

Pauta Pregunta 1

Total de 24 puntos

1.

a) (4 puntos) Porque los requisitos seguramente cambiarán durante el desarrollo del proyecto, no porque se hubieran entendido mal sino porque el negocio cambió.

b) (4 puntos) Por la naturaleza iterativa y de mejoramiento continuo del proceso. Cada dos semanas se muestra a los interesados el producto y hay oportunidad de cambiar prioridades o ajustar cosas. La reunión llamada *Product Review* es fundamental para que esto suceda.

Nota de pauta: el alumno debe mencionar las características iterativas y el *product review*

c) (4 puntos) Probablemente la mejor opción es el comercial jefe de personal. El es quien mejor entiende la problemática de sueldos de la institución y qué es lo más importante para quienes lo van a utilizar. El ingeniero que ya desarrolló otro sistema de sueldos podría terminar implementando una solución que no funciona bien en este caso.

d) (4 puntos) Porque el esfuerzo de desarrollo y el tiempo de desarrollo no crece en forma lineal con el tamaño del software a construir sino más rápidamente. El uso de una línea recta es válido con tamaños relativamente pequeños.

e) (4 puntos) Las notas pueden contener relatos de usuario, tareas de detalle asociadas a la implementación de un relato u otras cosas menores. Todas ellas son retiradas del tablero al terminar el sprint pues tienen que ganarse el derecho a estar al comenzar la siguiente iteración (podrían no ser prioritarias)

f) (4 puntos) Iterativo implica que el desarrollo se hace en ciclos y en cada ciclo se revisa lo avanzado, los riesgos, etc. Incremental implica además que en cada iteración se agrega un trozo de funcionalidad que es de valor para el cliente. En este caso cada incremento genera un producto potencialmente entregable. Puede haber desarrollo iterativo y no incremental (RUP, espiral) pero el inverso es difícil de imaginar.

Pauta Pregunta 2

Total de 25 puntos

2.

a) (5 pts) Cajero, cliente normal, cliente autoservicio, elaborador, gerente

Nota de Pauta: Es importante distinguir entre el cliente que es atendido por caja del cliente autoatendido porque podría tener relatos distintos

b) (15 pts)

Epica: Ingresar un nuevo pedido por cajero

Relato 1: agregar un nuevo producto a la orden

Yo como cajero necesito agregar un nuevo ítem a la orden para avanzar en la generación del pedido del cliente

Relato 2: eliminar un producto de la orden

Yo como cajero necesito poder eliminar un producto ya agregado a la orden ya sea porque hay que corregir o porque el cliente cambió de opinión

Relato 3: terminar la orden

Yo como cajero necesito cerrar la orden para generar la comanda y la boleta del cliente de modo de recibir el pago correspondiente.

Epica: Ingresar un nuevo pedido por cliente

Relato 4: agregar un nuevo producto a la orden

Yo como cliente de autoservicio necesito agregar un nuevo ítem a la orden para avanzar en la generación de mi pedido

Relato 5: eliminar un producto de la orden

Yo como cliente de autoservicio necesito poder eliminar un producto ya agregado a la orden ya sea porque hay que corregir o porque cambié de opinión

Relato 5: realizar el pago

Yo como cliente de autoservicio necesito hacer el pago de la orden que he ingresado de modo que se genere la comanda asociada y me preparen el pedido

Relato 7: Marcar un pedido como completo

Yo como elaborador necesito indicar que se ha completado la elaboración de una orden determinada para que el producto sea entregado al cliente

Relato 8: Obtener monto de ventas del día

Yo como gerente necesito obtener el total de ventas del día para poder tomar posibles decisiones relacionadas con la gestión del negocio

Relato 9: Obtener cantidades vendidas por producto

Yo como gerente necesito obtener las cantidades vendidas de cada producto para decidir sobre promociones y precios

Relato 10: Obtener el número de ordenes por cajero

Yo como gerente necesito obtener el número de ordenes que cada cajero pudo ingresar para determinar cifras de productividad y calcular bonos y premios

Relato 11: Obtener el número de ordenes de clientes

Yo como gerente necesito conocer cuantas ordenes son colocadas directamente por los clientes para proyectar número de cajeros a futuro

Nota de Pauta: Hay varias cosas a tener en cuenta

Aspectos de Uso de la Técnica

- elección correcta del título (comenzando con un verbo activo)
- estructura de acuerdo al formato: yo como <actor> necesito <que> para <su objetivo>
- concordancia del actor de la descripción con el actor de la parte a)

Número de Relatos, Epicas, etc

No es obligación que identifiquen épicas. Sin embargo un relato como la Epica que ingresa un nuevo pedido es demasiado grande (granularidad incorrecta)

Deben incluir las funcionalidades que permiten ingresar un pedido (tanto por cajero como autoservicio), indicar elaboración completa, procesar pago en el caso de autoservicio y las principales funcionalidades asociadas al gerente.

c) (5 puntos)

Relato 1

Probar agregando un combo 1

Probar agregando un Kids Meal

Probar agregando un Big Mac sin mayonesa

Probar agregando un combo 2 agrandado

Relato 5

Probar con un pago con tarjeta bancaria de prepago

Probar con tarjeta de crédito bancaria

Probar con pago con Mercado de Pago

Probar con pago con Mach

Lo importante aquí es imaginar distintos escenarios del relato de modo de ayudar al implementador en entender mejor los detalles de él.

Pauta Pregunta 3

Total de 26 puntos

3.

a) 10 puntos

Software a desarrollar: 60 puntos de relato para 24 relatos

Equipo: 6 ingenieros

Tiempo de desarrollo prometido: 12 semanas

El equipo se basó en la productividad histórica para hacer una estimación temprana:

36 puntos en 3 sprints => 12 puntos por sprint (de dos semanas)

Si ahora había que desarrollar un producto de 60 puntos...

$60/12 \Rightarrow 5$ sprints, es decir 10 semanas

Es por eso que se comprometió "conservadoramente" para 12 semanas

Pero bueno, no todos los proyectos son iguales y la situación ahora es que conocemos lo que ha sucedido en los 3 primeros sprints: 6, 8 y 4 puntos en cada uno. Es razonable pensar que la productividad que tendremos en los sprints restantes estará cercana al promedio de esos valores:

$(6 + 8 + 4)/3 = 6$ puntos por sprint

Nos quedan por implementar 12 relatos con un total de $60 - 18 = 42$ puntos de relato

De acuerdo a ésto el número de iteraciones necesarias será:

$N = 42/6 = 7$ es decir 14 semanas más y no 6 semanas que es lo que nos queda para cumplir

Nota de Pauta

Es muy importante que la estimación se haga sobre la base de las 3 iteraciones de este proyecto y no sobre los datos del proyecto anterior. Siempre es mucho mas confiable lo que corresponde al mismo proyecto.

b) 4 puntos

Se debe explicar al cliente que el agregar mas ingenieros solo aumentará los costos sin producir ningún cambio importante en la situación. Es decir el total de funcionalidades no estará disponible sino hasta 14 semanas mas

c) 4 puntos

Es posible ajustar por tiempo. Se le indica al cliente que se requerirá extender el deadline en 8 semanas (quedaban 6 al terminar el tercer sprint)

d) 4 puntos

Es posible ajustar por alcance. Se le indica al cliente que se entregará el producto en la fecha indicada pero solo tendrá incorporadas las funcionalidades mas importantes (de mayor valor). Al completar las 12 semanas (6 sprints) tendremos alrededor de 36 puntos de relato implementados del total de 60.

e) 4 puntos

El peor escenario. Seguramente se producirá un ajuste por calidad y a pesar del sacrificio extremo del equipo no se logrará la meta en la fecha comprometida. Sería conveniente volver a intentar reajustar al final de cada sprint restante.

Pauta Pregunta 4

Total de puntos: 25

a) (10 puntos totales) El objetivo de la pregunta es que los alumnos demuestren su conocimiento en ruby, en particular con los conceptos de métodos y parámetros, manejo de string y bloques. El código puede variar entre cada alumno, pero lo importante es que no haya modificado código que no debía modificar, que utilice el bloque dado, que se invoque correctamente desde la función `print_definition`, que haya un correcto manejo de strings y el output sea el adecuado. No es necesario que haya considerado casos límites (0 o 1) o negativos.

- Utilización de bloque e invocación correcta. (3 puntos)
- No modificó código fuera de `#_____`. (2 puntos)
- Cálculo correcto de factorial (método `factorial_of_number`). (2 puntos)
- Manipulación correcta del string. (2 puntos)
- Output correcto. (1 punto)

Posible código solución:

```
# This method receive a number
# and returns the factorial of that number
def factorial_of_number(number)
  result = 1
  for i in 1..number
    result = result * i
  end
  return result
end
```

```
# This method print the definition of factorial
# for a given number n.
# Example:
# 8! = 8 X 7 X 6 X 5 X 4 X 3 X 2 X 1 = 40320
def print_definition
  n = gets.to_i
  r = factorial_of_number(n)
  yield(n, r)
end
```

```
print_definition { |number, result|
  s = "#{number}! = "
  i = number
  while(i > 1) do
    s = s + "#{i} X "
    i -= 1
  end
  print number == 0 || number == 1 ? "#{s}#{result}" : "#{s}1 = #{result}"
```

}

b) (5 puntos totales) El objetivo de esta pregunta es que los alumnos hayan participado en el análisis sobre qué es un ingeniero vs un desarrollador de código. Debe explicar cómo no sólo construye, sino que participa en el proceso previo de encontrar el problema. Lo importante es que esto se vea reflejado en su análisis.

Posible respuesta: Un buen ingeniero no sólo codifica lo que se le pide sino que se encarga de encontrar un problema y darle solución. Encontrar un problema significa que efectivamente descubre la raíz de lo que está ocurriendo, y no sólo escucha lo que el cliente cree que es su problema. Participa activamente de todas las partes del proceso, desde encontrar el verdadero problema, levantamiento de requisitos, diseño de una solución, ejecución o implementación y validación. Se podría decir que muchas veces el ingeniero toma otros roles, como investigador, científico e incluso psicólogo. Un desarrollador, por otro lado, sólo toma un requisito, lo programa y lo entrega, sin participar de todo el proceso.

Se ha hablado en clases sobre buenas prácticas en distintos ámbitos, lo que pueden mencionar es: DRY (Rails), Convention over Configuration, hacer un diseño de la solución antes de implementar, ser consistentes y ordenados en su código, utilizar patrones de diseño, no usar variables globales.

- Distinción entre ingeniero y desarrollador de código. (1 punto)
- Explicación correcta de buena práctica o principio 1. (2 puntos)
- Explicación correcta de buena práctica o principio 2. (2 puntos)

c) (10 puntos totales) Es un ejercicio de modelación de un problema. La solución más simple es hacer una aplicación web que consiste en un formulario para crear posibles invitados. Es importante que mencione qué debe preguntar en el formulario y qué clases necesita para el formulario.

Información mínima necesaria (3.5 puntos):

- Nombre y apellido de la persona. (0.5 puntos)
- Rut o algún identificador único. (0.5 puntos)
- Si se ha vacunado o no, o bien si es alguien que puede vacunarse o no. (0.5 puntos)
- Si sufre de alguna comorbilidad (Puede ser un boolean o un string con la enfermedad que sufre). (0.5 puntos)
- Fecha de nacimiento, o bien edad. (0.5 puntos)
- Si se es lactante o no. Podría hacer un supuesto que sobre cierta edad se deja de ser lactante, pero debe ser un supuesto explícito. (0.5 puntos)
- Fecha de su último viaje al extranjero. Aquí pueden ser más creativos y pensar en fechas de futuros viajes desde que el invitado contesta el formulario hasta la fecha de la fiesta. También pueden guardar un boolean si es que cumplen o no esta condición. (0.5 puntos)

Cómo programarlo: aspectos mínimos que debe explicar (6.5 puntos):

- Debe existir un modelo que represente a un invitado. (1.3 puntos)
- Se debe crear una migración para crear la tabla de invitado junto con sus atributos en la base de datos. (1.3 puntos)

- Debe crear un controlador de invitado con método new y create. Debe explicar qué hace cada método. (1.3 puntos)
- Debe crear una vista con el formulario del método new. (1.3 puntos)
- Debe mencionar que se agregó la ruta de tipo GET para el método new, y la ruta de tipo POST para guardar el formulario. (1.3 puntos)