

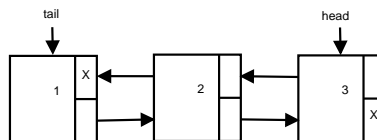
Esta práctica de laboratorio se ha de realizar utilizando el lenguaje de programación Ruby y el paradigma de **Programación Orientada a Objetos**.

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. Crear una clase Ruby para representar y gestionar listas doblemente enlazadas (*List*) de manera que se puedan recorrer en ambos sentidos.

Por ejemplo, si se considera la lista (1,2,3), su representación como una lista doblemente enlazada sería:

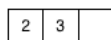


En Ruby, puesto que no existe un objeto *List*, se puede conseguir lo mismo con *Arrays*:

```
a = []
a.unshift 3
a.unshift 2
```

```
a = []
a << 2
a << 3
```

Internamente, Ruby creará una array como este:



Por lo tanto, los arrays de Ruby pueden funcionar como listas enlazadas, puesto que los métodos **push**, **unshift** e **insert** añaden un nuevo elemento a la lista y los métodos **pop** y **shift** eliminan un elemento de la lista. Sin embargo, NO SE UTILIZARÁ ESTA APROXIMACIÓN. Se abordará la implementación mediante la creación de una estructura de datos. Utilizar un **Struct** para su definición:

```
# create a Struct with :value, :next and :prev
Node = Struct.new(:value, :next, :prev)
```

Ejemplos de descripción de expectativas son las siguientes:

- Debe existir un Nodo de la lista con sus datos, su siguiente y su previo
- Debe existir una Lista con su cabeza y su cola
- Se puede insertar un elemento en la Lista
- Se pueden insertar varios elementos en la Lista
- Se extrae el primer elemento de la Lista
- Se extrae el último elemento de la Lista

2. Considere las clases Ruby para representar alimentos de prácticas anteriores. Siguiendo la “pirámide alimenticia”, cree una jerarquía de clases que permita clasificar los alimentos **por grupos**: ‘*huevos, lácteos y helados*’, ‘*carnes y derivados*’, ‘*pescados y mariscos*’, ‘*alimentos grasos*’, ‘*alimentos ricos en carbohidratos*’, ‘*verduras y hortalizas*’, ‘*frutas*’ y ‘*bebidas*’

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

Además cree pruebas para la siguiente lista de alimentos:

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

3. Utilizar la herramienta *Guard* de *Comprobación Continua* con el código Ruby desarrollado. Se ha de configurar de manera que permita la ejecución de las pruebas definidas con Rspec cuando se modifiquen.

- Añadir *Guard* como una dependencia de desarrollo en el fichero de especificaciones de la gema `.gemspec`.

```
spec.add_development_dependency "guard"  
spec.add_development_dependency "guard-rspec"  
spec.add_development_dependency "guard-bundler"
```

- Instalar *Guard* a través de *Bundler*.

```
bin/setup
```

- Generar un fichero de guardia por defecto (*Guardfile*).

```
bundle exec guard init
```

- Ejecutar *Guard* a través de *Bundler* en una terminal.

```
bundle exec guard
```

4. 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>)