UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA

FACULTAD DE PRODUCCIÓN Y SERVICIOS

ESCUELA DE CIENCIAS DE LA COMPUTACIÓN

PRÁCTICA DE LABORATORIO 15

CURSO DE CIENCIAS DE LA COMPUTACIÓN II

ESTUDIANTE:

RUIZ MAMANI, EDUARDO GERMÁN

EMAIL: eruizm@unsa.edu.pe

CUI: 20193061

TURNO:

C

AREQUIPA- PERÚ

2021

LINK DEL REPOSITORIO: https://github.com/EGRM23/CCII_20193061.git

1. EJERCICIO 1

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

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

class Carro {
private:
	string puertas;
	string llantas;
	string timon;
	string asientos;
	string motor;
	string espejos;
	string vidrios;
public:
	void setcolorpuertas(string c) {puertas = c;}
	void setcolorllantas(string c) {llantas = c;}
	void setcolortimon(string c) {timon = c;}
	void setcolorasientos(string c) {asientos = c;}
	void setcolormotor(string c) {motor = c;}
	void setcolorespejos(string c) {espejos = c;}
	void setcolorvidrios(string c) {vidrios = c;}

	void ListaPartes()const {
		cout << "Puertas: " << this->puertas;
		cout << "\nLlantas: " << this->llantas;
		cout << "\nTimon: " << this->timon;
		cout << "\nAsientos: " << this->asientos;
		cout << "\nMotor: " << this->motor;
		cout << "\nEspejos: " << this->espejos;
		cout << "\nVidrios: " << this->vidrios;
	}
};

class ICarroBuilder {
public:
	virtual ~ICarroBuilder() {}
	virtual void ProducirPuertas() const = 0;
	virtual void ProducirLlantas() const = 0;
	virtual void ProducirTimon() const = 0;
```

```
	virtual void ProducirAsientos() const = 0;
	virtual void ProducirMotor() const = 0;
	virtual void ProducirEspejos() const = 0;
	virtual void ProducirVidrios() const = 0;
};

class BuilderCarroEspecifico : public ICarroBuilder {
private:
	Carro* product;
public:
	BuilderCarroEspecifico() {
		this->Reset();
	}
	~BuilderCarroEspecifico() {
		delete product;
	}
	void Reset() {
		this->product = new Carro();
	}
	void ProducirPuertas()const override{
		int c;
		string colores[] =
{"Rojo","Azul","Amarillo","Negro","Blanco"};
		cout << "Puertas ¿Qué color desea?\n";
		for (int i = 0; i < 5; i++) {
			cout << i+1 << " " << colores[i] << endl;
		}
		cout << "Opcion? ";
		cin >> c;
		this->product->setcolorpuertas(colores[c-1]);
		cout << endl;
	}
	void ProducirLlantas()const override {
		int c;
		string colores[] = {"Negro","Blanco","Gris"};
		cout << "Llantas ¿Qué color desea?\n";
		for (int i = 0; i < 3; i++) {
			cout << i+1 << " " << colores[i] << endl;
		}
		cout << "Opcion? ";
		cin >> c;
		this->product->setcolorllantas(colores[c-1]);
		cout << endl;
	}
	void ProducirTimon()const override {
		int c;
		string colores[] = {"Negro","Blanco","Marrón"};
		cout << "Timon ¿Qué color desea?\n";
		for (int i = 0; i < 3; i++) {
			cout << i+1 << " " << colores[i] << endl;
		}
		cout << "Opcion? ";
		cin >> c;
		this->product->setcolortimon(colores[c-1]);
		cout << endl;
```

```
	}
	void ProducirAsientos()const override {
		int c;
		string colores[] = {"Negro","Blanco","Marrón"};
		cout << "Asientos ¿Qué color desea?\n";
		for (int i = 0; i < 3; i++) {
			cout << i+1 << " " << colores[i] << endl;
		}
		cout << "Opcion? ";
		cin >> c;
		this->product->setcolorasientos(colores[c-1]);
		cout << endl;
	}
	void ProducirMotor()const override {
		int c;
		string colores[] = {"Negro","Plateado","Blanco"};
		cout << "Motor ¿Qué color desea?\n";
		for (int i = 0; i < 3; i++) {
			cout << i+1 << " " << colores[i] << endl;
		}
		cout << "Opcion? ";
		cin >> c;
		this->product->setcolormotor(colores[c-1]);
		cout << endl;
	}
	void ProducirEspejos()const override {
		int c;
		string colores[] =
{"Rojo","Azul","Amarillo","Negro","Blanco"};
		cout << "Espejos ¿Qué color desea?\n";
		for (int i = 0; i < 5; i++) {
			cout << i+1 << " " << colores[i] << endl;
		}
		cout << "Opcion? ";
		cin >> c;
		this->product->setcolorespejos(colores[c-1]);
		cout << endl;
	}
	void ProducirVidrios()const override {
		int c;
		string colores[] =
{"Transparentes","Polarizado","Rosado","Celeste"};
		cout << "Vidrios ¿Qué color desea?\n";
		for (int i = 0; i < 4; i++) {
			cout << i+1 << " " << colores[i] << endl;
		}
		cout << "Opcion? ";
		cin >> c;
		this->product->setcolorvidrios(colores[c-1]);
		cout << endl;
	}
	Carro* getCarro() {
		Carro* resultado = this->product;
		this->Reset();
		return resultado;
```

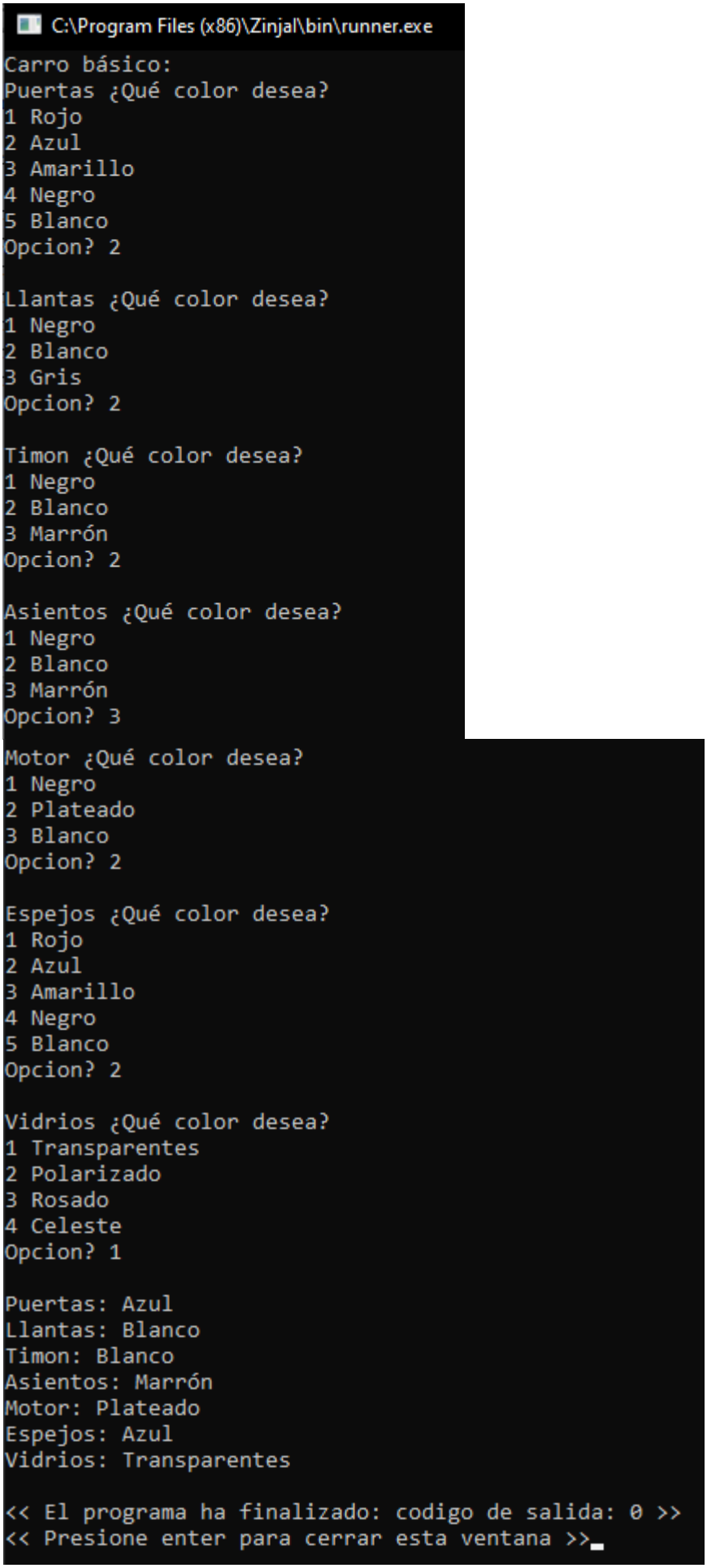
```
	}
};

class Director {
private:
	ICarroBuilder* builder;
public:
	void set_builder(ICarroBuilder* builder) {
		this->builder = builder;
	}
	void construct() {
		this->builder->ProducirPuertas();
		this->builder->ProducirLlantas();
		this->builder->ProducirTimon();
		this->builder->ProducirAsientos();
		this->builder->ProducirMotor();
		this->builder->ProducirEspejos();
		this->builder->ProducirVidrios();
	}
};

void ClienteCode(Director& director)
{
	BuilderCarroEspecifico* builder = new
BuilderCarroEspecifico();
	director.set_builder(builder);
	cout << "Carro básico:\n";
	director.construct();
	Carro* c = builder->getCarro();
	c->ListaPartes();
	delete c;
	delete builder;
}

int main() {
	Director* director = new Director();
	ClienteCode(*director);
	delete director;
	return 0;
}
```

- CAPTURAS


C:\Program Files (x86)\Zinjal\bin\runner.exe
Carro básico:
Puertas ¿Qué color desea?
1 Rojo
2 Azul
3 Amarillo
4 Negro
5 Blanco
Opcion? 2
Llantas ¿Qué color desea?
1 Negro
2 Blanco
3 Gris
Opcion? 2
Timon ¿Qué color desea?
1 Negro
2 Blanco
3 Marrón
Opcion? 2
Asientos ¿Qué color desea?
1 Negro
2 Blanco
3 Marrón
Opcion? 3
Motor ¿Qué color desea?
1 Negro
2 Plateado
3 Blanco
Opcion? 2
Espejos ¿Qué color desea?
1 Rojo
2 Azul
3 Amarillo
4 Negro
5 Blanco
Opcion? 2
Vidrios ¿Qué color desea?
1 Transparentes
2 Polarizado
3 Rosado
4 Celeste
Opcion? 1
Puertas: Azul
Llantas: Blanco
Timon: Blanco
Asientos: Marrón
Motor: Plateado
Espejos: Azul
Vidrios: Transparentes
<< El programa ha finalizado: codigo de salida: 0 >>
<< Presione enter para cerrar esta ventana >>