_p==6){
    cout<<"MENU PARA ELEGIR EL COLOR DEL ESPEJO\n";
```

```
cout<<"1. Azul Metalico\n";
cout<<"2. Rojo Metalico\n";
cout<<"3. Blanco\n";
do{
    cout<<"Ingrese el número de la opción: ";
    cin>>op_ps6;
    if(op_ps6==1||op_ps6==2||op_ps6==3){
        break;
    }
    else{
        cout<<"Error, ingrese nuevamente el número de opción...\n\n";
    }
}while(true);
cout<<endl;
if(op_ps6==1){
    builder->ProducirEspejo("Azul Metalico");
    cout<<"Se agrego un espejo color \"Azul Metalico\"\n\n";
}
else if(op_ps6==2){
    builder->ProducirEspejo("Rojo Metalico");
    cout<<"Se agrego un espejo color \"Rojo Metalico\"\n\n";
}
else if(op_ps6==3){
    builder->ProducirEspejo("Blanco");
    cout<<"Se agrego un espejo color \"Blanco\"\n\n";
}
}
//Guantera
if(op_p==7){
    cout<<"MENU PARA ELEGIR EL COLOR DE LA GUANTERA\n";
    cout<<"1. Negro\n";
    cout<<"2. Gris\n";
    do{
        cout<<"Ingrese el número de la opción: ";
        cin>>op_ps7;
        if(op_ps7==1||op_ps7==2){
            break;
        }
        else{
            cout<<"Error, ingrese nuevamente el número de opción...\n\n";
        }
    }while(true);
    cout<<endl;
    if(op_ps7==1){
        builder->ProducirGuantera("Negro");
        cout<<"Se agrego una guantero color \"Negro\"\n\n";
    }
    else if(op_ps7==2){
        builder->ProducirGuantera("Gris");
        cout<<"Se agrego una guantero color \"Gris\"\n\n";
    }
}
```



```
}
//Caja de Cambios
if(op_p==8){
    cout<<"MENU PARA ELEGIR EL COLOR DE LA CAJA DE CAMBIOS\n";
    cout<<"1. Café\n";
    cout<<"2. Negro\n";
    cout<<"3. Blanco\n";
    do{
        cout<<"Ingrese el número de la opción: ";
        cin>>op_ps8;
        if(op_ps8==1||op_ps8==2||op_ps8==3){
            break;
        }
        else{
            cout<<"Error, ingrese nuevamente el número de opción...\n\n";
        }
    } while(true);
    cout<<endl;
    if(op_ps8==1){
        builder->ProducirCajaCambios("Café");
        cout<<"Se agrego una caja de cambios color \"Café\"\n\n";
    }
    else if(op_ps8==2){
        builder->ProducirCajaCambios("Negro");
        cout<<"Se agrego una caja de cambios color \"Negro\"\n\n";
    }
    else if(op_ps8==3){
        builder->ProducirCajaCambios("Blanco");
        cout<<"Se agrego una caja de cambios color \"Blanco\"\n\n";
    }
}
//Ventana
if(op_p==9){
    cout<<"MENU PARA ELEGIR EL COLOR DE LA VENTANA\n";
    cout<<"1. Transparente\n";
    cout<<"2. Opaco\n";
    do{
        cout<<"Ingrese el número de la opción: ";
        cin>>op_ps9;
        if(op_ps9==1||op_ps9==2){
            break;
        }
        else{
            cout<<"Error, ingrese nuevamente el número de opción...\n\n";
        }
    } while(true);
    cout<<endl;
    if(op_ps9==1){
        builder->ProducirVentana("Transparente");
        cout<<"Se agrego una ventana color \"Transparente\"\n\n";
    }
}
```

```
        else if(op_ps9==2){
            builder->ProducirVentana("Opaco");
            cout<<"Se agrego una ventana color \"Opaco\\\"\\n\\n";
        }
    }
    //Batería
    if(op_p==10){
        cout<<"MENU PARA ELEGIR EL COLOR DE LA BATERIA\\n";
        cout<<"1. Gris\\n";
        cout<<"2. Blanco\\n";
        do{
            cout<<"Ingrese el número de la opción: ";
            cin>>op_ps10;
            if(op_ps10==1||op_ps10==2){
                break;
            }
            else{
                cout<<"Error, ingrese nuevamente el número de opción...\\n\\n";
            }
        }while(true);
        cout<<endl;
        if(op_ps10==1){
            builder->ProducirBateria("Gris");
            cout<<"Se agrego una batería color \"Gris\\\"\\n\\n";
        }
        else if(op_ps10==2){
            builder->ProducirBateria("Blanco");
            cout<<"Se agrego una batería color \"Blanco\\\"\\n\\n";
        }
    }
    //Construir Vehiculo
    else if(op_p==11){
        if(comp_ps11==false){
            delete p;
            p = builder->GetProducto();
        }
        else{
            p = builder->GetProducto();
            comp_ps11=false;
        }
        cout<<"Vehiculo Construido\\n\\n";
    }
    //Ver componentes del auto construido
    else if(op_p==12){
        p->ListaComp();
    }
    //Salir del programa
    else if(op_p==13){
        if(comp_ps11==false){
            delete p;
        }
    }
```

```
        delete builder;  
        break;  
    }  
}  
  
return 0;  
}
```

Funcionamiento (Agregar un componente):

```
MENU PARA FABRICAR UN AUTOMOVIL  
1. Agregar puerta  
2. Agregar llanta  
3. Agregar timón  
4. Agregar asiento  
5. Agregar motor  
6. Agregar espejo  
7. Agregar guantera  
8. Agregar caja de cambios  
9. Agregar ventana  
10. Agregar batería  
11. Construir vehiculo(Solo usar una vez terminado de agregar todo)  
12. Mostrar componentes actuales(Usé primero la opción 11)  
13. Salir  
Ingrese el número de la opción: 1  
  
MENU PARA ELEGIR EL COLOR DE LA PUERTA  
1. Azul Metalico  
2. Rojo Metalico  
3. Blanco  
Ingrese el número de la opción: 1  
  
Se agrego una puerta color "Azul Metalico"
```

Funcionamiento (Construir Vehículo):

```
MENU PARA FABRICAR UN AUTOMOVIL
1. Agregar puerta
2. Agregar llanta
3. Agregar timón
4. Agregar asiento
5. Agregar motor
6. Agregar espejo
7. Agregar guantera
8. Agregar caja de cambios
9. Agregar ventana
10. Agregar batería
11. Construir vehiculo(Solo usar una vez terminado de agregar todo)
12. Mostrar componentes actuales(Usé primero la opción 11)
13. Salir
Ingrese el número de la opción: 11

Vehiculo Construido
```

Funcionamiento (Mostrar Componentes):

```
MENU PARA FABRICAR UN AUTOMOVIL
1. Agregar puerta
2. Agregar llanta
3. Agregar timón
4. Agregar asiento
5. Agregar motor
6. Agregar espejo
7. Agregar guantera
8. Agregar caja de cambios
9. Agregar ventana
10. Agregar batería
11. Construir vehiculo(Solo usar una vez terminado de agregar todo)
12. Mostrar componentes actuales(Usé primero la opción 11)
13. Salir
Ingrese el número de la opción: 12

Componentes: Puerta[Azul Metalico]
```

2. Entregables

Al final estudiante deberá:

1. Compactar el código elaborado y subirlo al aula virtual de trabajo. Agregue sus datos personales como comentario en cada archivo de código elaborado.
2. Elaborar un documento que incluya tanto el código como capturas de pantalla de la ejecución del programa. Este documento debe de estar en formato PDF.
3. El nombre del archivo (comprimido como el documento PDF), será su LAB15_GRUPO_A/B/C_CUI_1erNOMBRE_1erAPELLIDO.

(Ejemplo: LAB15_GRUPO_A_2022123_PEDRO_VASQUEZ).

4. Debe remitir el documento ejecutable con el siguiente formato:

LAB15_GRUPO_A/B/C_CUI_EJECUTABLE_1erNOMBRE_1erAPELLIDO

(Ejemplo: LAB15_GRUPO_A_EJECUTABLE_2022123_PEDRO_VASQUEZ).

En caso de encontrarse trabajos similares, los alumnos involucrados no tendrán evaluación y serán sujetos a sanción.