



E.S. Ingeniería y Tecnología

Ingeniería Informática y de Sistemas

Área: Lenguajes y Sistemas Informáticos

Lenguajes y Paradigmas de Programación

Práctica de laboratorio #8

Esta práctica de laboratorio se ha de realizar utilizando el lenguaje de programación Ruby y los conceptos de **herencia** y **módulo** del paradigma de **Programación Orientada a Objetos**.

Se ha de partir de la estructura de la gema generada con la herramienta Bundler y realizar el control de versiones con git siguiendo su filosofía de ramas.

Para el desarrollo de las clases se ha de seguir la metodología de Desarrollo dirigido por pruebas (*Test Driven Development - TDD*) y la herramienta RSpec.

1. Considere la jerarquía de clases Ruby para representar *alimentos* con la eficiencia energética de su producción. Incorporar las modificaciones necesarias de manera que las instancias sean comparables.
2. Genere pruebas para comparar la información nutricional y ambiental entre dos alimentos. Utilizar todos los operadores disponibles en el *mixin*.
3. Considere la clase Ruby para representar y gestionar *listas doblemente enlazadas*. Incorporar las modificaciones necesarias de manera que las instancias sean enumerables.
4. Genere pruebas para enumerar listas de alimentos. Utilizar los métodos collect, select, max, min, sort.
5. Desarrollar una clase Ruby para representar la información nutricional de los *platos* de un menú de una dieta.

Ejemplos de expectativas son las siguientes:

- Obteniendo el nombre de un plato
 - Existe un conjunto de alimentos.
 - Existe un conjunto de cantidades de alimentos en gramos.
 - Porcentaje de proteínas del conjunto de alimentos.
 - Porcentaje de lípidos del conjunto de alimentos.
 - Porcentaje de hidratos de carbono del conjunto de alimentos.
 - Valor Calórico Total (V.C.T) del conjunto de alimentos expresado en kilocalorías.
 - Se obtiene el plato formateado.
6. Cree una jerarquía de clases a partir de la clase desarrollada en 5, de manera que se contemple la eficiencia energética en la producción de los alimentos de los *platos* de un menú de una dieta.
- Ejemplos de expectativas son las siguientes:

- Valor total de la emisiones diarias de gases de efecto invernadero.
- Estimación de los metros cuadrados de uso de terreno.
- Se obtiene la eficiencia energética formateada.

Además, genere pruebas para comprobar la **clase** de un objeto, el **tipo** de un objeto y su **pertenencia a una jerarquía**.

7. Considere la jerarquía de clases que permite la creación de *platos* de un menú de una dieta creada en 6. Incorporar las modificaciones necesarias de manera que las instancias sean comparables.
8. Genere pruebas para comparar la valoración nutricional y la eficiencia energética entre *platos* de un menú de una dieta. Utilizar todos los operadores disponibles.

Se ha de trabajar con cinco dietas:

- *Española* (40 % de carbohidratos, 40 % de grasas, 20 % de proteínas)
- *Vasca* (60 % de carbohidratos, 25 % de grasas, 15 % de proteínas)
- *Vegetaria* (incluye leche y huevos) asume las mismas proporciones que la *española* y la porción de dieta animal correspondiente a carnes y pescados se reparte como un 85 % a la leche y un 15 % a los huevos.
- *Vegetaliana* no incorpora ningún alimento de origen animal y la porción de dieta correspondiente a proteínas se reparte de forma proporcional entre el resto.
- *Locura por la carne* que asume un consumo de carne del 50 % .

9. Genere pruebas para enumerar listas de *platos* de un menú de una dieta. Utilizar los métodos collect, select, max, min, sort.

Se ha de trabajar con las cinco dietas descritas en el punto anterior.

10. Escribir la dirección HTTP del repositorio de la organización ‘ULL-ESIT-LPP-1920/tdd’ en la tarea habilitada en el campus virtual.

(<https://github.com/ULL-ESIT-LPP-1920/tdd-aluXXX.git>)