



IIC2113 - Diseño Detallado de Software

Interrogación 2

Instrucciones: Sea preciso: no es necesario escribir mucho pero sí ser preciso. En caso de ambigüedad, utilice su criterio y explicita los supuestos que considere convenientes. Esta interrogación fue diseñada para durar 80 minutos.

Responda cada pregunta en hojas separadas y recuerde poner su nombre en cada una.

1. (1.0 pt) Nombre 3 características de una buena métrica. Explique el objetivo principal para utilizar métricas en un proyecto de *software*.
2. (0.9 pts) Nombre y explique 3 métricas propuestas por Chidamber & Kemerer (*CK metrics suite*).
3. (0.6 pts) Desarrolle 3 razones por las cuales se podría explicar que “buenos desarrolladores” generen “mal código” (*code smells*).

4. (1.8 pts) Identifique *code smells* presentes en el siguiente extracto de código. Explique qué problemas generan cada uno de estos *code smells*.

```
1  class Pokedex
2    def initialize
3      # Lista para guardar pokemones
4      @pl = []
5    end
6
7    # Agrega nombre, ataque y defensa de un pokémon a la lista
8    def add_pokemon(n, a, d)
9      @pl << [n, a, d]
10    end
11
12    # Para aumentar el ataque de un pokémon
13    def increase_attack(n)
14      # Guarda si el pokémon fue encontrado
15      f = false
16      for pk in @pl
17        if pk[0] == n
18          pk[1] += 1
19          f = true
20        end
21      end
22      if !f
23        puts n + " no encontrado"
24      end
25    end
26  end
```

```

27  ··# Para aumentar la defensa de un pokémon↵
28  ··def increase_defense(name)↵
29  ···# Guarda si el pokémon fue encontrado↵
30  ···f = false↵
31  ···for pk in @pl↵
32  ····if pk[0] == name↵
33  ·····pk[2] += 1↵
34  ·····f = true↵
35  ····end↵
36  ···end↵
37  ···if !f↵
38  ····puts name + " no encontrado"↵
39  ···end↵
40  ··end↵
41  ↵
42  ··# Para obtener los atributos de un pokémon↵
43  ··def get_stats(name)↵
44  ···for pk in @pl↵
45  ····if pk[0] == name↵
46  ·····puts pk[1].to_s + "/" + pk[2].to_s↵
47  ····end↵
48  ···end↵
49  ··end↵
50  ↵

```

```

51  ··# Se llama al invocar puts sobre una instancia↵
52  ··def to_s↵
53  ··  i = 0↵
54  ··  temp = ""↵
55  ··  for pk in @pl↵
56  ··    i += 1↵
57  ··    temp += i.to_s + ". " + pk[0] + "\n"↵
58  ··  end↵
59  ··  temp↵
60  ··end↵
61  end↵
62  ↵
63  pokedex = Pokedex.new↵
64  pokedex.add_pokemon('Pikachu', 12, 10)↵
65  pokedex.add_pokemon('Cubone', 8, 12)↵
66  pokedex.get_stats('Pikachu')↵
67  puts pokedex↵

```

5. (0.5 pts) Con respecto a los atributos de calidad *Maintainability*, *Flexibility* y *Testability* de una aplicación: ¿cuál es el *trade-off* que se debe considerar al momento de desarrollar un proyecto?
6. (1.2 pts) Diseñe *tests* unitarios para el siguiente código. No es necesario que implemente código, aunque puede utilizar pseudo-código si lo desea. Se pide como mínimo que describa las pruebas con una breve descripción del escenario, un contexto (configuración inicial) y un resultado esperado. Considere que se descontará puntaje por *tests* redundantes.

```
1  class Item
2    attr_reader :name, :size
3    def initialize(name, size)
4      @size = size
5      @name = name
6    end
7  end

9  class Fridge
10   def initialize(max_space = 10)
11     @max_space = max_space
12     @current_space = 0
13     @items = {}
14   end
15
16   def store(item)
17     item_size = item.size
18     item_name = item.name
19     item_amount = @items[item_name]
20     if @current_space + item_size <= @max_space
21       @current_space += item_size
22       if item_amount
23         item_amount += 1
24       else
25         item_amount = 1
26       end
27       @items[item_name] = item_amount
28       puts "Se guardo un #{item_name}"
29     else
30       puts "No hay espacio para guardar #{item_name}"
31     end
32   end
33 end
```

```
34 · def take(item)↵
35 ·   · item_name = item.name↵
36 ·   · item_amount = @items[item_name]↵
37 ·   · if item_amount↵
38 ·     · item_amount -= 1↵
39 ·     · @current_space += item.size↵
40 ·     · @items[item_name] = item_amount↵
41 ·     · puts "Quedan #{item_amount} unidades de #{item_name}"↵
42 ·   · else↵
43 ·     · puts "No hay unidades de #{item_name}"↵
44 ·   · end↵
45 · end↵
46 end↵
```