

**INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE
MONTERREY**

Programación orientada a objetos

Profesores:

Mónica Larre Bolaños Cacho

**Proyecto Integrador “Modelado de un sistema de administración del consumo de
alimentos”**

Integrantes:

Erika Nohemí García Martínez	A01425330
Axel Daniel Godoy Barrios	A01425316

Fecha de entrega: 15 de junio del 2023

Índice:

Introducción (planteamiento del problema).....	3
Diagrama de clases UML con una argumentación del por qué del diseño.....	4
Ejemplo de ejecución (capturas de pantalla).....	5
Argumentación de las partes del proyecto relacionadas con cada uno de los puntos mencionados arriba.....	9
Identificación de casos que harían que el proyecto deje de funcionar.....	10
Conclusión.....	10
Referencias consultadas.....	11

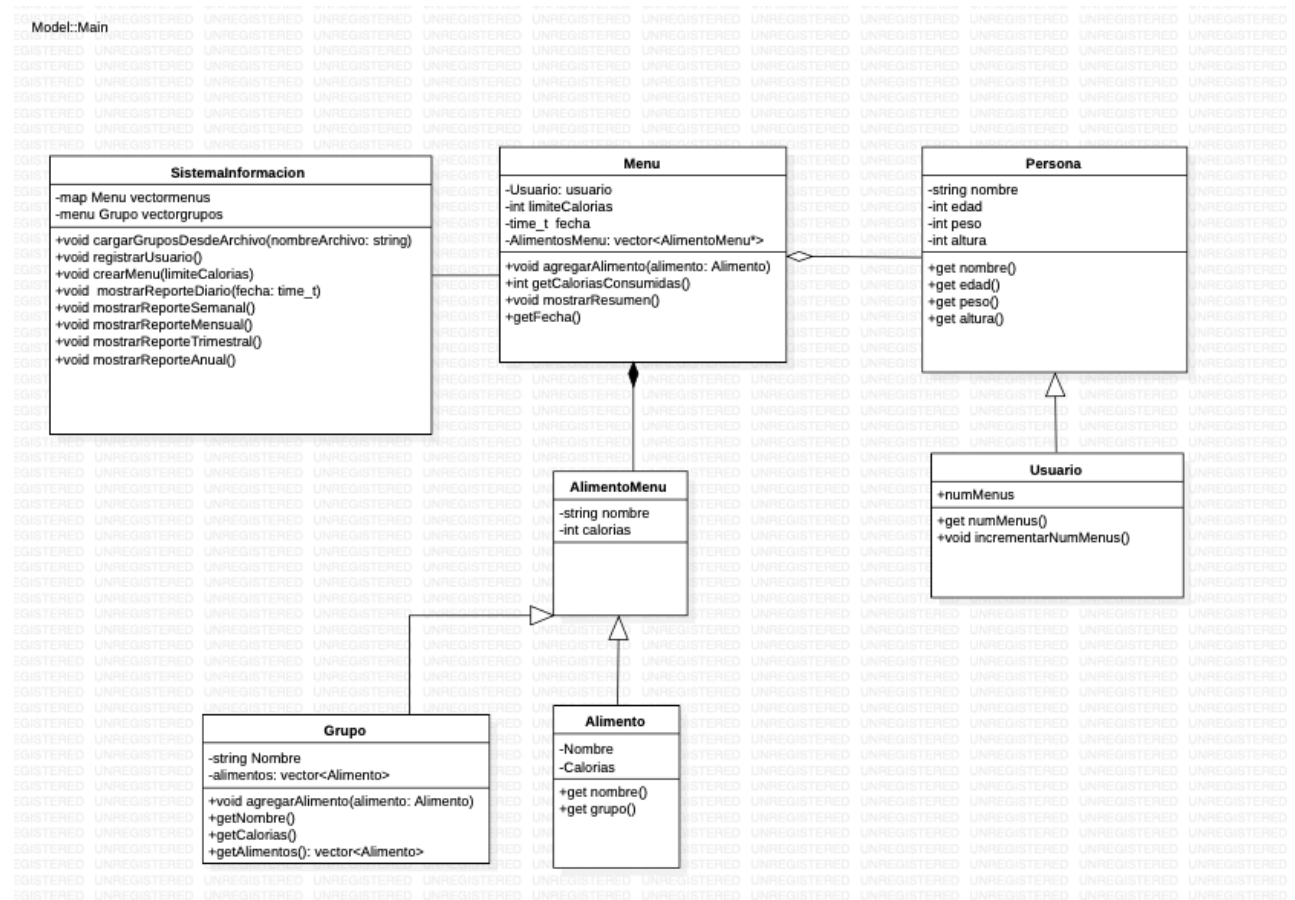
Introducción

En los últimos años, ha crecido la conciencia por una alimentación sana, por lo que se necesita el diseño e implementación de un sistema de información que permita administrar el consumo diario, semanal y mensual de alimentos.

Se quiere trabajar con catálogos de alimentos clasificados por grupos, por ejemplo Grupo 1: leche, queso, yogur, ..., Grupo 3: leguminosas, frutos secos, ... Investiga y decide la mejor clasificación. Se Necesita la aplicación de herencia en este caso. Decidan las clases que requieren y diseñarlas con los atributos y métodos necesarios para lograr construir el sistema solicitado. El sistema deberá ayudar a construir menús a partir de los catálogos de alimentos con cantidades, por ejemplo 100 gramos de pollo y 100 gramos de leguminosas. El menú creado deberá mostrar la cantidad de calorías del mismo, por ejemplo 800 calorías. El programa podrá generar hasta cuatro menús diarios y colocar las restricciones necesarias para que funciones como una herramienta útil. El sistema deberá contar con la funcionalidad de personalizar la aplicación de tal forma que el usuario pueda tomar algunas decisiones. Una vez que un menú es creado, el sistema deberá almacenarlo con todos sus datos junto con la fecha y hora en que fue creado.

El sistema deberá generar reportes diarios, semanales y mensuales mostrando diferentes cantidades de información a petición del usuario. Por ejemplo, cantidad semanal de calorías sin detalle de los menús, o bien, la descripción detallada de los menús de un mes, o tal vez, los catálogos de alimentos registrados.

Diagrama de clases UML con una argumentación del por qué del diseño



El diseño del código está compuesto por las siguientes funciones, en primer lugar esta abstracción y herencia, el código utiliza la abstracción de la clase abstracta "AlimentoMenu" para representar un concepto genérico de alimento en el menú. La clase "Alimento" hereda de "AlimentoMenu" y representa un tipo específico de alimento. Lo que nos permite tratar a los alimentos de manera polimórfica y agregar nuevos tipos de alimentos en el futuro. En segundo lugar utilizamos composición en varias clases, como la clase "Menu", que tiene una relación de "tiene un" con objetos de las clases "Usuario" y "AlimentoMenu". Esta composición permite que un menú está compuesto por un usuario y una lista de alimentos seleccionados y la clase `Grupo` también utiliza composición para almacenar una lista de alimentos.

También utiliza contenedores STL vector y map para almacenar y administrar objetos de las clases para una implementación eficiente en alimentos, menús y otros datos. Por último implementamos el manejo de excepciones para capturar errores al cargar el archivo de grupos alimenticios y al convertir las calorías en enteros.

Ejemplo de ejecución

“Caso prueba creación de 1 menú en 1 día con un usuario “

Registro del usuario

El programa comienza solicitando al usuario que ingrese el nombre de un archivo que contiene los grupos alimenticios y sus alimentos correspondientes. Luego, se le pide al usuario que se registre ingresando su nombre, edad, peso y altura. Después del registro exitoso, se muestra un menú de opciones para que el usuario interactúe con el programa.

```
Ingrese el nombre del archivo de grupos alimenticios: Comidas.txt

----- Menú Principal -----
1. Registrar Usuario
2. Salir
Ingrese la opción: 1
----- Registro de Usuario -----
Ingrese su nombre: Axel
Ingrese su edad: 19
Ingrese su peso en kg: 65
Ingrese su altura en cm: 189
----- Usuario registrado exitosamente -----

----- Menú de Opciones -----
1. Crear Menú
2. Mostrar Reporte Diario
3. Mostrar Reporte Semanal
4. Mostrar Reporte Mensual
5. Mostrar Reporte Trimestral
6. Mostrar Reporte Anual
7. Salir
Ingrese la opción: 1
```

Proceso para crear menú

Si el usuario elige la opción "Crear Menú", se le solicita ingresar el límite de calorías para el menú y procede a agregar alimentos al menú. Se muestra una lista de grupos alimenticios disponibles y se le pide al usuario que seleccione un grupo. Una vez que se selecciona un grupo, se muestra una lista de alimentos disponibles dentro del grupo y se le pide al usuario que seleccione un alimento para agregarlo al menú. Después de agregar un alimento al menú, se le pregunta al usuario si desea agregar otro alimento. Si elige "Sí", se repite el proceso de selección de grupo y alimento. Si elige "No", se muestra un mensaje indicando que el menú se ha creado con éxito y se asigna un ID al menú.

```
----- Menú de Opciones -----
1. Crear Menú
2. Mostrar Reporte Diario
3. Mostrar Reporte Semanal
4. Mostrar Reporte Mensual
5. Mostrar Reporte Trimestral
6. Mostrar Reporte Anual
7. Salir
Ingrese la opción: 1

Ingrese el límite de calorías para el menú: 450

----- Agregar Alimento al Menú -----
Seleccione el grupo alimenticio:
1. : Lácteos
2. : Carnes y pescados
3. : Legumbres y frutos secos
4. : Frutas y verduras
5. : Cereales y panes

Ingrese el número del grupo: 2

Seleccione el alimento del grupo:
1. Pollo (200 kcal)
2. Carne de res (300 kcal)
3. Pescado (180 kcal)

Ingrese el número del alimento: 1
```

```
Seleccione el alimento del grupo:
1. Pollo (200 kcal)
2. Carne de res (300 kcal)
3. Pescado (180 kcal)

Ingrese el número del alimento: 1

¿Desea agregar otro alimento al menú? (1: Sí / 0: No): 1

----- Agregar Alimento al Menú -----
Seleccione el grupo alimenticio:
1. : Lácteos
2. : Carnes y pescados
3. : Legumbres y frutos secos
4. : Frutas y verduras
5. : Cereales y panes

Ingrese el número del grupo: 4

Seleccione el alimento del grupo:
1. Manzana (60 kcal)
2. Plátano (90 kcal)
3. Zanahoria (40 kcal)

Ingrese el número del alimento: 1

¿Desea agregar otro alimento al menú? (1: Sí / 0: No): 0

Menú creado exitosamente con ID: 1
```

Caso donde se agregan 4 menús

```
----- Agregar Alimento al Menú -----
Seleccione el grupo alimenticio:
1. : Lácteos
2. : Carnes y pescados
3. : Legumbres y frutos secos
4. : Frutas y verduras
5. : Cereales y panes

Ingrese el número del grupo: 1

Seleccione el alimento del grupo:
1. Leche (150 kcal)
2. Queso (250 kcal)
3. Yogur (120 kcal)

Ingrese el número del alimento: 1

¿Desea agregar otro alimento al menú? (1: Sí / 0: No): 0
Ha alcanzado el límite diario de menús. No puede crear más menús hoy.
```

Creación del reporte

1. Si el usuario elige la opción "Mostrar Reporte Diario", se muestra un reporte detallado de los menús creados en el día actual. Se muestra el nombre del usuario que creó el menú, la fecha de creación, la lista de alimentos en el menú y las calorías totales consumidas. También se muestra un resumen de calorías totales y si se ha alimentado bien o ha excedido el límite calórico.
2. Si el usuario elige la opción "Mostrar Reporte Semanal", se muestra un reporte resumido de los menús creados durante la semana. Se muestra el total de calorías consumidas y si se ha alimentado bien o ha excedido el límite calórico.
3. Si el usuario elige la opción "Mostrar Reporte Mensual", se muestra un reporte resumido de los menús creados durante el mes. Se muestra el total de calorías consumidas y si se ha alimentado bien o ha excedido el límite calórico.
4. Si el usuario elige la opción "Mostrar Reporte Trimestral", se muestra un reporte resumido de los menús creados durante el trimestre. Se muestra el total de calorías consumidas y si se ha alimentado bien o ha excedido el límite calórico.
5. Si el usuario elige la opción "Mostrar Reporte Anual", se muestra un reporte resumido de los menús creados durante el año. Se muestra el total de calorías consumidas y si se ha alimentado bien o ha excedido el límite calórico.

```
----- Menú de Opciones -----
1. Crear Menú
2. Mostrar Reporte Diario
3. Mostrar Reporte Semanal
4. Mostrar Reporte Mensual
5. Mostrar Reporte Trimestral
6. Mostrar Reporte Anual
7. Salir
Ingrese la opción: 2

----- Reporte Diario - 14/6/2023 -----

Menú creado por: Axel
Fecha de creación: Wed Jun 14 16:30:37 2023
Alimentos en el menú:
- Pollo (200 kcal)
- Manzana (60 kcal)
Calorías totales: 260 kcal
Se ha alimentado bien.
Total de calorías consumidas: 260 kcal
Se ha alimentado bien.

----- Menú de Opciones -----
1. Crear Menú
2. Mostrar Reporte Diario
3. Mostrar Reporte Semanal
4. Mostrar Reporte Mensual
5. Mostrar Reporte Trimestral
6. Mostrar Reporte Anual
7. Salir
Ingrese la opción: 3

----- Reporte Semanal -----
Total de calorías consumidas: 260 kcal
Se ha alimentado bien.
```

Resumen diario de 4 menús, en la parte donde indica que el límite calórico ha excedido se refiere a que sobrepasó las calorías establecidas del menú.

```
----- Reporte Diario - 14/6/2023 -----  
  
Menú creado por: Axel  
Fecha de creación: Wed Jun 14 16:30:37 2023  
Alimentos en el menú:  
- Pollo (200 kcal)  
- Manzana (60 kcal)  
Calorías totales: 260 kcal  
Se ha alimentado bien.  
  
Menú creado por: Axel  
Fecha de creación: Wed Jun 14 16:35:45 2023  
Alimentos en el menú:  
- Queso (250 kcal)  
Calorías totales: 250 kcal  
Se ha pasado del límite calórico.  
  
Menú creado por: Axel  
Fecha de creación: Wed Jun 14 16:35:53 2023  
Alimentos en el menú:  
- Pasta (180 kcal)  
Calorías totales: 180 kcal  
Se ha alimentado bien.  
  
Menú creado por: Axel  
Fecha de creación: Wed Jun 14 16:36:09 2023  
Alimentos en el menú:  
- Pescado (180 kcal)  
Calorías totales: 180 kcal  
Se ha alimentado bien.  
Total de calorías consumidas: 870 kcal  
Se ha alimentado bien.
```

Reporte semanal, mensual y trimestral

```
----- Menú de Opciones -----  
1. Crear Menú  
2. Mostrar Reporte Diario  
3. Mostrar Reporte Semanal  
4. Mostrar Reporte Mensual  
5. Mostrar Reporte Trimestral  
6. Mostrar Reporte Anual  
7. Salir  
Ingrese la opción: 3  
  
----- Reporte Semanal -----  
Total de calorías consumidas: 870 kcal  
Se ha alimentado bien.  
  
----- Menú de Opciones -----  
1. Crear Menú  
2. Mostrar Reporte Diario  
3. Mostrar Reporte Semanal  
4. Mostrar Reporte Mensual  
5. Mostrar Reporte Trimestral  
6. Mostrar Reporte Anual  
7. Salir  
Ingrese la opción: 4  
  
----- Reporte Mensual -----  
Total de calorías consumidas: 870 kcal  
Se ha alimentado bien.  
  
----- Menú de Opciones -----  
1. Crear Menú  
2. Mostrar Reporte Diario  
3. Mostrar Reporte Semanal  
4. Mostrar Reporte Mensual  
5. Mostrar Reporte Trimestral  
6. Mostrar Reporte Anual  
7. Salir  
Ingrese la opción: 5  
  
----- Reporte Trimestral -----  
Total de calorías consumidas: 870 kcal  
Se ha alimentado bien.
```


Reporte anual y salida del programa.

Si el usuario elige la opción "Salir", el programa finaliza.

```
----- Reporte Anual -----  
Total de calorías consumidas: 870 kcal  
Se ha alimentado bien.  
  
----- Menú de Opciones -----  
1. Crear Menú  
2. Mostrar Reporte Diario  
3. Mostrar Reporte Semanal  
4. Mostrar Reporte Mensual  
5. Mostrar Reporte Trimestral  
6. Mostrar Reporte Anual  
7. Salir  
Ingrese la opción: 7  
PS C:\Users\Danie\OneDrive\Documentos\Tec\P00> |
```

Argumentación de las partes del proyecto relacionadas con cada uno de los puntos mencionados arriba

Polimorfismo: El polimorfismo se utiliza en el programa a través del uso de funciones virtuales puras en la clase base `AlimentoMenu`. La clase `Alimento` y la clase `Grupo` son clases derivadas de `AlimentoMenu` y deben implementar las funciones virtuales puras `getNombre()` y `getCalorias()`. Esto permite que se llamen a los métodos `getNombre()` y `getCalorias()` de manera polimórfica, lo que significa que se pueden utilizar para objetos de las clases derivadas a través de un puntero o referencia a `AlimentoMenu`. Esto proporciona una forma genérica de obtener el nombre y las calorías de un alimento, independientemente de si es un `Alimento` o un `Grupo`.

Herencia: La herencia se utiliza en varias partes del programa. En la clase `Alimento` hereda de la clase `AlimentoMenu` y la clase `Grupo` hereda de la clase `AlimentoMenu`. Esta herencia permite que `Alimento` y `Grupo` compartan la funcionalidad común definida en `AlimentoMenu`, como los métodos `getNombre()` y `getCalorias()`. Al heredar de la clase base, las clases derivadas también pueden agregar atributos y métodos adicionales específicos de ellas mismas.

Sobrecarga de operadores: En el código, se sobrecarga el operador `==` en la clase `Alimento` mediante la definición del método `operator==`. Esta sobrecarga permite comparar dos objetos `Alimento` utilizando el operador `==`. En la implementación, se verifica si los nombres y las calorías de los alimentos son iguales para determinar si son iguales. Esto permite realizar comparaciones de igualdad entre objetos `Alimento` de manera más conveniente y legible.

Excepciones: Se utiliza la gestión de excepciones para manejar posibles errores al leer el archivo de grupos alimenticios. En el método `cargarGruposDesdeArchivo`, se captura una excepción `std::invalid_argument` cuando las calorías de un alimento no son un número válido. Si se produce esta excepción, se muestra un mensaje de

error indicando el alimento problemático y se continúa con la siguiente línea del archivo. Esta gestión de excepciones ayuda a manejar los errores de manera controlada y evitar que el programa se detenga abruptamente en caso de errores en los datos de entrada.

Identificación de casos que harían que el proyecto deje de funcionar

Formato incorrecto del archivo de grupos alimenticios: Si el archivo de grupos alimenticios no sigue el formato esperado, es posible que el programa no pueda cargar los grupos y sus alimentos correctamente.

Entrada de datos inválida o inesperada: Si el usuario ingresa datos inválidos o inesperados durante la interacción con el programa, como opciones numéricas fuera de rango, caracteres no válidos, o valores no numéricos donde se esperan números, esto podría llevar a errores o comportamientos inesperados.

Límites o restricciones del programa: El código actual tiene algunas restricciones y límites establecidos, como un límite de 4 menús diarios por usuario. Si se excede alguno de estos límites, el programa mostrará mensajes de advertencia o no permitirá realizar ciertas acciones.

Dependencias externas: El código depende de la disponibilidad y correcta funcionalidad de las bibliotecas y funciones externas utilizadas, como la biblioteca de entrada y salida estándar de C++ (iostream).. etc.

Conclusión

Al crear este código, se han abordado varios temas importantes en la programación orientada a objetos y el desarrollo de aplicaciones. Primero “Abstracción y encapsulación”, la abstracción permite representar conceptos del dominio de manera clara y concisa, mientras que la encapsulación oculta los detalles internos y proporciona una interfaz para interactuar con los objetos. Estos principios son fundamentales para construir una estructura de clases. En segundo lugar “Herencia y polimorfismo”, la herencia permite la reutilización de código y la creación de jerarquías de clases. El polimorfismo permite tratar los objetos derivados como objetos de la clase base, lo que proporciona flexibilidad y permite el uso de una interfaz común para diferentes tipos de objetos. Por otro lado “Sobrecarga de operadores”, la sobrecarga de operadores permite definir comportamientos personalizados para operadores en clases definidas por el usuario. En el caso de “Excepciones”, el manejo de excepciones es fundamental para controlar y manejar errores en tiempo de ejecución de manera controlada. Las excepciones permiten capturar y manejar situaciones excepcionales, evitando que el programa se detenga abruptamente y proporcionando la capacidad de recuperación.

La creación de este código ha permitido aplicar y comprender conceptos fundamentales de la programación orientada a objetos, como la abstracción, la encapsulación, la herencia, el polimorfismo, la sobrecarga de operadores y el manejo de excepciones. Además, hemos aprendido sobre el uso de bibliotecas y contenedores de la biblioteca estándar de C++, así como la importancia de la validación de entrada y el manejo de errores. Estos conocimientos sientan una base sólida para desarrollar aplicaciones más complejas y comprender mejor los principios de diseño y las mejores prácticas de programación.

Referencias consultadas

C++ polymorphism. (2017, mayo 22). GeeksforGeeks.
<https://www.geeksforgeeks.org/cpp-polymorphism/>

Exception handling in C++. (2013, diciembre 23). GeeksforGeeks.
<https://www.geeksforgeeks.org/exception-handling-c/>

Inheritance in C++. (2017, mayo 25). GeeksforGeeks.
<https://www.geeksforgeeks.org/inheritance-in-c/>