

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.

1. Escribir un método para calcular el **Índice Glucémico** de un alimento que siga el paradigma de programación funcional.

El Área Incremental Bajo la Curva (AIBC) se determina como la suma de 24 áreas trapezoidales (S_i), donde: G_0 es la concentración de glucosa antes de cada ingesta, G_i es la concentración de glucosa en el tiempo i y Δt es el intervalo de tiempo entre las mediciones (en este caso siempre 5 minutos) :

$$AIBC = \sum_{i=1}^{24} S_i; \quad S_i = \frac{(G_i - G_0) + (G_{i-1} - G_0)}{2} \times \Delta t$$

Un código fuente para calcular AIBC, usando programación imperativa, es el siguiente:

```
def aibc
  i = 0
  r = []
  while i < g.size
    index = 1
    s = []
    while index < g[i].size
      if g[i][index] < g[i][0]
        s << 0.0
      else
        s << (((g[i][index] - g[i][0]) + (g[i][index-1] - g[i][0]))/2)*5
      end
      index = index + 1
    end
    r << s
    i = i + 1
  end

  suma = []
  j = 0
  while j < g.size
    k = 0
    s = 0
    while k < r[j].size
      s = s + r[j][k]
      k = k + 1
    end
    suma << s
    j = j + 1
  end
  suma
end
```

Se calcula el índice glucémico relativo a un individuo IG_{ind} para un alimento como:

$$IG_{ind} = \frac{AIBC_{alimento}}{AIBC_{glucosa}} \times 100[\%]$$

Finalmente, el **Índice Glucémico** (IG) de un alimento se calcula como la media de los índices glucémicos de los individuos del grupo.

$$IG = \frac{\sum_{i=1}^N IG_{ind}}{N}$$

Un código fuente para calcular IG, usando programación imperativa, es el siguiente:

```
ig_alimento = (((aibc_alimento[0] /aibc_glucosa[0])*100) + (((aibc_alimento[1] /aibc_glucosa[1]))*100))/2
```

Escriba un código fuente que realice los mismos cálculos que los anteriores utilizando métodos como `collect`, `reduce`, `zip`, etc.

2. Comprobar con la herramienta **Travis** de **Integración Continua**. el código Ruby desarrollado. Con ella se comprueba la portabilidad de los desarrollos entre distintas plataformas y versiones de Ruby.

El fichero `.travis.yml` contiene el listado de plataformas sobre las que se quiere comprobar el correcto funcionamiento.

La comprobación se activa cada vez que se hace un *push* al repositorio de la gema.

3. 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>)