

Sistema de Gestion de Dietas para Pacientes

Descripción:

Desarrollar un sistema que permita a los nutricionistas gestionar planes de alimentación personalizados para sus pacientes. Se deberá crear un modelo de objetos que represente pacientes, nutricionistas y planes de dieta, almacenando la información relevante en archivos CSV. Todos los nombres de clases, métodos y comentarios en el código fuente se realizarán en inglés.

Requerimientos Funcionales:

1. Gestión de Pacientes:

- Implementar una clase Patient con atributos como patientId, name, age, weight, height, preexistingConditions.
- Incluir métodos para registrar, actualizar y eliminar la información de los pacientes.

2. Gestión de Nutricionistas:

- Diseñar una clase Dietitian con atributos tales como dietitianId, name, specialty.
- Permitir el registro y la gestión de nutricionistas.

3. Plan de Alimentación:

- Crear una clase DietPlan que vincule un Patient con un Dietitian e incluya detalles como planId, dailyCalories, macronutrientDistribution, specificRecommendations.
- Desarrollar métodos para crear y ajustar planes de alimentación.

4. Registro de Comidas:

- Implementar una clase Meal con atributos como name, macronutrients, calories, timeOfDay.
- Elaborar planes de comidas diarias que se alineen con la estrategia dietética general.

5. **Persistencia de Datos:**

- Leer y guardar información sobre pacientes, nutricionistas y planes de dieta en archivos CSV.

6. **Interfaz de Usuario:**

- Desarrollar una interfaz de consola que permita a los usuarios (nutricionistas) interactuar con el sistema, gestionar pacientes y sus planes de dieta.

Requerimientos No Funcionales:

- **Principios de Orientación a Objetos:**

- Uso efectivo de encapsulamiento, herencia y polimorfismo.

- **Manejo de Excepciones:**

- Manejo adecuado de errores y excepciones, especialmente en la entrada/salida de datos.

- **Código Limpio y Mantenable:**

- El código fuente y los comentarios estarán en inglés.

- **Documentación:**

- Comentarios detallados en el código y documentación del usuario sobre la operación del sistema.

Entregables:

1. **Código Fuente:**

- Código fuente completo del proyecto en Java, con nombres de clases, métodos y comentarios en inglés.

2. **Archivos CSV:**

- Ejemplos de archivos CSV para pacientes, nutricionistas y planes de dieta.

3. **Documentación del Sistema:**

- Un archivo README.md que describa las clases y métodos utilizados en el sistema.

4. Informe del Proyecto:

- Un documento que describa el proceso de diseño, los problemas encontrados y cómo se solucionaron, junto con propuestas de mejoras.

Evaluación:

El proyecto se evaluará basándose en la correcta implementación de los conceptos de POO, la funcionalidad del sistema, la calidad y claridad del código, el manejo de archivos CSV.

Este proyecto permitirá aplicar conocimientos de POO en un escenario que simula una aplicación real, reforzando la capacidad para manejar datos y archivos, y promoviendo la importancia de crear software que sea fácil de entender y mantener.