Patient Diet Management System

Proyecto final prog 2

Ana Isabella Suárez Cardona

34335

Jesús David Milian Saza

46632

Se desarrolla un sistema que permita a los nutricionistas gestionar planes de alimentación personalizados para sus pacientes por tanto se deben implementar los siguientes paso:

**1.** Modelado de Objetos:

Se crearon las siguientes clases principales para representar entidades clave:

**Patient:** La clase Patient representa a un paciente.

Incluye métodos para registrar, actualizar y eliminar pacientes.

Tiene los siguientes **atributos**:

private int patientId;

private String name;

private int age;

private double weight;

private double height;

private List<String> preexistingConditions;

El **constructor** de la clase Patient inicializa los atributos de la siguiente manera:

public Patient(int patientId, String name, int age, double weight, double height, List<String> preexistingConditions) {

this.patientId = patientId;

this.name = name;

this.age = age;

this.weight = weight;

this.height = height;

this.preexistingConditions = preexistingConditions; }

La clase Patient proporciona los siguientes **métodos**:

getPatientId(): Devuelve el identificador del paciente.

setName(): Establece el nombre del paciente.

getName(): Devuelve el nombre del paciente.

setAge(): Establece la edad del paciente.

getAge(): Devuelve la edad del paciente.

setWeight(): Establece el peso del paciente.

getWeight(): Devuelve el peso del paciente.

setHeight(): Establece la altura del paciente.

getHeight(): Devuelve la altura del paciente.

setPreexistingConditions(): Establece las condiciones médicas preexistentes del paciente.

getPreexistingConditions(): Devuelve las condiciones médicas preexistentes del paciente.

Clase Dietitian

**Dietitian:** Contiene métodos para registrar y gestionar nutricionistas.

Tiene los siguientes **atributos**:

private int dietitianId;

private String name;

private String specialty;

El **constructor** de la clase Dietitian inicializa los atributos de la siguiente manera:

public Dietitian(int dietitianId, String name, String specialty) {

this.dietitianId = dietitianId;

this.name = name;

this.specialty = specialty; }

La clase Dietitian proporciona los siguientes **métodos**:

getDietitianId(): Devuelve el identificador del nutricionista.

setName(): Establece el nombre del nutricionista.

getName(): Devuelve el nombre del nutricionista.

setSpecialty(): Establece la especialidad del nutricionista.

getSpecialty(): Devuelve la especialidad del nutricionista.

**DietPlan:** Contiene métodos para crear y ajustar planes de alimentación.

Tiene los siguientes **atributos**:

private int planId;

private int patientId;

private int dietitianId;

private int dailyCalories;

private Map<String, Double> macronutrientDistribution;

private List<String> specificRecommendations;

El **constructor** de la clase DietPlan inicializa los atributos de la siguiente manera:

public DietPlan(int planId, int patientId, int dietitianId, int dailyCalories, Map<String, Double> macronutrientDistribution, List<String> specificRecommendations) {

this.planId = planId;

this.patientId = patientId;

this.dietitianId = dietitianId;

this.dailyCalories = dailyCalories;

this.macronutrientDistribution = macronutrientDistribution;

this.specificRecommendations = specificRecommendations; }

La clase DietPlan proporciona los siguientes **métodos**:

getPlanId(): Devuelve el identificador del plan de dieta.

setPatientId(): Establece el identificador del paciente asociado al plan de dieta.

getPatientId(): Devuelve el identificador del paciente asociado al plan de dieta.

setDietitianId(): Establece el identificador del nutricionista asociado al plan de dieta.

getDietitianId(): Devuelve el identificador

**Meal:** Representa una comida con atributos como name, macronutrientes, calorías, y timeOfDay.

Tiene los siguientes **atributos**:

private int mealId;

private String name;

private Map<String, Double> macronutrients;

private int calories;

private String timeOfDay;

El **constructor** de la clase Meal inicializa los atributos de la siguiente manera:

public Meal(int mealId, String name, Map<String, Double> macronutrients, int calories, String timeOfDay) {

this.mealId = mealId;

this.name = name;

this.macronutrients = macronutrients;

this.calories = calories;

this.timeOfDay = timeOfDay;

}

La clase Meal proporciona los siguientes **métodos**:

getMealId(): Devuelve el identificador de la comida.

setName(): Establece el nombre de la comida.

getName(): Devuelve el nombre de la comida.

setMacronutrients(): Establece los macronutrientes contenidos en la comida.

getMacronutrients(): Devuelve los macronutrientes contenidos en la comida.

setCalories(): Establece las calorías de la comida.

getCalories(): Devuelve las calorías de la comida.

setTimeOfDay(): Establece la hora del día en la que se debe consumir la comida.

getTimeOfDay(): Devuelve la hora del día en la que se debe consumir la comida.

**Csv File Manager:** Proporciona métodos para la lectura y escritura de datos en archivos CSV.

**Console Interface:** Se encarga de la interacción del usuario a través de la consola. Contiene métodos para gestionar pacientes, nutricionistas y planes de dieta.

Paquete de clases

**2.** Principios de Orientación a Objetos: Se emplearon los principios de encapsulamiento, herencia y polimorfismo para diseñar clases eficientes y sostenibles.