



Examen

Pregunta 1

```
1 class Salesman
2   attr_accessor :commissions, :sponsor, :n_sponsors
3
4   def initialize
5     @commissions = 0
6     @sponsor = nil
7     @n_sponsors = 0
8   end
9
10  def pay_sponsor(commission)
11    @commissions += commission
12    if @sponsor
13      @sponsor.pay_sponsor(commission)
14    end
15  end
16 end
17
18 class MotorcycleSalesman < Salesman
19   def complete_sale(amount)
20     fraction = 0.03
21     commission = fraction * amount # Sin tope
22
23     # Recibir comisión
24     @commissions += commission * 10/(@n_sponsors + 10)
25     # Pagar comisión a padrinos
26     if @sponsor
27       @sponsor.pay_sponsor(commission * 1/(@n_sponsors + 10))
28     end
29   end
30 end
31
32 class CarSalesman < Salesman
33   def complete_sale(amount)
34     fraction = amount < 20000000 ? 0.02 : 0.05
35     commission = [fraction * amount, 2000000].min # No más de dos millones
36
37     # Recibir comisión
```

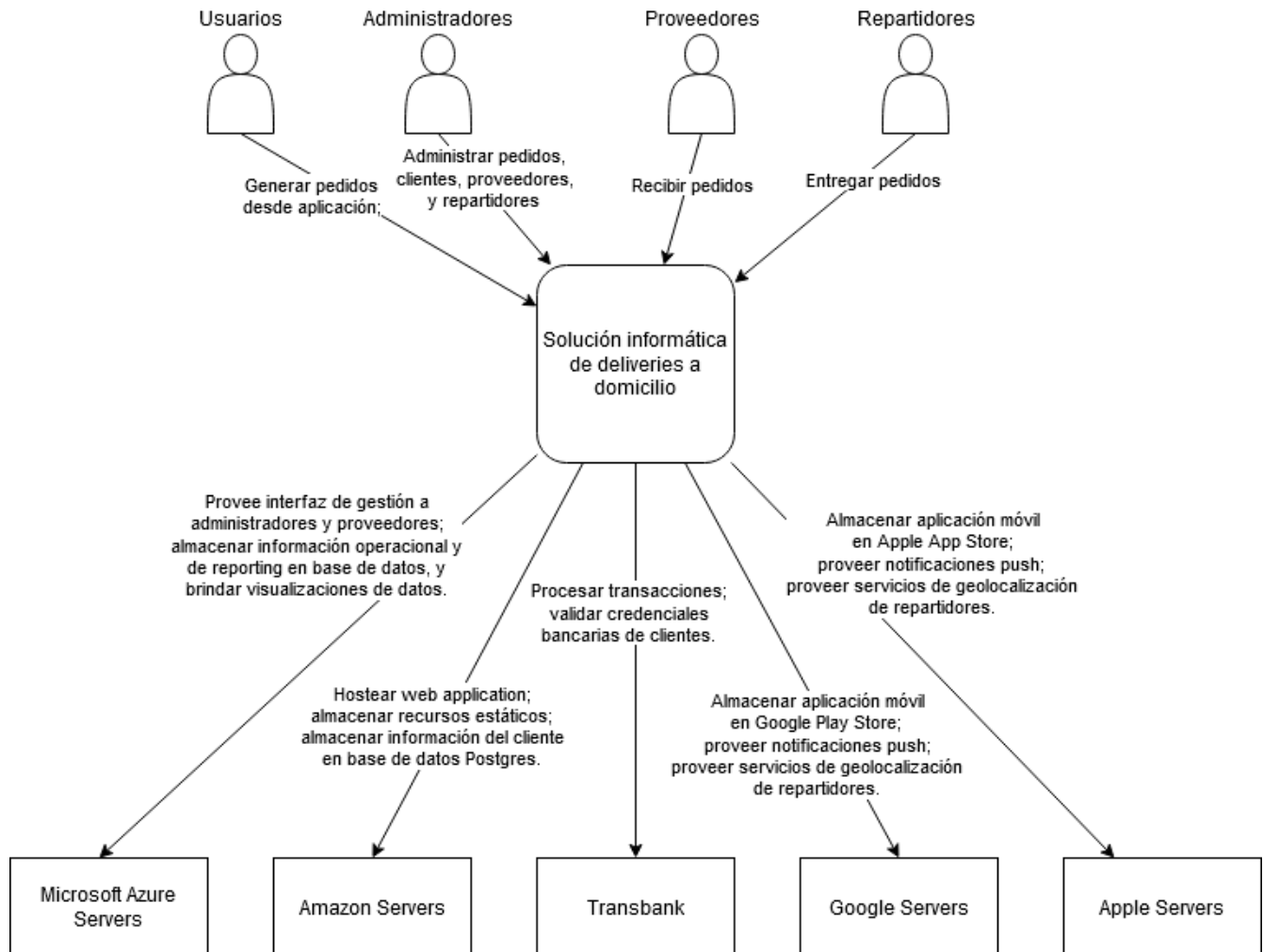
```

38     @commissions += commission * 10/(@n_sponsors + 10)
39     # Pagar comisi3n a padrinos
40     if @sponsor
41         @sponsor.pay_sponsor(commission * 1/(@n_sponsors + 10))
42     end
43 end
44 end
45
46 class SalesmanFactory
47     attr_accessor :salesmen
48
49     def initialize
50         @salesmen = []
51     end
52
53     def create_salesman
54         # create instance
55         if rand < 0.5
56             salesman = MotorcycleSalesman.new
57         else
58             salesman = CarSalesman.new
59         end
60
61         # Assign sponsor randomly from pointer list (padrino)
62         sponsor = @salesmen.sample
63         # If list is empty sponsor = nil; doesn't change anything
64         salesman.sponsor = sponsor
65         # Check n_sponsors
66         current_salesman = salesman
67         while current_salesman.sponsor
68             current_salesman = current_salesman.sponsor
69             salesman.n_sponsors += 1
70         end
71
72         # Add instance to list
73         @salesmen << salesman
74         salesman
75     end
76 end

```

Pregunta 2

Diagrama de contexto



Pregunta 3

Parte a)

Dado que este equipo de desarrollo se compone de 3 personas, una con jornada completa y 2 con media jornada, se cuenta con 88 staff hours semanales.

Para el primer release, con sprints semanales, y asumiendo un esfuerzo pesimista, se puede establecer la siguiente planificación:

| Sprint | Relato | Esfuerzo (Pesimista) | Total Esfuerzo | Prioridad |
|----------|--------|----------------------|----------------|-----------|
| Sprint 1 | 2 | 30 | 80 | a |
| | 3 | 18 | | a |
| | 11 | 32 | | a |
| Sprint 2 | 5 | 36 | 84 | a |
| | 7 | 18 | | a |
| | 9 | 30 | | m |
| Sprint 3 | 1 | 24 | 88 | m |
| | 4 | 24 | | m |
| | 12 | 40 | | b |
| Sprint 4 | 6 | 30 | 84 | b |
| | 8 | 24 | | m |
| | 10 | 30 | | b |

Dado que se cuenta con un mes de plazo, con esta planificación sería posible cumplir con las expectativas, incluso asumiendo un enfoque pesimista.

Parte b)

Siguiendo la lógica de sprints semanales, se puede definir el siguiente orden para los releases requeridos

| Release | Sprints totales | Funcionalidades | Esfuerzo (Pesimista) | Total Esfuerzo |
|-----------|-----------------|-----------------|----------------------|----------------|
| Release 1 | 6 | 2 | 250 | 450 |
| | | 4 | 200 | |
| Release 2 | 5 | 3 | 300 | 390 |
| | | 6 | 90 | |
| Release 3 | 4 | 5 | 340 | 340 |

De esta manera, se necesitarían 15 sprints para cumplir con el resto de las funcionalidades, los cuales se pueden completar dentro del plazo de 5 meses. Esto, sumado a la planificación del primer release (primer mes), permitiría finalizar el producto dentro del plazo total de 6 meses, con el equipo actual, incluso con un enfoque pesimista.

Pregunta 4

Consideraciones generales

- Si bien en un contexto lógico, el hecho de que un cliente haga su primera compra implica que su antigüedad sea de 0 meses, para efectos de *testing*, decidí incluir casos en los cuales se mezcle la primera compra con antigüedad variable, con el objetivo de asegurar la robustez de la clase.
- Para los valores de monto, se usaron {5000, 9999, 10000, 50000, 99999, 100000, 500000}.
- Para los valores de antigüedad, se usaron {0, 3, 5, 6, 12}.
- De esta manera, se tienen en cuenta también los casos bordes que se le puede presentar a nuestra clase.

| Test | Monto (\$) | Es primera compra (bool) | Antigüedad (meses) |
|------|------------|--------------------------|--------------------|
| 1 | 5000 | TRUE | 0 |
| 2 | 9999 | TRUE | 0 |
| 3 | 10000 | TRUE | 0 |
| 4 | 50000 | TRUE | 0 |
| 5 | 99999 | TRUE | 0 |
| 6 | 100000 | TRUE | 0 |
| 7 | 500000 | TRUE | 0 |
| 8 | 5000 | FALSE | 0 |
| 9 | 9999 | FALSE | 0 |
| 10 | 10000 | FALSE | 0 |
| 11 | 50000 | FALSE | 0 |
| 12 | 99999 | FALSE | 0 |
| 13 | 100000 | FALSE | 0 |
| 14 | 500000 | FALSE | 0 |
| 15 | 5000 | TRUE | 3 |
| 16 | 9999 | TRUE | 3 |
| 17 | 10000 | TRUE | 3 |
| 18 | 50000 | TRUE | 3 |
| 19 | 99999 | TRUE | 3 |
| 20 | 100000 | TRUE | 3 |
| 21 | 500000 | TRUE | 3 |
| 22 | 5000 | FALSE | 3 |
| 23 | 9999 | FALSE | 3 |
| 24 | 10000 | FALSE | 3 |
| 25 | 50000 | FALSE | 3 |
| 26 | 99999 | FALSE | 3 |
| 27 | 100000 | FALSE | 3 |
| 28 | 500000 | FALSE | 3 |
| 29 | 5000 | TRUE | 5 |
| 30 | 9999 | TRUE | 5 |
| 31 | 10000 | TRUE | 5 |
| 32 | 50000 | TRUE | 5 |
| 33 | 99999 | TRUE | 5 |
| 34 | 100000 | TRUE | 5 |
| 35 | 500000 | TRUE | 5 |
| 36 | 5000 | FALSE | 5 |

| | | | |
|----|--------|-------|----|
| 37 | 9999 | FALSE | 5 |
| 38 | 10000 | FALSE | 5 |
| 39 | 50000 | FALSE | 5 |
| 40 | 99999 | FALSE | 5 |
| 41 | 100000 | FALSE | 5 |
| 42 | 500000 | FALSE | 5 |
| 43 | 5000 | TRUE | 6 |
| 44 | 9999 | TRUE | 6 |
| 45 | 10000 | TRUE | 6 |
| 46 | 50000 | TRUE | 6 |
| 47 | 99999 | TRUE | 6 |
| 48 | 100000 | TRUE | 6 |
| 49 | 500000 | TRUE | 6 |
| 50 | 5000 | FALSE | 6 |
| 51 | 9999 | FALSE | 6 |
| 52 | 10000 | FALSE | 6 |
| 53 | 50000 | FALSE | 6 |
| 54 | 99999 | FALSE | 6 |
| 55 | 100000 | FALSE | 6 |
| 56 | 500000 | FALSE | 6 |
| 57 | 5000 | TRUE | 12 |
| 58 | 9999 | TRUE | 12 |
| 59 | 10000 | TRUE | 12 |
| 60 | 50000 | TRUE | 12 |
| 61 | 99999 | TRUE | 12 |
| 62 | 100000 | TRUE | 12 |
| 63 | 500000 | TRUE | 12 |
| 64 | 5000 | FALSE | 12 |
| 65 | 9999 | FALSE | 12 |
| 66 | 10000 | FALSE | 12 |
| 67 | 50000 | FALSE | 12 |
| 68 | 99999 | FALSE | 12 |
| 69 | 100000 | FALSE | 12 |
| 70 | 500000 | FALSE | 12 |