UNSA

UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA

“UNIVERSIDAD NACIONAL DE SAN AGUSTÍN”

FACULTAD DE INGENIERÍA,PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

CURSO:

Ciencias de la Computación - Grupo “B”

DOCENTE:

Enzo Edir Velásquez Lobatón

ALUMNO:

Fabricio Huaquisto Quispe

REPOSITORIO:

https://github.com/fhuaquisto21/EPCC-CCII

Arequipa - Perú

2022

1. Product.h

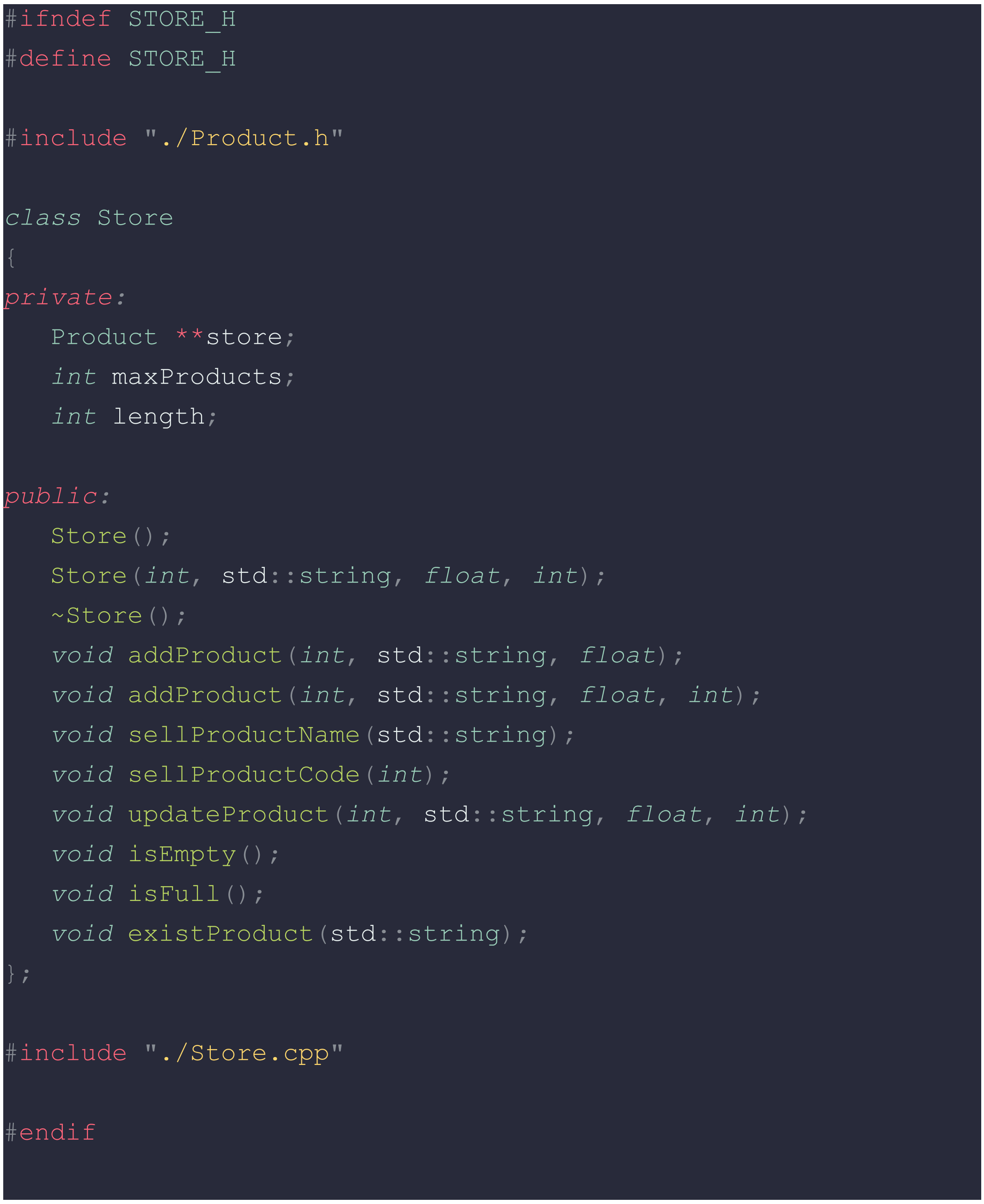
```cpp
#ifndef PRODUCT_H
#define PRODUCT_H

#include <iostream>

class Product
{
private:
   int code;
   std::string name;
   float price;
   int quantity;

public:
   Product();
   Product(int, std::string, float, int);
   ~Product();
   int getCode();
   std::string getName();
   float getPrice();
   int getQuantity();
   void setCode(int);
   void setName(std::string);
   void setPrice(float);
   void setQuantity(int);
};

#include "./Product.cpp"

#endif
```

2. Product.cpp

```
#include "./Product.h"

Product::Product() {}

Product::Product(int _code, std::string _name, float _price,
int _quantity)
{
   this->code = _code;
   this->name = _name;
   this->price = _price;
   this->quantity = _quantity;
}

Product::~Product() {}

int Product::getCode()
{
   return this->code;
}

std::string Product::getName()
{
   return this->name;
}

float Product::getPrice()
{
   return this->price;
}

int Product::getQuantity()
{
   return this->quantity;
}

void Product::setCode(int _code)
{
   this->code = _code;
}

void Product::setName(std::string _name)
```

```
{
    this->name = _name;
}

void Product::setPrice(float _price)
{
    this->price = _price;
}

void Product::setQuantity(int _quantity)
{
    this->quantity = _quantity;
}
```

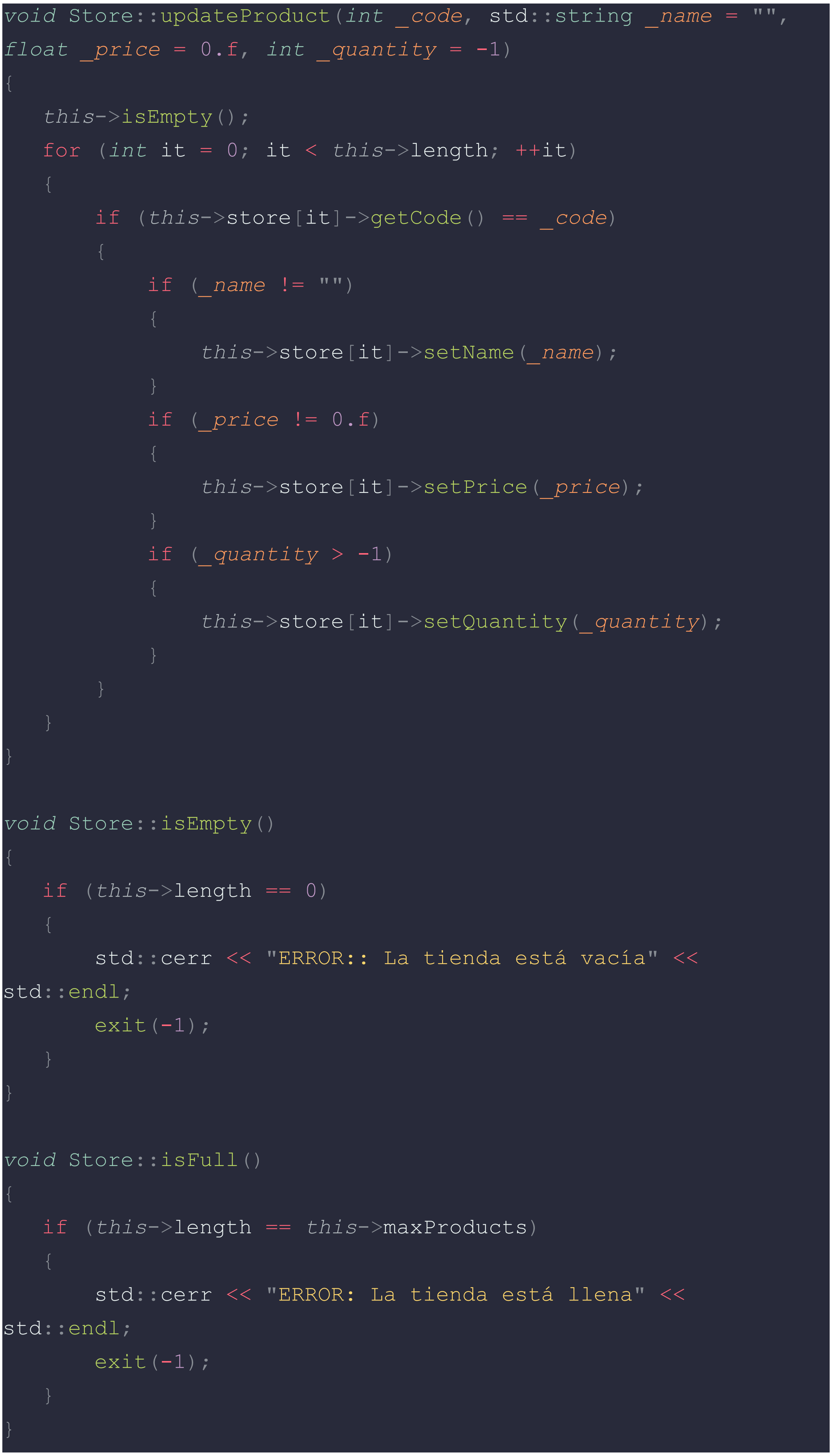
3. Store.h



```
#ifndef STORE_H
#define STORE_H

#include "./Product.h"

class Store
{
private:
   Product **store;
   int maxProducts;
   int length;

public:
   Store();
   Store(int, std::string, float, int);
   ~Store();
   void addProduct(int, std::string, float);
   void addProduct(int, std::string, float, int);
   void sellProductName(std::string);
   void sellProductCode(int);
   void updateProduct(int, std::string, float, int);
   void isEmpty();
   void isFull();
   void existProduct(std::string);
};

#include "./Store.cpp"

#endif
```

4. **Store.cpp**

```
#include "./Store.h"
Store::Store()
{
   this->maxProducts = 10;
   this->store = new Product *[this->maxProducts];
   this->length = 0;
}

Store::Store(int _code, std::string _name, float _price, int
_quantity = 0)
{
   this->maxProducts = 10;
   this->store = new Product *[this->maxProducts];
   Product *newProduct = new Product(_code, _name, _price,
_quantity);
   this->store[0] = newProduct;
   this->length = 1;
}

Store::~Store() {}

void Store::addProduct(int _code, std::string _name, float
_price, int _quantity = 0)
{
   this->isFull();
   this->existProduct(_name);
   Product *newProduct = new Product(_code, _name, _price,
_quantity);
   this->store[this->length] = newProduct;
   ++this->length;
}

void Store::sellProductCode(int _code)
{
   this->isEmpty();
   for (int it = 0; it < this->length; ++it)
   {
       if (this->store[it]->getCode() == _code)
       {
           if (this->store[it]->getQuantity() == 0)
           {
               std::cerr << "ERROR: El producto con código "
<< _code << " está agotado" << std::endl;
                    exit(-1);
                }

this->store[it]->setQuantity(this->store[it]->getQuantity() -
1);
            std::cout << "Comprado" << std::endl;
            return;
        }
    }
    std::cerr << "ERROR: El producto con código " << _code << "
no existe" << std::endl;
    exit(-1);
}

void Store::sellProductName(std::string _name)
{
    this->isEmpty();
    for (int it = 0; it < this->length; ++it)
    {
        if (this->store[it] != nullptr)
        {
            if (this->store[it]->getName() == _name)
            {
                if (this->store[it]->getQuantity() == 0)
                {
                    std::cerr << "ERROR: El producto " << _name
<< " está agotado" << std::endl;
                    exit(-1);
                }

this->store[it]->setQuantity(this->store[it]->getQuantity() -
1);
                std::cout << "Comprado" << std::endl;
                return;
            }
        }
    }
    std::cerr << "ERROR: El producto " << _name << " no existe"
<< std::endl;
    exit(-1);
}
```

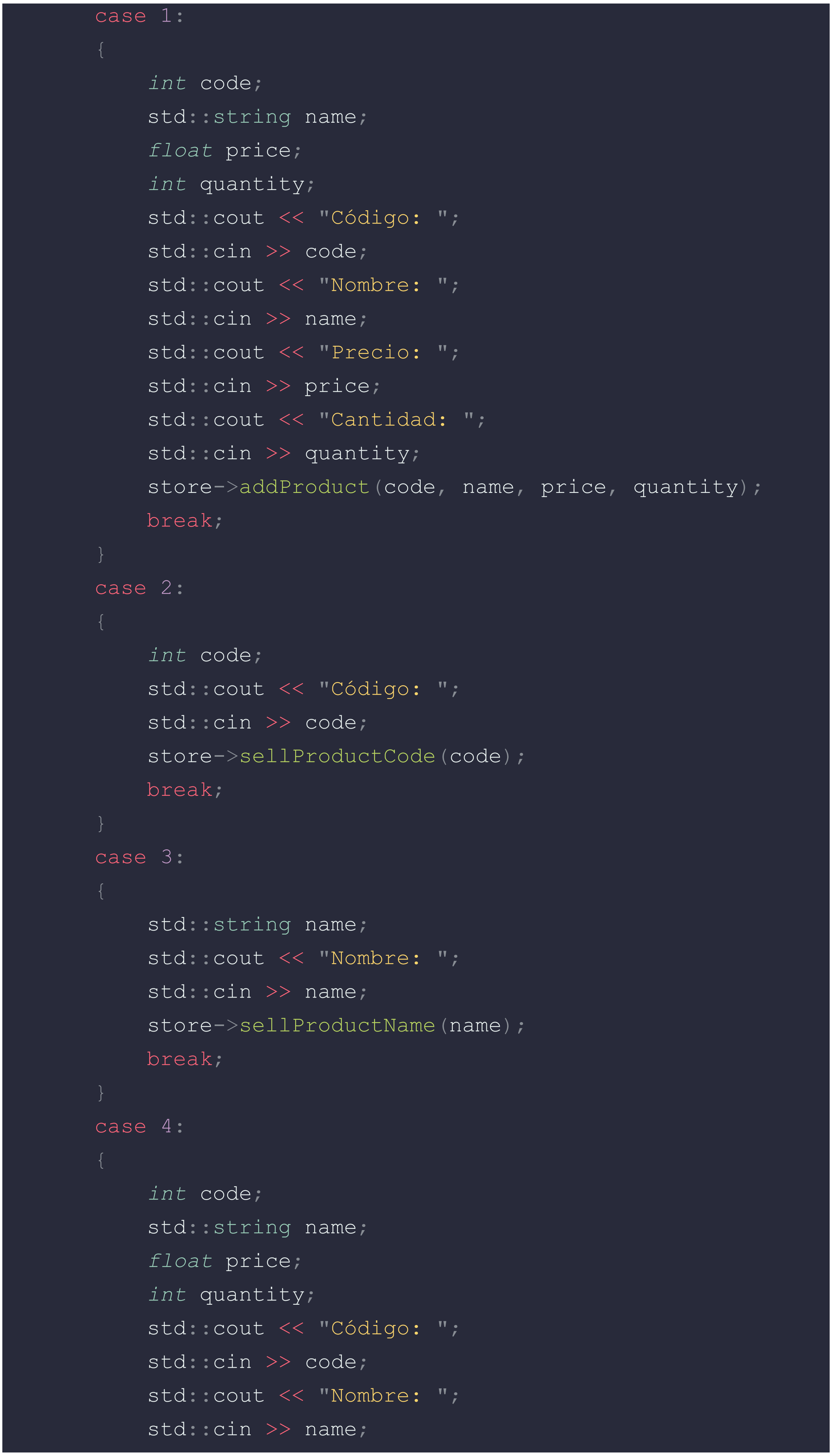


```
void Store::updateProduct(int _code, std::string _name = "",
float _price = 0.f, int _quantity = -1)
{
   this->isEmpty();
   for (int it = 0; it < this->length; ++it)
   {
       if (this->store[it]->getCode() == _code)
       {
           if (_name != "")
           {
               this->store[it]->setName(_name);
           }
           if (_price != 0.f)
           {
               this->store[it]->setPrice(_price);
           }
           if (_quantity > -1)
           {
               this->store[it]->setQuantity(_quantity);
           }
       }
   }
}

void Store::isEmpty()
{
   if (this->length == 0)
   {
       std::cerr << "ERROR:: La tienda está vacía" <<
std::endl;
       exit(-1);
   }
}

void Store::isFull()
{
   if (this->length == this->maxProducts)
   {
       std::cerr << "ERROR: La tienda está llena" <<
std::endl;
       exit(-1);
   }
}
```

```
void Store::existProduct(std::string _name)
{
    for (int it = 0; it < this->length; ++it)
    {
        if (this->store[it]->getName() == _name)
        {
             std::cerr << "ERROR: El producto ya existe en la
tienda" << std::endl;
             exit(-1);
        }
    }
}
```

5. **main.cpp**

```
#include <iostream>
#include "./Store.h"

void printMenu()
{
    std::cout << "[1] Añadir producto" << std::endl;
    std::cout << "[2] Comprar producto por código" <<
std::endl;
    std::cout << "[3] Comprar producto por nombre" <<
std::endl;
    std::cout << "[4] Actualizar producto por código" <<
std::endl;
    std::cout << "[0] Salir" << std::endl;
    std::cout << std::endl
              << "Option: ";
}

int main()
{
    Store *store = new Store();
    int opt = 0;
    do
    {
        printMenu();
        std::cin >> opt;
        printf("\e[1;1H\e[2J");
        switch (opt)
        {
```



```
        case 1:
        {
            int code;
            std::string name;
            float price;
            int quantity;
            std::cout << "Código: ";
            std::cin >> code;
            std::cout << "Nombre: ";
            std::cin >> name;
            std::cout << "Precio: ";
            std::cin >> price;
            std::cout << "Cantidad: ";
            std::cin >> quantity;
            store->addProduct(code, name, price, quantity);
            break;
        }
        case 2:
        {
            int code;
            std::cout << "Código: ";
            std::cin >> code;
            store->sellProductCode(code);
            break;
        }
        case 3:
        {
            std::string name;
            std::cout << "Nombre: ";
            std::cin >> name;
            store->sellProductName(name);
            break;
        }
        case 4:
        {
            int code;
            std::string name;
            float price;
            int quantity;
            std::cout << "Código: ";
            std::cin >> code;
            std::cout << "Nombre: ";
            std::cin >> name;
            std::cout << "Precio: ";
            std::cin >> price;
            std::cout << "Cantidad: ";
            std::cin >> quantity;
            store->updateProduct(code, name, price, quantity);
            break;
        }
        default:
        {
            break;
        }
        }
    } while (opt != 0);
    return 0;
}
```