Universidad de La Laguna
E.S. Ingeniería y Tecnología
Ingeniería Informática
Lenguajes y Sistemas Informáticos

Lenguajes y Paradigmas de Programación
Práctica de laboratorio **#10**

Esta práctica de laboratorio se ha de realizar utilizando el lenguaje de programación Ruby y el paradigma de programación **funcional**.

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 del código se ha de seguir la metodología de Desarrollo dirigido por pruebas (*Test Driven Development - TDD*) y utilizar la herramienta RSpec.

Considere la jerarquía de clases Ruby para representar *alimentos* y *listas* de prácticas anteriores.

Considere también, la tabla de alimentos por grupo de prácticas anteriores:

Huevos, lácteos y helados

	Proteínas	Glúcidos	Lípidos
Huevo frito	14.1	0.0	19.5
Leche vaca	3.3	4.8	3.2
Yogurt	3.8	4.9	3.8

Carnes y derivados

	Proteínas	Glúcidos	Lípidos
Cerdo	21.5	0.0	6.3
Ternera	21.1	0.0	3.1
Pollo	20.6	0.0	5.6

Pescados y mariscos

	Proteínas	Glúcidos	Lípidos
Bacalao	17.7	0.0	0.4
Atún	21.5	0.0	15.5
Salmón	19.9	0.0	13.6

Alimentos grasos

	Proteínas	Glúcidos	Lípidos
Aceite de oliva	0.0	0.2	99.6
Mantequilla	0.7	0.0	83.2
Chocolate	5.3	47.0	30.0

Alimentos ricos en carbohidratos

	Proteínas	Glúcidos	Lípidos
Azúcar	0.0	99.8	0.0
Arroz	6.8	77.7	0.6
Lentejas	23.5	52.0	1.4
Papas	2.0	15.4	0.1

Verduras y Hortalizas

	Proteínas	Glúcidos	Lípidos
Tomate	1.0	3.5	0.2
Cebolla	1.3	5.8	0.3
Calabaza	1.1	4.8	0.1

Frutas

	Proteínas	Glúcidos	Lípidos
Manzana	0.3	12.4	0.4
Plátanos	1.2	21.4	0.2
Pera	0.5	12.7	0.3

1. Escriba el código que permita la representación de la tabla mendiante un array.

2. Escriba el código que permita obtener un nuevo array con los elementos ordenados por su valor energético usando un bucle for.

3. Escriba el código que permita obtener un nuevo array con los elementos ordenados por su valor energético usando el método each.

4. Escriba el código que permita obtener un nuevo array con los elementos ordenados por su valor energético usando el método sort.

5. Haciendo uso del método `benchmark` de la clase `Benchmark` muestre un informe del tiempo de los ejercicios 2, 3, 4.

6. En este momento se debería comprobar con la herramienta ***Coverall*** la historia del ***cubrimiento*** y las estadísticas del código Ruby desarrollado.

   - Darse de alta en `https://https://coveralls.io/` y permitir que se acceda desde *Github* (poniendo a ON el proyecto).
   - Crear un fichero `.coveralls.yml` que contenga la configuración, por ejemplo:

     ```
     service_name: travis-ci
     ```

   - Especificar la dependencia de desarrollo de esta gema en el fichero `.gemspec`

     ```
     spec.add_development_dependency "coveralls"
     ```

   - Ejecutar `bundler` para que se instalen las dependencias:

     ```
     bin/setup
     ```

   - Incluir en el fichero `spec_helper.rb` el código de `coverall`:

     ```
     require 'coveralls'
     Coveralls.wear!
     ```

   Sin embargo, como los repositorios con los que se está trabajando son privados se realizarán las pruebas con la gema pública de la clase textsfPoint.

7. Escribir la dirección HTTP del repositorio de la organización 'ULL-ESIT-LPP-1718/tdd' en la tarea habilitada en el campus virtual.

   (`https://github.com/ULL-ESIT-LPP-1718/tdd-aluXXX.git`)