



TC1030

Proyecto integrador

*Modelado de un sistema de administración del
consumo de alimentos*

Dra. Mónica Larre Bolaños Cacho

Dulce Nahomi Bucio Rivas A01425284 || Desirée

Espinosa Contreras A01425162

Índice

Introducción.....	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.....	7
Identificación de casos que harían que el proyecto deje de funcionar.....	8
Conclusión.....	9
Referencias consultadas.....	9

Introducción

La alimentación saludable se ha convertido en un pilar fundamental en la vida cotidiana de muchas personas. Esta tendencia ha dado lugar a una creciente demanda de herramientas y aplicaciones que puedan facilitar la gestión y planificación de las dietas. Sin embargo, la variedad y la falta de personalización de las soluciones existentes dejan un espacio significativo para la innovación. Además, muchas de estas herramientas no utilizan todo el potencial de la tecnología, como la programación orientada a objetos (POO), para proporcionar un sistema robusto, eficiente y adaptable a las necesidades cambiantes de los usuarios.

Se ha identificado la necesidad de un sistema de información integral que permita a los usuarios administrar su consumo de alimentos de manera eficiente y personalizada. Este sistema debe incluir catálogos de alimentos clasificados en grupos, permitir la construcción de menús basados en estos catálogos, y también proporcionar información nutricional relevante, como la cantidad de calorías en cada menú. Además, la herramienta deberá adaptarse a las necesidades específicas de cada usuario, proporcionando recomendaciones de consumo calórico basadas en factores como la edad y el peso.

Para lograr este objetivo, se propone utilizar los conceptos de la programación orientada a objetos, como la herencia, el polimorfismo y la sobrecarga de operadores, para crear un conjunto de clases que faciliten la solución de este problema. Estas técnicas proporcionan una manera de diseñar y organizar el código de una manera intuitiva y flexible, lo que facilita la adaptación a nuevas características y la solución de problemas.

Este trabajo abordará el diseño e implementación de dicho sistema, identificando las clases requeridas, sus atributos y métodos, y cómo estos elementos se interrelacionan para cumplir con los requisitos de la herramienta. Se espera que este enfoque proporcione una solución flexible y personalizada que pueda ayudar a las personas a administrar su dieta de manera efectiva y promover hábitos alimenticios saludables.

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



Iniciamos con la clase 'Alimento', que define cualquier tipo de alimentos con dos atributos esenciales: 'nombre' y 'calorías por 100 gramos'. A continuación, la clase 'Grupo' permite agrupar alimentos en vectores, facilitando su clasificación por categoría.

Estos conjuntos de alimentos son posteriormente estructurados en la clase 'Menú', la cual también almacena la fecha y hora de creación, y el conteo calórico total del menú generado. Se implementarán métodos que permitan crear diferentes comidas del día a partir de estos grupos de alimentos, junto con los correspondientes métodos getter.

La clase 'Persona' alberga los datos personales del usuario, que serán la base para nuestro sistema de recomendación de dietas, centrado en el índice de masa corporal. Estos datos y métodos serán heredados por la clase 'Usuario', aplicando polimorfismo dado que una persona puede ser también un usuario. En 'Usuario', almacenaremos todos los datos relevantes para las funcionalidades y solicitudes del usuario: un menú personalizado y un vector de menús para su posterior consulta, utilizando los métodos desarrollados en 'Menú' para segmentar las comidas del día.

Finalmente, la clase 'MyApp' gestiona todas las operaciones de la aplicación, como la creación de usuarios, la generación de menús para estos, y la elaboración de informes. Esta clase es necesaria para almacenar la información de los usuarios en un vector de datos, y no se utiliza 'main' para evitar la pérdida de información tras cada ejecución.

Ejemplo de ejecución (capturas de pantalla)

El registro es imprescindible para acceder a las diversas funciones que ofrece la aplicación.

```
└─¡Bienvenido a Vida Sana!
1. Iniciar sesion
2. Registrarse
3. Salir
2
Introduzca un identificador de usuario
dul
Usuario creado con exito
└─¡Bienvenido a Vida Sana!
1. Iniciar sesion
2. Registrarse
3. Salir
1
Ingrese su nombre de usuario:
dul
El usuario no existe
└─¡Bienvenido a Vida Sana!
1. Iniciar sesion
2. Registrarse
3. Salir
```

Tras iniciar sesión correctamente, la aplicación ofrece las siguientes funcionalidades:

```

¡Bienvenido a Vida Sana!
1. Iniciar sesión
2. Registrarse
3. Salir
1
Ingrese su nombre de usuario:
Dul
1. Registrar datos
2. Recomendar menú
3. Generar reporte completo
4. Generar reporte mensual
5. Generar reporte diario
6. Generar reporte anual
7. Generar reporte calórico
8. Salir

```

Para poder generar un menú, es esencial registrar los datos del usuario. Una vez se haya desplegado el menú, el usuario tiene la opción de decidir si requiere crear otro menú o no.

<pre> 2 No hay datos registrados para poder crear tu menú. 1. Registrar datos 2. Recomendar menú 3. Generar reporte completo 4. Generar reporte mensual 5. Generar reporte diario 6. Generar reporte anual 7. Generar reporte calórico 8. Salir 1 Ingrese su nombre: Dulce Ingrese su peso: 50 Ingrese su altura: 1.50 Ingrese su edad: 18 1. Registrar datos 2. Recomendar menú 3. Generar reporte completo 4. Generar reporte mensual 5. Generar reporte diario 6. Generar reporte anual 7. Generar reporte calórico 8. Salir 2 </pre>	<pre> Te hemos construido este menú: Tu IMC es menor a 18.5, se te recomienda consumir 2000 calorías. Alimentos para el día Contenido nutricional (calorías) Desayuno Trigo 92 Naranja 43 Trigo 92 Fresas 32 Quinoa 120 Manzana 52 Maíz 96 Comida Cordero 294 Zanahorias 41 Cordero 294 Pimientos rojos 40 Pescado 105 Espinacas 23 Cerdo 242 Cena Queso cottage 98 Yogur 61 Queso cottage 98 Queso cottage 98 Crema 340 Desea crear otro menú? (1: Sí, 0: No) </pre>
--	--

Si el usuario decide no solicitar más menús, tiene la opción de acceder a diversos reportes, tales como visualizar su historial completo o simplemente consultar el contenido calórico de los menús creados previamente.

```
3
Todos los menus creados
Fecha de creacion: Jun 16 2023
Hora de creacion: 21:23:52
Trigo 92
Naranja 43
Trigo 92
Fresas 32
Quinoa 120
Manzana 52
Maiz 96
Cordero 294
Zanahorias 41
Cordero 294
Pimientos rojos 40
Pescado 105
`ηEspinacas 23
Cerdo 242
Queso cottage 98
Yogur 61
Queso cottage 98
Queso cottage 98
Crema 340

1. Registrar datos
2. Recomendar menu
3. Generar reporte completo
4. Generar reporte mensual
5. Generar reporte diario
6. Generar reporte anual
7. Generar reporte calorico
8. Salir
7
El menu que creaste el dia: Jun 16 2023
A las: 21:23:52
Contiene: 2261 calorias
```

Argumentación de las partes del proyecto

Mediante el constructor de la clase 'Alimento', podemos crear un objeto de este tipo a partir de un archivo Excel. Sin embargo, es la clase 'Grupo' la que se encarga de leer y cargar este archivo, puesto que los alimentos se incorporan a un vector de este tipo, permitiéndonos cerrar el archivo y crear subgrupos a partir del vector en lugar de mantener el archivo abierto. Es crucial destacar la sobrecarga del operador "<<" para objetos del tipo 'Alimento', ya que esta funcionalidad es esencial para la construcción posterior del menú.

En la clase 'Menú', se estructuran los grupos de las distintas categorías de alimentos y se crean vectores del tipo 'Alimento' que almacenan la información de las comidas del día. Aquí se emplean diversos métodos getter y se crean las comidas en base al número de calorías ingresado, añadiendo alimentos de una categoría específica de manera aleatoria hasta sobrepasar el límite calórico establecido, mientras se imprimen simultáneamente.

La clase 'Persona', como se mencionó previamente, sirve para almacenar los datos personales y calcular el índice de masa corporal. Todos estos atributos son heredados por la clase 'Usuario', la cual implementa un objeto 'Persona' en su propio constructor, utilizando el concepto de polimorfismo. Dentro de la clase

'Usuario', se guardan los menús generados mediante el método 'Implementar Menú', el cual utiliza el IMC para generar una recomendación de calorías a consumir. El método 'Ver Historial' accede al vector de menús del usuario y lo recorre para imprimirlo, utilizando el operador "<<" que fue sobrecargado previamente en la clase 'Menú'.

La clase 'MyApp' se encarga de almacenar los usuarios en un vector, sobre el cual ejecuta funciones como recomendar un menú y acceder a los datos del usuario, mientras implementa menús inteligentes para el correcto funcionamiento de la aplicación. A partir de un menú inicial que ofrece opciones para registrarse o iniciar sesión, una vez que se ha iniciado sesión exitosamente, se puede acceder a todas las funcionalidades de la aplicación.

Identificación de casos que harían que el proyecto deje de funcionar

La aplicación presenta fallos en la gestión de datos proporcionados por el usuario y en el manejo de excepciones. Un ejemplo de esto es su forma de procesar la opción seleccionada por el usuario y el identificador de usuario, denotado por ".", en una misma instancia en lugar de en dos pasos distintos, lo que podría generar confusión para el usuario.

```
└─¡Bienvenido a Vida Sana!  
1. Iniciar sesion  
2. Registrarse  
3. Salir  
2.  
Introduzca un identificador de usuario  
Usuario creado con exito  
└─¡Bienvenido a Vida Sana!  
1. Iniciar sesion  
2. Registrarse  
3. Salir
```

Aunque no prevemos que la funcionalidad se detenga, sí creemos que el proceso de carga del archivo podría optimizarse. Actualmente, este proceso no permite la intervención del usuario y se realiza directamente en los métodos.

Además, tenemos cierta incertidumbre respecto al funcionamiento de los reportes diarios, mensuales y anuales. A pesar de nuestros intentos por modificar la fecha y hora del dispositivo en el que se ejecuta el programa, éste parece no verse afectado y continúa obteniendo la fecha y hora real.

Conclusión

El proyecto ha demostrado ser relevante y significativo en el contexto actual, donde la alimentación saludable es cada vez más una prioridad. Basándose en los principios de POO, se ha logrado un diseño robusto, adaptable y personalizado para responder a las necesidades cambiantes y específicas de los usuarios.

La herencia, el polimorfismo y la sobrecarga de operadores han sido herramientas cruciales para el desarrollo del proyecto, facilitando la representación de los alimentos y sus propiedades, el manejo uniforme de funciones y tipos de datos, y la ejecución de operaciones complejas de manera intuitiva, respectivamente.

La implementación exitosa del sistema cuenta con el potencial para llenar un vacío en el mercado de las aplicaciones de planificación dietética, proporcionando la construcción de menús personalizados y la entrega de información nutricional pertinente. Además, la capacidad de adaptarse a las características individuales de cada usuario, como la edad y el peso, y proporcionar recomendaciones de consumo calórico, ha demostrado ser una característica valiosa del sistema.

Por último, es importante destacar que este proyecto no sólo ha resultado en una herramienta eficiente y eficaz para la gestión de dietas, sino que también ha contribuido a la promoción de hábitos alimenticios saludables. En términos futuros, la adaptabilidad y flexibilidad del sistema permitirán la integración de nuevas características y mejoras para continuar respondiendo a las necesidades de los usuarios en el ámbito de la alimentación saludable.

Referencias consultadas

- *Tabla de calorías: calorías de cientos de alimentos.* (2019). TablaDeCalorias.net. Recuperado 12 de junio de 2023, de <https://www.tabladedecalorias.net/>
- cplusplus.com. (s.f.). *The C++ Resources Network*. Recuperado el 13 de junio de 2023, de <https://cplusplus.com/reference/stl/>