**UNIVERSIDAD NACIONAL AGRARIA DE LA SELVA**

**FACULTAD DE INGENIERÍA EN INFORMÁTICA Y SISTEMAS**

**ESCUELA ACADÉMICA PROFESIONAL DE INGENIERÍA EN**

**INFORMÁTICA Y SISTEMAS**



**PROYECTO FINAL C++**

APLICACIÓN DE VENTA

**ASIGNATURA : Programación Básica**

**DOCENTE : Ing. Carlos A. Rio Rivera**

**INTEGRANTES : Gonzalo Coz Paz**

**Victor Adrian Orihuela Aspilcueta**

**Tingo María - Perú**

14 de Septiembre, 2023

Indice

[Introduccion 3](#_Toc135079146)

[Desarrollo 4](#_Toc135079147)

[Clase Padre (Persona) 4](#_Toc135079148)

[Clase Amiga (Producto) 6](#_Toc135079149)

[Clase Hija (Usuario) 7](#_Toc135079150)

[Principal 8](#_Toc135079151)

[Función: Interfaz\_Principal() 10](#_Toc135079152)

[Función: Opción\_venta() 10](#_Toc135079153)

[Función: Interfaz\_vender() 10](#_Toc135079154)

[Función: vender() 11](#_Toc135079155)

[Función: opcion\_inventario() 11](#_Toc135079156)

[Función: NewCodigo() 12](#_Toc135079157)

[Función: interfaz\_inventario() 12](#_Toc135079158)

[Función: prod\_iniciales() 12](#_Toc135079159)

[Función: Users\_iniciales() 12](#_Toc135079160)

[Conclusion 13](#_Toc135079161)

Introduccion

Este es un informe del proyecto creado en Visual Estudio Code para el curso de Programación Básica de la Universidad Nacional Agraria de la Selva,

En el presente informe se da a conocer las clases y sus funciones, así como los archivos y que contiene cada uno.

El informe se centra mucho en las funciones del archivo principal, ya que esta tiene toda la lógica del para conectar los archivos.

El programa creado es un sistema de ventas por usuario, pudiendo agregar productos si es necesario, crear una nueva cuenta para nuevos empleados.

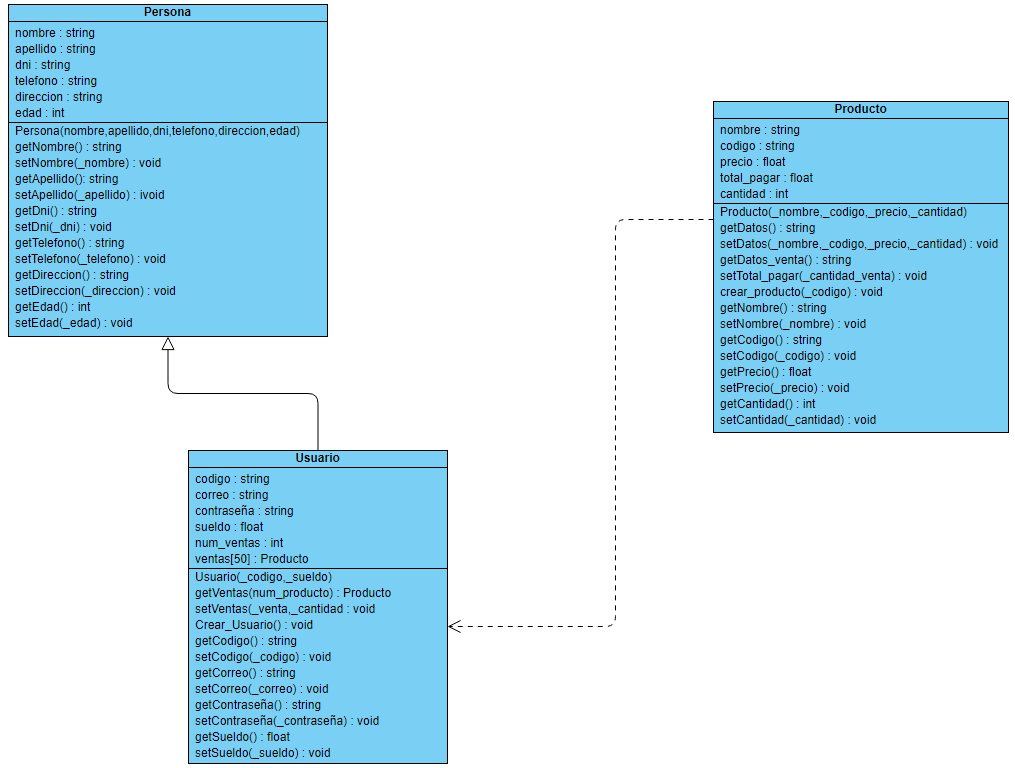
El programa puede mejorar pero la base es lo que se explica en este informe.

Desarrollo

Para el programa se necesita 4 archivos “.cpp”:

1 Clase Padre (Persona)  
1 Clase Hija (Usuario)  
1 Clase Amiga (Producto)  
1 Principal

Antes de todo se elaboró el diagrama de clases

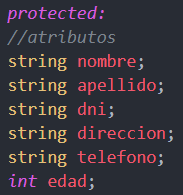


Clase Padre (Persona)

Las librería que usaremos seran las siguientes.  
a) iostream.  
b) string.

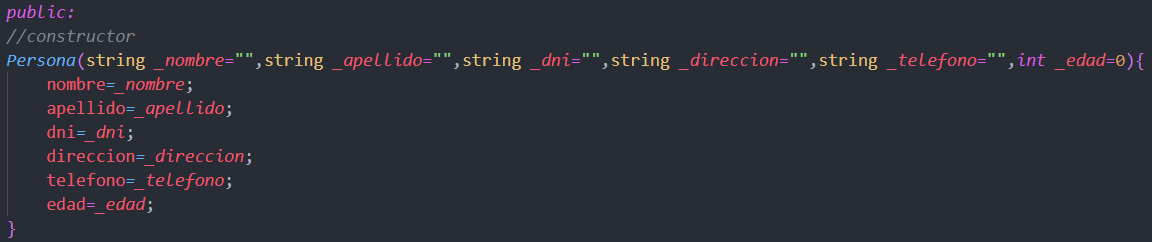
Los atibutos de esta clase son de tipo “protected” y tendra los siguientes atributos :nombre, apellido, dni, telefono, direccion, edad.

Todos los atributos de persona son de tipo cadena excepto la edad que es entero.



Cuando acabemos con los atributos seguimos con las funciones que seran de tipo “public”.

La primera fincion de tipo public sera el constructor de la clase.

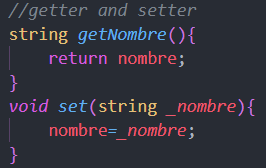


Para esta clase solo se usaran funciones basicas, estas seran los get y set de los atributos.

Para crea la funcion get primero va el tipo de la variable del atributo y esta funcion debe reornar el atributo respectivo.

Para la funcion set se emplea el void (no retorna nada) luego el nombre de la funcion y dentro de los parentesis de la funcion el tipo de la variable del atributo y luego el nombre del atributo con un “\_” al inicio.

Ejemplo:



Cuando acabemos de hacer los geter y setter habremos terminado esta clase.

Clase Amiga (Producto)

Las librería que usaremos seran las siguientes.  
a) iostream.  
b) string.

Los atibutos de esta clase son de tipo “private” y tendra los siguientes atributos :nombre, codigo, precio, total\_pagar, catidad.

Los atrivutos de tipo string son nombre y codigo, los de tipo float son precio y total\_pagar, por ende, el atributo cantidad es de tipo int.

Para esta clase sus funciones sigue siendo “public”, se crea los get y set de los atributos .

Esta clase tendra 5 funciones distintas.

a) getDatos() : esta funcion sera de tipo string y retornara todos los atributos concatenados como cadena.

b) setDatos(‘atributos’) : este es de tipo void se asemejara al constructor, la utilidad de tener esta funcion, es que no nesesitamos inicializar la clase para poder usarla y se puede inicializar despues, usando esta funcion.

c) getDatos\_Venta() : esta clase es de tipo string, y la diferencia con la funcion getDatos es que a este se le agrega el atributo total a pagar.

d) setTotal\_pagar(\_cantidad\_venta) : es de tipo void y lo que hace es darle el valor al atributo total\_pagar,multiplicando el precio por la cantidad de venta.

e) Crear\_producto(\_codigo) : es de tipo void y lo que hace esta funcion es pedir los datos del nuevo producto para luego colocarlo en los atributos respectivos, el atributo codigo se obtendra el archivo principal.

Clase Hija (Usuario)

Las librería que usaremos seran las siguientes.  
a) iostream  
b) string  
c) producto.cpp  
d) persona.cpp

Esta clase es especial, ya que esta es herencia de la clase Persona, y esto nos da el poder de usar los atributos de la clases persona sin nesesidad de llamar funciones.

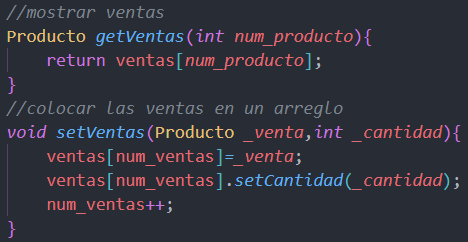
Los atibutos de esta clase son de tipo “private” y tendra los siguientes atributos :codigo, correo, contraseña, sueldo, num\_ventas,ventas[50].

Los atributos se dividen de la siguiente forma:  
a) string = codigo,correo,contraseña  
b) float =sueldo  
c) int = num\_ventas  
d) Producto = ventas[50]

El constructor de una clase que tiene herencia es distinta ya que al momento de colocar los atributos de la clase hija seguido se tendra que colocar los atributos de la clase padre, pero al momento de inicializar la variables en el contructor sigue siendo igual, es mas para esta clase su constructor tiene inicializado el correo como la concatenacion del nombre mas el punto y el arroba es predefinido, su contraseña predefinido es igual al apellido.

Se crean los get y set de los atributos simples ya que el atributo ventas es otra clase, en este caso, clase amiga y es un arreglo.

Los get y set de venta, van a tener el numero de producto de tipo entero en el caso del get, para el set tendremos que pedir el producto de la venta de tipo producto y la cantidad que se vendio.



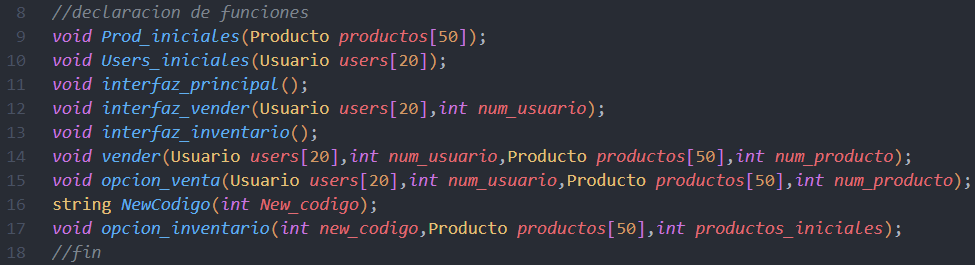
Para esta clase solo hay una función diferente de los get y set y este es:  
Crear\_Usuario() : es de tipo void, lo que hace la función es que pedir los datos del usuario y colocarlos en su respectivo atributo, antes de finalizar la función se le proporciona el correo y la contraseña que se crea con el nombre y apellido.

Principal

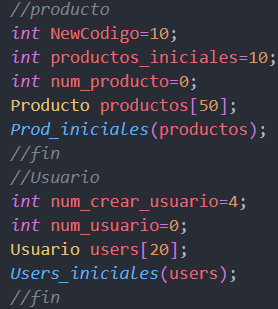
En el archivo principal con nombre “main.cpp” se le incluirá el archivo “usuario.cpp”, incluyendo este archivo también estaremos incluyendo las otras 2 clases.

Las librería que usaremos seran las siguientes.  
a) iostream  
b) string  
c) usuario.cpp

Para la comodidad del proyecto las declaraciones de las funciones usadas en el archivo principal se escriben después de las librerías.



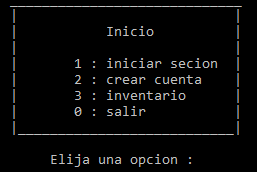
En la función “int main ()” primero se declara e inicializa los usuarios y los productos que habrá en un inicio.



Luego creamos un bucle con un switch dentro del el, antes del switch está la función de interfaz principal.

Función: Interfaz\_Principal()

Esta función lo que hace es, mostrar un cuadrado y dentro hay instrucciones para el usuario.



Para la opción 1 se utiliza la función de “opción\_venta”.

Para la opción 2 se llama a la función de la clase usuario “crear\_cuenta()”.

Para la opción 3 se utiliza la función “opcion\_inventario()”.

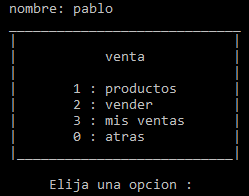
Función: Opción\_venta()

Esta función lo que hace es, primero simular un inicio de sesión, pidiendo que ingresemos el correo, y luego le pedimos que ingrese su contraseña, solo se podrá ingresar si la contraseña que escribió le pertenece al correo que escribió.

Cuando la contraseña y el correo coincidan, se obtiene también el número de arreglo del usuario, esto es importante ya que es el usuario que vamos a usar cuando estamos en esta función, se llamara a la función “Interfaz \_vender”

Función: Interfaz\_vender()

En esta función se muestra un cuadro y dentro hay instrucciones para el usuario.



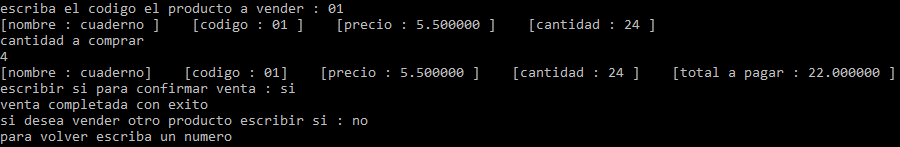
Cuando se elige la primera opción se hace un bucle para imprimir todos los productos que tenga datos relevantes.

La opción 3 imprime las ventas del usuario.

Para la opción vender se llama a la función “vender()”.

Función: vender()

En esta función se nos pedirá el código del producto que se va vender, cuando ingreses el código se te imprimirá los datos del producto y te pedirá la cantidad a vender, después te dará el monto a pagar, para después confirmar la venta del producto, luego te preguntar si quieres vender otro producto.



Función: opcion\_inventario()

Primero se llamará a la función “interfaz\_inventario()”

La primera opción es para agregar producto al arreglo de productos, para esto se usa la función “crear\_producto()” que es función de la clase producto y este necesita el parámetro código que se crea en una función llamada “NewCodigo()”.

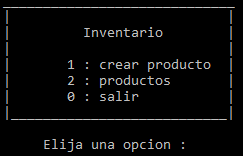
La segunda imprime los productos con valor creados.

Función: NewCodigo()

Esta función tiene una variable de tipo entera que se va sumando cada vez que se crea un producto, pero la función retorna un valor de tipo string, así que el número se convierte a string luego se suma 1 a la variable, pero retorna la variable tipo string.

Función: interfaz\_inventario()

Esta función se escribe un cuadrado y dentro hay instrucciones para el usuario.



Función: prod\_iniciales()

En esta función se inicializan algunos productos.

Función: Users\_iniciales()

En esta función se inicializan algunos usuarios.

Conclusion

En conclusion para trrabajr con clases en distintos archivos es imortante crear el diagrama de clases para entender mejor lo que vamos a crear y como funciona una clase en especifico.

Usar las funciones con mas fracuencia para maximizar el rendimiento de estas.