IIC 2143 Ingeniería de Software
Examen - Semestre 2 /2019
Secciones 01 y 02

Responda cada pregunta en hoja separada
Entregue una hoja con su nombre para cada pregunta aunque sea en blanco
Tiempo: 2:00
Recuerden que están bajo el código de honor

Pregunta 1:

Un banco ofrece a todos sus clientes dos productos bancarios: una cuenta corriente en la cual puede depositar dinero, y una línea de crédito que se utiliza para financiar giros de dinero superiores al monto disponible en la cuenta (sobregiros u "overdraft" en inglés). De esta forma, cuando un cliente intenta realizar un giro superior al monto que tiene disponible, su balance pasa a ser 0 y la diferencia se traspasa a la línea de crédito. Igualmente, al realizar un depósito, este en primer lugar se utiliza para pagar la línea de crédito y posteriormente pasa a aumentar el balance.

A continuación, se ofrece la definición de la clase Account utilizada por el sistema computacional de este banco que contiene esta lógica.

```
class Account
  attr_reader :amount, :overdraft

  def initialize
    @amount = 0
    @overdraft = 0
  end

  def deposit(dollars)
    if @overdraft > 0
      if dollars <= @overdraft
        @overdraft -= dollars
      else
        @overdraft = 0
        @amount += dollars - @overdraft
      end
    else
      @amount += dollars
    end
  end

  def withdraw(dollars)
    if dollars <= @amount
      @amount -= dollars
    else
      @amount = 0
      @overdraft += dollars - @amount
    end
  end
end
```

En la versión 2.0 de este Sistema computacional, se desea ofrecer un mecanismo que le permita a los clientes ser notificados cuando entran en sobregiro. Decimos que un cliente entra en sobregiro cuando, no teniendo ningún monto cargado en su línea de crédito, comienza a usarla luego de girar una cantidad de dinero superior a su balance. Se ofrece una definición de la clase Client a continuación:

```
class Client
  def overdraft_warning
    puts("Overdraft Warning!")
  end
end
```

Lamentablemente, por temas de retrocompatibilidad, se le impone la restricción de no modificar de ninguna forma la clase Account definida anteriormente.

a) Utilice el patrón Decorator para añadir a la clase Account la capacidad de notificar a clientes cuando entren en sobregiro.

(3.5 puntos)

b) Demuestre cómo operaría su implementación en la práctica con un breve script en donde producto de su ejecución se gatille una advertencia de sobregiro.

(0.5 puntos)

c) Ilustre el funcionamiento de su script con un diagrama de secuencia.

(2 puntos)

Pregunta 2:

A continuación, el código Ruby correspondiente a un procedimiento que calcula las raíces de una ecuación de segundo grado:

```ruby
def cuadratica(a, b, c)
  # encuentra las raices de una ecuacion del tipo ax2 + bx + c = 0
  if (a == 0)
    return 'No es una ecuacion cuadratica'
  else
    d = b*b - 4*a*c
    if (d == 0)
      return 'x='+(-b/2/a).to_s
    else
      if (d > 0)
        return 'x1='+((-B-Math.sqrt(d))/2/a).to_s +
            ' x2='+((-B+Math.sqrt(d))/2/a).to_s
      else
        return 'x1=('+(- b/2/a).to_s+','+(Math.sqrt(-d)/2/a).to_s+')' +
            ' x2=('+(-b/2/a).to_s+','+(-Math.sqrt(-d)/2/a).to_s+')'
      end
    end
  end
end
```

Se pide diseñar casos de prueba (justifique) para este código considerando:

a) Estrategia de "black box", clases de equivalencia y bordes

(3 puntos)

b) Estrategia de "white box" y criterio de cobertura de caminos

(3 puntos)

Para cada uno de sus tests, es suficiente con que especifique los valores de input; no es necesario que especifique el output esperado, aunque si lo desea, puede precisar la clase de equivalencia, borde o camino al cual corresponde su test.

Pregunta 3:

Vamos a suponer que Ud se ve enfrentado a diseñar una aplicación EatFine de pedidos de comida a restaurants desde cero con las siguientes características:

- La aplicación debe estar disponible en smartphones, tablets y laptops.
- Los usuarios de la aplicación pueden ver platos ofrecidos por diversos restaurants y ordenarlos desde sus casas para entrega a domicilio.
- Los restaurants que quieren enrolarse no se les cobra nada de base, pero deben proveer una API para acceder a su oferta e interactuar con la aplicación.
- Los usuarios pueden autenticarse con un servicio externo.
- Los usuarios pueden pagar solo con WebPay o PayPal.
- La aplicación utiliza los servicios de Google Maps para ayudar a buscar restaurants en un área.
- Los repartidores (motoristas) deben recoger la comida y llevarla a su destino (la aplicación debe ayudar en esto también).
- Se les envía una vez al mes a cada restaurant adherido el dinero recaudado de la venta de sus platos menos un descuento de 5% por servicio.
- Los restaurantes adheridos pueden monitorear el detalle de sus ventas a través de la aplicación para revisar el detalle de las platas.

Se pide hacer un primer borrador de la arquitectura de este sistema en base a una arquitectura de microservicios y expresarlo en la forma del modelo C4 (debe detallar el de Contexto y el de Containers pero no es necesario hacerlo con el de componentes).

Suplemento I1:

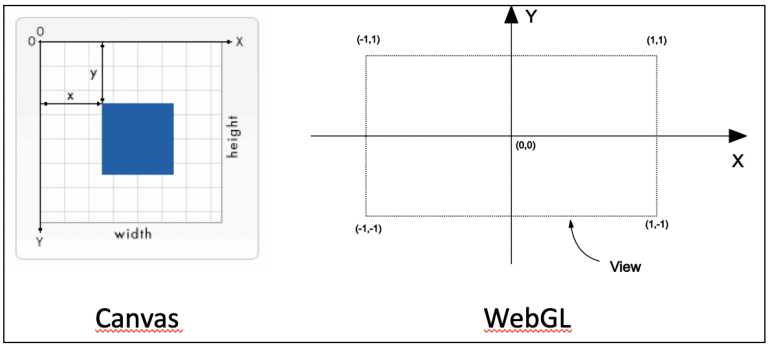
La empresa Initech se encuentra realizando un post-mortem de un proyecto de software fallido. Con el objetivo de entender qué salió mal, se le contrata a usted como consultor para analizar el proceso con que se llevó a cabo el proyecto. El gerente encargado de este comparte con usted el siguiente relato:

"En Initech nos enorgullecemos de seguir las mejores prácticas de desarrollo y de estar siempre a la vanguardia de los procesos modernos de manejo de proyectos, por lo que decidimos adoptar Scrum desde el primer día. Al inicio del proyecto, el cliente nos entregó un documento con los requisitos funcionales del proyecto y procedimos a plasmarlos en relatos de usuario. En base a ellos, construimos una carta Gantt para tener una aproximación inicial de cómo ir avanzando en el proyecto. Dividimos nuestro equipo de trabajo en 5 grupos, cada uno a cargo de un módulo distinto de la aplicación. Cada grupo consistía en 5 desarrolladores y 1 supervisor. Toda comunicación entre equipos debía hacerse a través de los supervisores para optimizar la toma de decisiones. Todos los días, realizamos un stand-up meeting entre yo y los supervisores para ir monitoreando avances. Estas reuniones se limitarían a resumir los sucesos de interés de la jornada anterior y no solían durar más de una hora. Siguiendo las buenas prácticas de Scrum, dividimos el proyecto en Sprints cortos de entre una y tres semanas de duración según la tarea planificada para el momento. Las tareas por realizar en cada Sprint se desprendieron de la carta Gantt inicial. Al finalizar cada Sprint, se le enviaba al cliente un reporte exhaustivo de todos los avances del proyecto en un documento oficial. La retroalimentación del cliente era analizada por gerencia quien estaría a cargo de destilarla en requisitos adicionales. De ser aprobados, estos requisitos serían derivados a los respectivos equipos de desarrollo para incorporarlos a la lista de tareas planificadas para el Sprint actual de manera a no generar atrasos en la planificación."

En base a sus conocimientos de ingeniería de software, ¿qué estima que se hizo mal?

Suplemento I2:

Canvas y WebGL son dos tecnologías web disponibles en HTML5 para dibujar en pantalla. Ambas tecnologías permiten dibujar gráficos 2D, aunque Canvas suele preferirse para este propósito debido a la mayor simpleza de su SDK. Sin embargo, una diferencia fundamental entre ambos APIs es que operan con distintos sistemas de coordenadas (ver figura adjunta).



Una aplicación de procesamiento de imágenes ofrece dos mecanismos para dibujar Bitmaps en pantalla. En primer lugar, puede hacer uso de las clases HtmlCanvasRenderer y HtmlBitmap para dibujar Bitmaps utilizando el API Canvas de HTML5. Un extracto de estas clases se ofrece a continuación:

```
class HtmlCanvasRenderer
  def draw(bitmap, x, y)
    # Iterates over all pixels by calling the 'pixel' method and draws them
    # on screen starting from screen coordinates (x, y)
  end
end

class HtmlBitmap
  attr_reader :height, :width

  # Returns the RGB components of the pixel at coordinates (x, y)
  # as an array with three elements: [red, green, blue]
  def pixel(x, y)
    # ...
  end
end
```

Por otro lado, esta misma aplicación también permite dibujar Bitmaps utilizando el API de WebGL utilizando las clases WebGLRenderer y WebGLBitmap definidas a continuación:

```
class WebGLRenderer
  def draw(bitmap, x, y)
    # Iterates over all RGB components of the bitmap by calling the 'red', 'green' and
    # 'blue' methods respectively and draws them on screen starting from
    # screen coordinates (x, y)
  end
end
```

```ruby
class WebGLBitmap
  attr_reader :height, :width

  # Returns the red component of the pixel at coordinates (x, y)
  def red(x, y)
    # ...
  end

  # Returns the green component of the pixel at coordinates (x, y)
  def green(x, y)
    # ...
  end

  # Returns the blue component of the pixel at coordinates (x, y)
  def blue(x, y)
    # ...
  end
end
```

Note que el método *draw* de la clase HtmlCanvasRenderer espera recibir instancias de HtmlBitmap; mientras que el método *draw* de la clase WebGLRenderer espera recibir instancias de la clase WebGLBitmap.

a) Cree un adaptador que le permita a HtmlCanvasRenderer operar con instancias de WebGLBitmap.

(2 puntos)

b) Dibuje el diagrama UML de clases de su implementación.

(2 puntos)

c) Cree un adaptador que le permita a WebGLRenderer operar con instancias de HtmlBitmap.

(2 puntos)

Pauta Pregunta 1:

Inciso a)

```ruby
class ObservableAccountDecorator
  attr_reader :decorator

  def initialize(decorator)
    @decorator = decorator
    @listeners = []
  end

  def deposit(dollars)
    @decorator.deposit(dollars)
  end

  def withdraw(dollars)
    if @decorator.overdraft == 0 && @decorator.amount < dollars
      @listeners.each do |listener|
        listener.overdraft_warning
      end
    end
    @decorator.withdraw(dollars)
  end

  def amount
    @decorator.amount
  end

  def overdraft
    @decorator.overdraft
  end

  def add_listener(listener)
    @listeners << listener
  end
end
```

Nota de Pauta:

Desglose de puntaje:

- Correcta implementación de patrón decorator, incluyendo re-definición de todos los métodos de clase Account con delegación a decorator. Se debe apreciar que el alumno entienda que el atributo sostenido por un decorator no necesariamente es el Account, sino también puede ser una instancia de otro decorator. Se puede también agregar una superclase al decorator para contener implementaciones comunes. **1.5 puntos.**
- Correcta implementación de patrón observer dentro de decorator. **1 punto.**
- Correcta lógica de notificación de sobregiro cuando corresponde. **1 punto.**

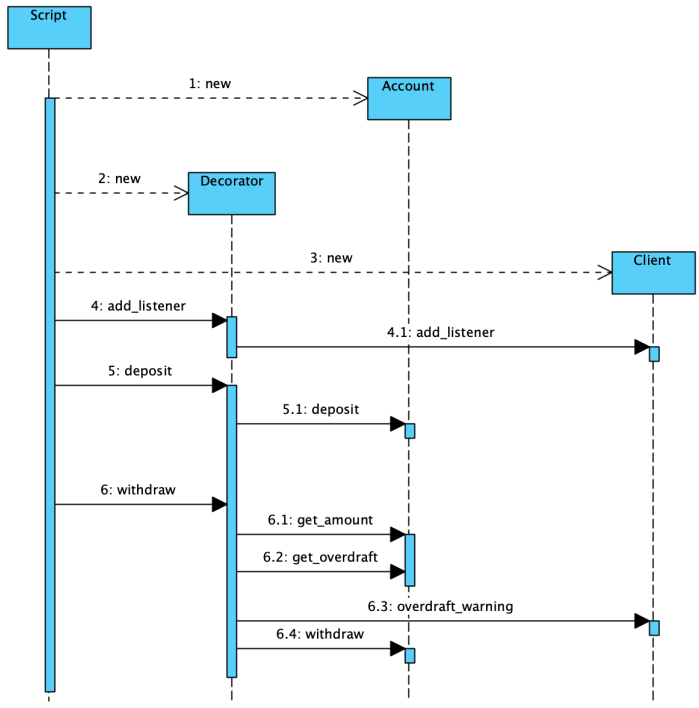
Inciso b)

```ruby
account = Account.new
observable = ObservableAccountDecorator.new(account)

client = Client.new
observable.add_listener(client)

observable.deposit(1000)
observable.withdraw(2000)
```

Nota de Pauta:

La respuesta del alumno en esta sección depende de su implementación, pero no debería ser muy distinto de esto.

Inciso c)



Nota de Pauta:

No es necesario crear un lifeline para Script ni explicitar la creación de los objetos. No obstante, deben figurar lifelines para Decorator, Account y Client. Los mensajes que figuren en la respuesta del alumno deben corresponder al código propuesto en a) y b).

Pauta Pregunta 2:

Inciso a)

Podríamos partir dividiendo el espacio de input según el tipo de resultado
input que produzca dos raíces iguales (x - 2) * (x-2) = 0
input que produzca dos raíces reales (x -2)* (x - 3) = 0
input que produzca raíces complejas x2 + x + 10 = 0
input que corresponde a una ecuación de primer grado x + 5 = 0
Podríamos en cada caso probar con coeficientes positivos o negativos

Posibles casos de prueba para los 3 de arriba

i. a = 1, b = - 4, c = 4
ii. a = 1, b = -5, c = 6
iii. a = 1, b = 1, c = 10
iv. a = 0, b = 1, c = 5

Si agregamos coeficientes de ambos signos podríamos agregar

v. a = 1, b = 4, c = 4
vi. a = 1, b = 10, c = -1
vii. a = 1, b = -1, c = 10
viii. a = 0, b = -2, c = -5

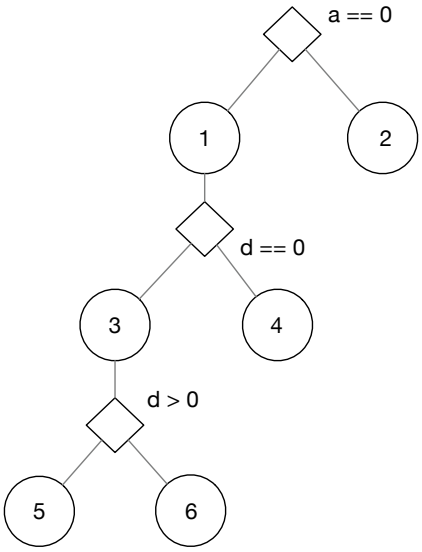
Nota de Pauta:

La respuesta del alumno deber incluir las clases de equivalencia que consideró, los bordes que consideró, y la correspondencia entre cada uno de sus tests propuestos y estas clases de equivalencia y bordes.

Verificar que las clases de equivalencia propuestas por el alumno constituyan una partición (deben ser disjuntas, y su unión debe formar todo el espacio posible de inputs). Nótese que en esta pauta se toma el supuesto que no hay bordes (se considera un supuesto válido). También se admite considerar que a=0 es un borde, en ese caso habría que considerar un test para a=0, y opcionalmente agregar un test para “a levemente inferior a 0” y “a levemente superior a 0”.

Inciso b)

El grafo asociado a ese código es el siguiente



Hay 4 posibles caminos, la cobertura de caminos exige casos de prueba para

i. 1 - 3 - 5
ii. 1 - 3 - 6
iii. 1 - 4
iv. 2

Un caso de prueba para esos caminos requiere entonces

i. a != 0, d != 0, d ≤ 0
ii. a != 0, d != 0, d > 0
iii. a != 0, d == 0
iv. a == 0

Dado que $d = b^2 - 4ac$
$d = 0 \Rightarrow b^2 = 4ac$
$d > 0 \Rightarrow b^2 \geq 4ac$
$d < 0 \Rightarrow b^2 < 4ac$

Por lo tanto posibles casos de prueba para cubrir los 4 caminos serían

i. a = 1, b = 2, c = 3
ii. a = 1, b = 4, c = 2
iii. a = 1, b = 2, c = 1
iv. a = 0, b = 1, c = 1

Nota de Pauta:

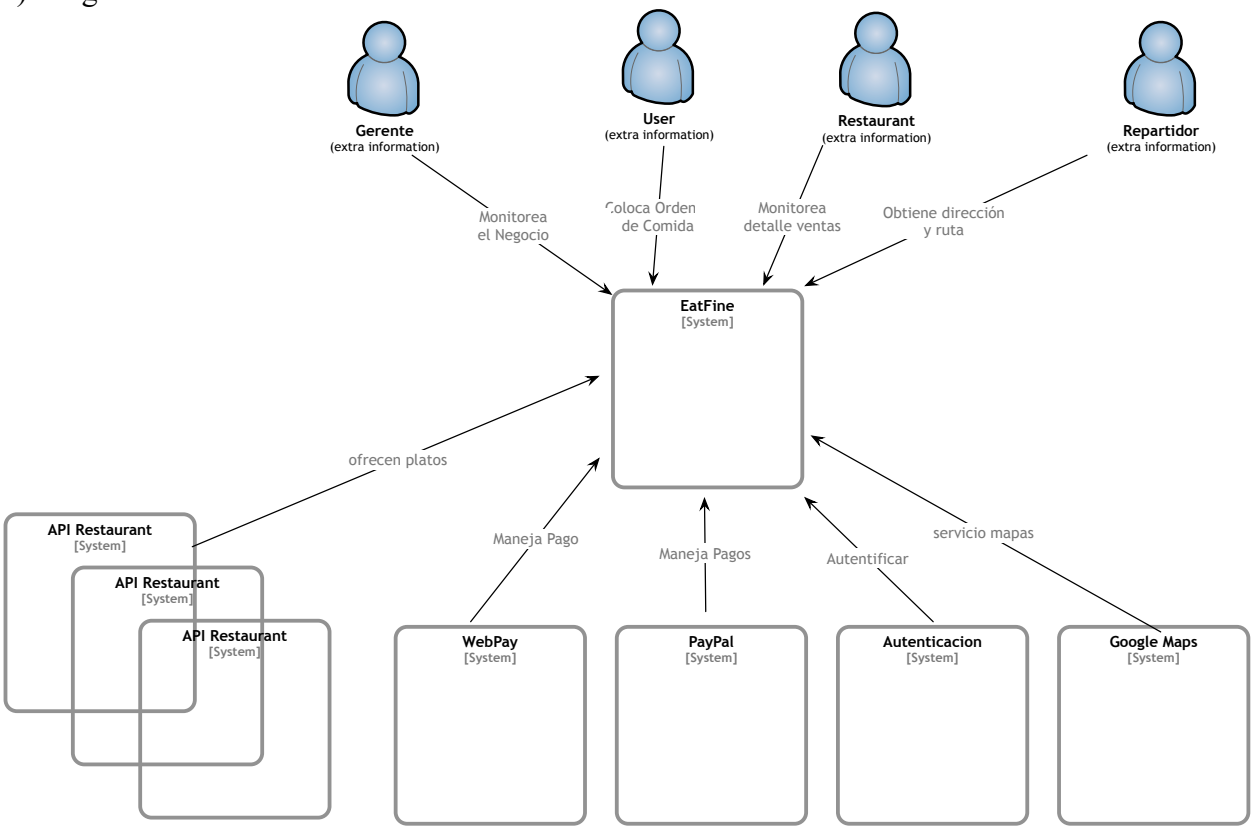
En la respuesta del alumno debe claramente figurar a qué camino corresponde cada test propuesto.

Pauta Pregunta 3:

Vamos a suponer que Ud se ve enfrentado a diseñar una aplicación EatFine de pedidos de comida a restaurants desde cero con las siguientes características
- la app debe estar disponible en smartphones, tablets y laptops
- los usuarios de la app pueden ver platos ofrecidos por diversos restaurants y ordenarlos desde sus casas para entrega a domicilio
- los restaurants que quieren enrolarse no se les cobra nada de base pero deben proveer una API para acceder a su oferta e interactuar con la aplicación
- los usuarios pueden autenticarse con un servicio externo
- los usuarios pueden pagar solo con WebPay o PayPal
- la app utiliza los servicios de Google Maps para ayudar a buscar restaurants en un área
- los repartidores (motoristas) deben recoger la comida y llevarla a su destino (la app debe ayudar en esto tambien)
- se les envía una vez al mes a cada restaurant adherido el dinero recaudado de la venta de sus platos menos un descuento de 5% por servicio
- los restaurantes adheridos pueden monitorear el detalle de sus ventas a través de la app para revisar el detalle de las platas

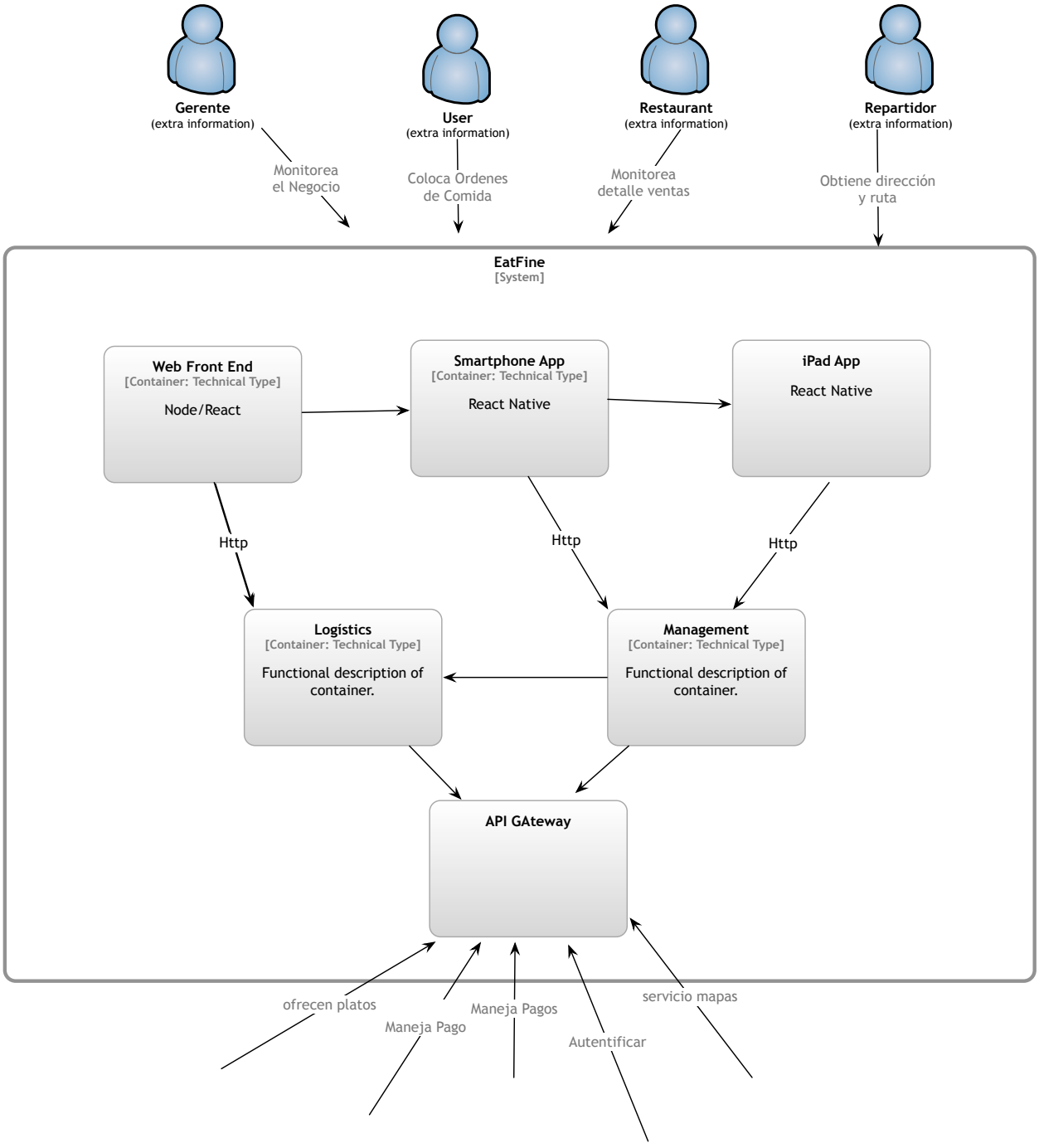
Se pide hacer un primer borrador de la arquitectura de este sistema en base a una arquitectura de microservicios y expresarlo en la forma del modelo C4 (Contexto, Containers y Componentes)

a) Diagrama de Contexto



A nivel de containers hay múltiples posibles arquitecturas pero una posibilidad es

1) Front End Web
2) Front End Smartphone
3) Front End Tablet
4) Back End Logistico (órdenes, platos, clientes, repartidores)
5) Back End Gestión (cobros a clientes, cobros a restaurants, apoyo a la gestión)
6) API Gateway



Es importante que se muestre consistencia con diagrama de contexto anterior

A nivel de componentes llegamos a los microservicios e indudablemente sería muy largo de detallar pero a modo de ejemplo

En el contenedor de la logística por ejemplo podríamos pensar en los siguientes microservicios

obtener platos (por tipo, por precio, etc)
armar ordenes
generar rutas

....

En el de gestión podríamos tener

cobrar al cliente
emitir factura
pagar a restaurant
resumen de ventas
estadísticas de ventas

Pauta Suplemento I1:

Se pueden señalar los siguientes inconvenientes en el relato del gerente:

- Los relatos de usuario deben ser generados por el cliente, no deben ser generados por el equipo de desarrollo en base al análisis de un documento de requisitos.
- El forzar la comunicación a través un canal oficial particular (los supervisores) e impedir que los desarrolladores de distintos equipos se comunicaran directamente dificulta la coordinación entre equipos.
- Los stand-up meetings deben realizarse entre todos los miembros del equipo y no durar más de 15-20 minutos. Una hora es excesivo y sugiere que se estaban haciendo tareas administrativas adicionales que no corresponden.
- La duración de los Sprints en Scrum es fija, no debe ser variable.
- El basar el sprint backlog en una carta Gantt impide adaptarse a errores en la planificación, retrasos y cambios en el proyecto.
- No se aprecia una sesión formal de Sprint Review al final del Sprint. El enviar un documento escrito de avances no es un sustituto a una reunión presencial con demos.
- La retroalimentación debe ser analizada por los mismos desarrolladores y en procesos ágiles estos mismos deben estar empoderados para tomar las decisiones que estimen más convenientes. La estructura organizaciones vertical que se desprende del relato es ajena a las buenas prácticas promovidas por los procesos ágiles de desarrollo.
- El incorporar un requisito nuevo a un Sprint debe necesariamente estar acompañado de una replanificación para no agobiar al equipo de trabajo.

Nota de Pauta:

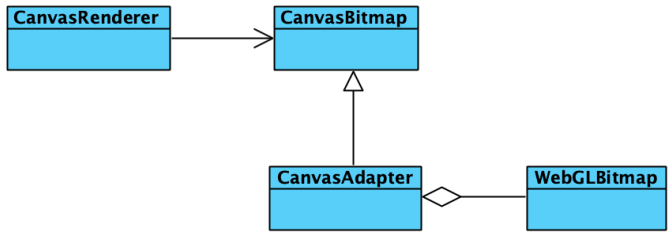
Cada una de las ideas expresadas en esta pauta debe figurar de alguna forma en la respuesta del alumno. Asignar 0.75 puntos por cada una.

Pauta Suplemento I2:

Inciso a)

```ruby
class CanvasAdapter
  def initialize(webgl_bitmap)
    @webgl_bitmap = webgl_bitmap
  end

  def pixel(x, y)
    red = @webgl_bitmap.red(x, @webgl_bitmap.height - y)
    green = @webgl_bitmap.green(x, @webgl_bitmap.height - y)
    blue = @webgl_bitmap.blue(x, @webgl_bitmap.height - y)
    [red, green, blue]
  end
end
```

Inciso b)



Nota de Pauta:

Basta con incluir las clases y sus relaciones en el diagrama UML para tener todo el puntaje. El alumno puede adicionalmente especificar los atributos y/o métodos de cada clase.

Inciso c)

```ruby
class WebGLAdapter
  def initialize(canvas_bitmap)
    @canvas_bitmap = canvas_bitmap
  end

  def red(x, y)
    @canvas_bitmap.pixel(x, @canvas_bitmap.height - y)[0]
  end

  def green(x, y)
    @canvas_bitmap.pixel(x, @canvas_bitmap.height - y)[1]
  end

  def blue(x, y)
    @canvas_bitmap.pixel(x, @canvas_bitmap.height - y)[2]
  end
end
```