

## **Pregunta 1 (30 puntos)**

Estos son los relatos que se desprenden directamente de la descripción del problema

### **poner un pedido de gas**

yo como cliente de la empresa de gas necesito ingresar un nuevo pedido de gas para que en el menor plazo de tiempo posible pueda disponer de este producto

### **anular pedido de gas**

yo como cliente de la empresa de gas quiero anular un pedido ya ingresado de gas para la eventualidad de que alguien de mi familia ya haya comprado antes

### **generar listas de despacho**

yo como gerente de operaciones necesito a primera hora de la mañana generar una lista con los despachos a realizar en el día

### **generar recorrido del camión**

yo como gerente de operaciones necesito generar el detalle del recorrido de cada camión para entregarlo al conductor

### **dilatar próxima dirección**

yo como conductor necesito alterar el recorrido indicado por imposibilidad temporal de entrega en la siguiente dirección

### **entregar producto al cliente**

yo como conductor necesito ingresar los datos de quien recibe el producto, el monto pagado para que el pedido quede registrado como entregado

### **obtener detalle entregas del día**

yo como gerente de operaciones necesito conocer en cualquier momento el estado de las entregas del día indicando para cada pedido su status y el conductor asociado para controlar el avance y la normalidad del proceso

### **obtener ventas del día**

yo como gerente de la empresa necesito saber cuanto se ha vendido en el día para tener una idea instantanea de la salud del negocio

### **obtener ventas mensuales por producto**

yo como gerente general de la empresa necesito saber cuanto se vendió el mes pasado de cada producto para planificar una estrategia de publicidad y manejar las compras

### **obtener ventas mensuales por comuna**

yo como gerente general de la empresa necesito conocer el detalle de ventas por comuna para poder planear una estrategia de marketing adecuada

Hay algunos relatos que no se deducen directamente del enunciado pero que deberían estar:

**registrarse como nuevo cliente**

yo como cliente de la empresa de gas necesito registrarme en la aplicación para poder poner pedidos ahora o mas adelante

**registrar nuevo conductor**

yo como gerente de operaciones necesito registrar a un nuevo conductor de modo que esté disponible para la distribución diaria

**registrar nuevo camión**

yo como gerente de operaciones necesito ingresar un nuevo camión a la flota para que pueda ser utilizado en las entregas

**asignar conductor a camión**

yo como gerente de operaciones necesito asignar un conductor a un camión para las entregas del día de modo de poder tener flexibilidad de cambio de conductores por enfermedad o despido

**Notas de Pauta**

- es muy importante correcto uso de la técnica
  - título del relato comienza con verbo y da una idea de la meta del actor
  - actores claros después del "yo" y siempre el mismo si corresponde al mismo actor en otro relato
  - descripción tiene que tener las 3 partes: yo como, necesito que, para tal cosa
- premiar la completitud de los relatos pero si falta uno o dos no castigar demasiado

## Pregunta 2 (20 puntos)

a) (5 puntos)

```
def CostoEnvio(distancia, peso)
  if distancia <= 1
    if peso <= 5
      return 2000
    else
      return 3000
    end
  elsif distancia <= 10
    if peso <= 5
      return 4000
    else
      return 5000
    end
  else
    if peso <= 5
      return 6000
    else
      return 7000
    end
  end
end
```

b) (5 puntos) Diseña pruebas de caja blanca que entreguen 100% de cobertura de caminos

Hay 6 caminos posibles en el código

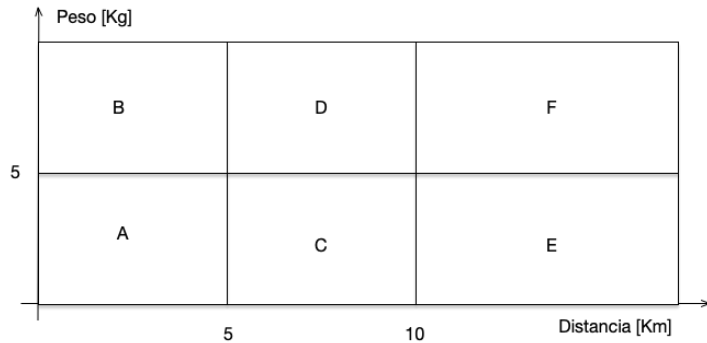
- $\text{distancia} \leq 1$  y  $\text{peso} \leq 5$
- $\text{distancia} \leq 1$ ,  $\text{peso} > 5$
- $1 < \text{distancia} \leq 10$ ,  $\text{peso} \leq 5$
- $1 < \text{distancia} \leq 10$ ,  $\text{peso} > 5$
- $10 < \text{distancia}$ ,  $\text{peso} \leq 5$
- $10 < \text{distancia}$ ,  $\text{peso} > 5$

Los siguientes 6 casos de prueba entregan un 100% de cobertura de caminos

Caso	Distancia	Peso	Esperado
1	0.5	4	2000
2	0.5	6	3000
3	2	4	4000
4	2	6	5000
5	12	4	6000
6	12	6	7000

- c) (10 puntos) Diseña casos de prueba de caja negra que nos entreguen una adecuada confianza de la correctitud de la función de acuerdo a lo que aprendimos en clase

Clases de equivalencia quedan definidas en base al dominio de cada uno de los inputs. La figura ilustra las clases que quedan definidas por los dos inputs



Deberíamos incluir un caso por clase + casos en los bordes de las clases. Para las distintas clases podríamos en este caso usar los mismos 6 casos de la parte b (no siempre es así) y agregar los de los bordes de cada dimensión (7 al 12). Si quisieramos ser mas rigurosos agregaríamos también los casos en que hay bordes en ambas dimensiones (13 al 18)

Caso	Distancia	Peso	Esperado
1	0.5	4	2000
2	0.5	6	3000
3	2	4	4000
4	2	6	5000
5	12	4	6000
6	12	6	7000
7	0	3	2000
8	5	3	4000
9	10	3	4000
10	3	0	4000
11	3	5	4000
12	3	10	5000
13	0	0	2000
14	5	0	4000
15	10	0	4000
16	0	5	2000

17	5	5	4000
18	10	5	4000

Puede observarse que las pruebas de caja negra revelan un comportamiento que tal vez no es el correcto cuando la distancia es 0.

### **Notas de Pauta**

Lo importante aquí es la sistematicidad de las pruebas

El código debe funcionar pero puede suponerse que está validado el input en cuanto a pesos menores a 10kg distancias menores que cero.

Pueden haber tomado decisiones distintas sobre la ambigüedad de los bordes (desde 10 km en adelante lo lógico era interpretarlo como + de 10 porque la regla anterior lo incluía explícitamente)

Deben haber 6 casos en la cobertura y si hay más debe justificarse (no pueden usarse conceptos como casos de borde)

Deben aparecer los casos de borde en las pruebas de caja negra aunque puedan faltar algunas (bordes en ambos lados por ejemplo)

### Pregunta 3 (20 puntos)

Productividad de los primeros dos sprints (la mitad)

3 relatos/sprint

$18/4 = 4.5$  relatos por semana

De acuerdo a ello los 28 puntos de relato que quedan (10 relatos) deberían requerir:

$28/4.5 = 6.2$  semanas

Dado que la velocidad de desarrollo puede aumentar un poco a medida que avanza el proyecto es razonable esperar que se pueda completar en 6 semanas (3 sprints)

a. Si no es posible cambiar la fecha de entrega es necesario negociar el alcance e indicar que en esta versión del producto no estarán implementados todos los relatos.

A esa velocidad en 4 semanas podemos desarrollar

$4 * 4.5 = 18$  puntos de relato

Se desarrollaran en orden de prioridad: 11, 10, 9, 12 y 8

Quedarían fuera del release los relatos 7, 13, 14, 15 y 16

Sprint 3

relato 10 (8) y relato 12 (1)

Sprint 4

relato 11 (5), relato 9 (2), relato 8 (2)

Tambien podríamos optar por no hacer el 10 en el Sprint 3

Sprint 3

relato 11 (5), relato 9 (2), relato 8 (2)

Sprint 4

relato 10 (8) y relato 12 (1)

a. Si la fecha de entrega es negociable podemos mantener el desarrollo de la totalidad del producto pero dejar en claro que se requieren 2 semanas adicionales (quedan 3 y no 2 sprints por desarrollar) por lo que estaríamos terminando el Viernes 13 de Noviembre.

Sprint 3

relato 10 (8) y relato 12 (1)

Sprint 4

relato 11 (5), relato 9 (2), relato 8 (2)

Sprint 5

relato 13 (1), relato 14 (3), relato 7 (3), 15 (1) y 16 (2)

#### Pregunta 4 (18 puntos)

Para cada uno de los aspectos a) es como lo maneja Scrum, b) que tiene de bueno como lo hace Shape Up y c) que tiene de malo.

1.
  - a) Los ciclos de Scrum suelen ser mas cortos (2 o 3 semanas) pero nada impide usar también ciclos de 6 semanas
  - b) El argumento que se da en la descripción. Permite completar algo mas significativo en una iteración
  - c) Un ciclo tan largo hace que se pierda agilidad. Si hay que hacer cambios solo nos daremos cuenta despues de 1 mes y medio. Tampoco es bueno un proceso tan rígido en este sentido (por que no 5 o 7)
2.
  - a) En Scrum no hay una etapa previa antes que lo tome el equipo de desarrollo. El equipo es responsable de las decisiones de diseño
  - b) Un grupo mas especializado puede tomar mejores decisiones. El equipo comienza a ser más productivo de inmediato al tener un trabajo mas definido
  - c) Exige tener equipos de especialistas que hagan el trabajo de "shaping". Puede no tenerse suficientes elementos a mano para tomar decisiones previas
3.
  - a) Scrum no fuerza a *"fixed time, variable scope"* Puede usarse ese esquema (lo más recomendable) pero puede usarse también *"fixed scope, variable time"* agregando iteraciones si es necesario. Si se refiere a que el tiempo de la iteración es fijo ambos modelos son idénticos
  - b) Es bueno que una iteración tenga un tiempo fijo y eso lo tiene tanto Shap Up como Scrum.
  - c) Si Shape Up obliga a fixed time del proyecto completo tiene una falta de flexibilidad para manejar situaciones en que el tiempo sí es negociable
4.
  - a) En Scrum el backlog del producto y de la iteración son piezas importantes puesto que son la base de la planificación
  - b) Lo bueno aparece en la justificación del texto. Fundamentalmente se trata de no perder tiempo gestionando el backlog (poniendo cosas, sacando cosas, priorizando, etc)
  - c) Lo malo es que se puede perder el sentido de la prioridad desde la perspectiva del usuario o del cliente. En Scrum la priorización la hace el product owner
5.
  - a) En Scrum los equipos son multifuncionales pequeños pero de tamaño un poco mas grande (3 a 7)
  - b) Lo bueno es que en proyectos pequeños se pueden tener varios equipos con pocas personas y que los equipos chicos maximizan la productividad
  - c) Un equipo muy pequeño tiene el riesgo de quedar parado si se enferma o se va un integrante del equipo. Es muy difícil tener un equipo multifuncional con dos personas de las cuales hay un solo programador.
6.
  - a) En Scrum las tareas se consideran terminadas o no terminadas. No existen tareas a lo largo de un cerro.
  - b) Lo bueno es que en algunos casos permite mayor precisión en tareas que no están terminadas pero están muy cerca de estarlo y distinguirla de una que solo está comenzando

- c) Es muy riesgoso. Es sabido que los programadores tienden a subestimar el trabajo por delante y por lo tanto se puede terminar pensando que falta muy poco cuando en realidad falta mucho mas.

### **Pregunta 5 (12 puntos)**

Lo más importante de esta pregunta es la identificación de los siguientes archivos y su rol al momento de desplegar la página. Esto es lo mínimo esperado:

- routes.rb - este archivo es donde se configura que la aplicación responde a la url /admin/new\_admin/cars.
- car.rb - (modelo) (puede ser cualquier nombre para el modelo) - para desplegar el listado de autos, es necesario tener al menos un modelo que tenga la información que se desplegará.
- cars\_controller.rb - (puede ser cualquier nombre para el controlador) - el controlador es donde se recibirá el request, cargando los autos a mostrar y delegando el despliegue de la página a una vista
- index.html.erb - (puede ser cualquier nombre para la vista) - la vista es la encargada de desplegar los autos
- cars.css - (puede ser cualquier archivo css) - por último es necesario aplicar estilos para que la página se vea como en el ejemplo

### **Notas de Pauta**

Lo importante aquí es:

- a) Identificar cada uno de los 5 archivos claves (6 puntos por identificar los 5 archivos: ruta, controlador, modelo, vista y css)
- b) Una declaración de para qué están haciendo algo con cada archivo (6 puntos por la descripción del rol de los 5 archivos).

Si dijeran cosas como “genero un scaffold y modifico todos los archivos generados”, eso de por si no tiene ningún valor (ni a favor ni en contra).

Por último, lo ideal sería que su descripción sea específica. Por ejemplo, “la vista muestra los autos” vs “la vista muestra información” o “el controlador carga los autos” vs “el controlador usa la vista y el modelo para entregar una respuesta”