

1)

11/16

salarios

JORGE DE GOYENECHE

SECCION 1

20624719

33

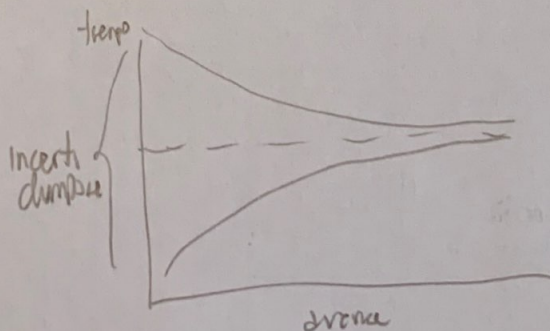
A) La sobreestimación tiene un costo menor, ya que lo único que ocurre es generar una pérdida para el cliente. Un problema importante es conocido como Ley de Parkinson, donde todo el tiempo será utilizado, es que el equipo tendrá mayor agotamiento. La subestimación genera graves problemas con la calidad del producto, porque se tiene menos tiempo para el debuggeo, diseño y testing.

✓ (1) **Fatón Problemas de subestimación.**

B) No es razonable, porque aunque la matemática sea simple y lógica a primera vista, al agregar código se aumenta la complejidad de manera exponencial, no lineal. Por lo que el esfuerzo requerido será mayor al disponible y se tendrá que hacer un ajuste por alcance o tiempo frente a la estimación original, la cual es lineal.

C) El cono de la incertidumbre es el rango que posee la estimación de un proyecto a medida de que este avanza. Forma un cono porque en un principio es muy difícil estimar el tiempo pero a medida de que se avanza y se tiene experiencia en el desarrollo actual la incertidumbre disminuye.

✓ (2)





D) Un ejemplo es la modelación de software y bases de datos del sistema de salud. Donde cada una no se logra comunicar con las demás en un sistema unificado porque cada una requiere distintos parámetros y configuraciones. Es un sistema no acoplado pero con alta cohesión y se debe tomar pasos extra para comunicarse. X

E) Porque una alta cohesión significa buena fluidez y comunicación entre objetos lo que agiliza y simplifica el mantenimiento y el bajo acoplamiento permite editar y extender el diseño de manera independiente. ✓ ①

F) Tier hace referencia a un nivel especializado independiente y layer a una capa de funcionamiento. Se relacionan ya que un Tier puede ser considerado una Layer en el modelo Multi-Tier arquitectónico. Y que si bien es independiente a otros tiers jerárquicos, aún es una capa en el funcionamiento de la arquitectura. ✓ ①

G) Ventajas Micro

- Facilidad de conexión
- Independencia por servicio
- Variedad de información

Desventajas Micro ✓ ②

- Menor cohesión
- Puede provocar lentitud en obtención de datos
- Costoso de mantener API

H) SOA

- Conexión mediante Service Bus
- Bloques Estructurados

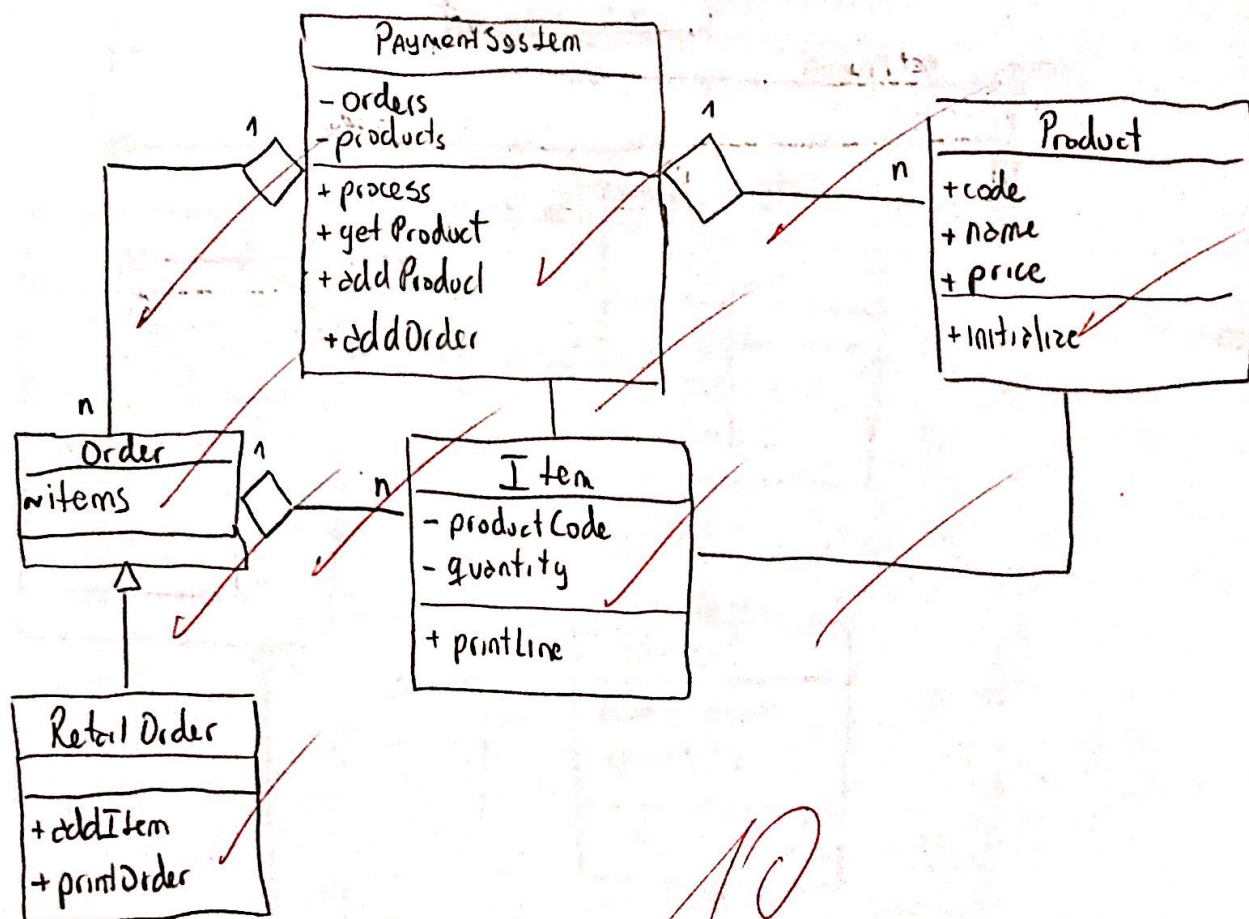
Microservicios ✓ ②

- Conexión mediante API
- Piezas que se estructuran en la interfaz

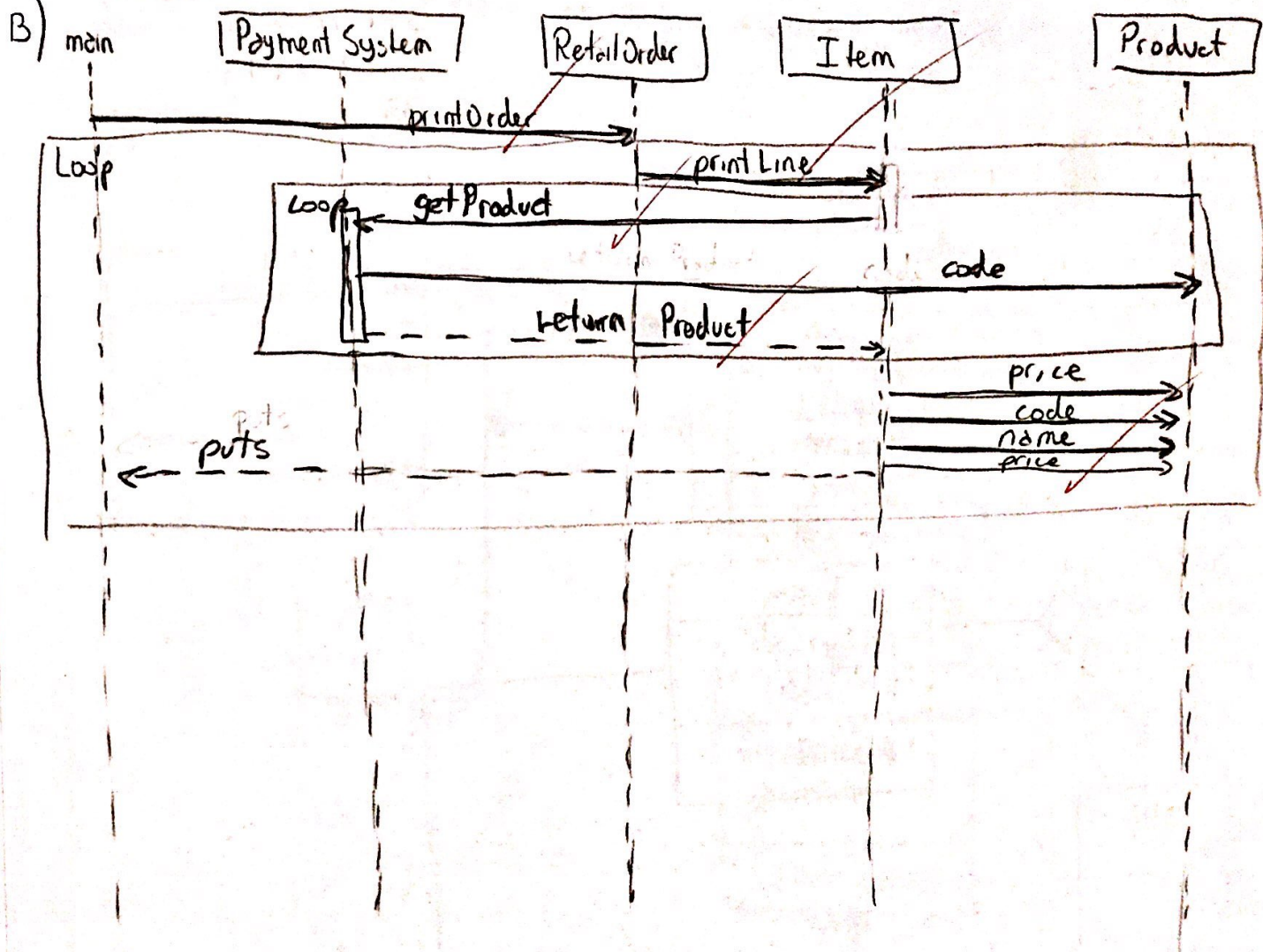
2)  
A)

20/20

JORGE DE BOYENECHE  
SECCION 1  
20624719  
33







10

3)

A)

```
class Game (+)
  def initialize (numberOfPlayers, strategy)
    @deck = CardDeck.new
    @players = []
    numberOfPlayers.times { |id| @players.push (Player.new(id)) }
    @strategy = strategy (+)
  end
  def printState() } Igual
end
  def split()
    strategy.split (@players, @deck) } Modificado
end
end (+)
```

agrega

def strategy = (strategy)  
@strategy = strategy  
end

class Block

```
def split (players, deck)
  players.each do |player|
    5.times { player.add (deck.pickLast) }
  end
end
```

end

class OneByOne

```
def split (players, deck)
  5.times {
    players.each do |player|
      player.add (deck.pickLast)
    end
  }
end
```

end

end

class One

```
def split (players, deck)
  players.sample.add (deck.pickRandom)
end
```

end



B)

```
game = Game.new(3, Black.new)
```

```
game.split()
```

```
game.strategy = OneByOne.new
```

✓ + 8

```
game.split()
```

```
game.strategy = One.new
```

```
game.split()
```

```
----- end
```



4)

JORGE DE BOYENECHE

SECCION 1

20624719

33

class Post

def initialize(msg, auth)

@message = msg

@author = auth

@likes = 0

@comments = [] → @tag = 'simple'

end

def like()

@likes = @likes + 1

end

def print

print Category

end

def printCategory

if @tag == 'Simple' puts "-- Simple Post --"

print Author

end

def print Author

puts "# #{@author} -- # #{@message}"

print Details

end

def print Details

'Abstract'

print LikesComments

end

def print LikesComments

puts "# #{@likes.to\_s} likes"

if @comments.length &gt; 0

puts "comments:"

@comments.each do |each|

puts " # #{each}"

end

end

end

end

Muy bien :)

20/20 ptos.

```

class ImagePost < Post
  def initialize (msg, auth, filePath)
    super (msg, auth)
    @filePath = filePath
    @tag = 'Image'
  end
  def printCategory
    puts "-- Image Post --"
    super
  end
  def printDetails
    puts "Photo: #{@filePath}"
    super
  end
end

```

```

class CheckInPost < Post
  def initialize (msg, auth, placeName)
    super (msg, auth)
    @placeName = placeName
    @tag = 'Check In'
  end
  def printCategory
    puts "-- Check In Post --"
    super
  end
  def printDetails
    puts "Checking at #{@placeName}"
    super
  end
end

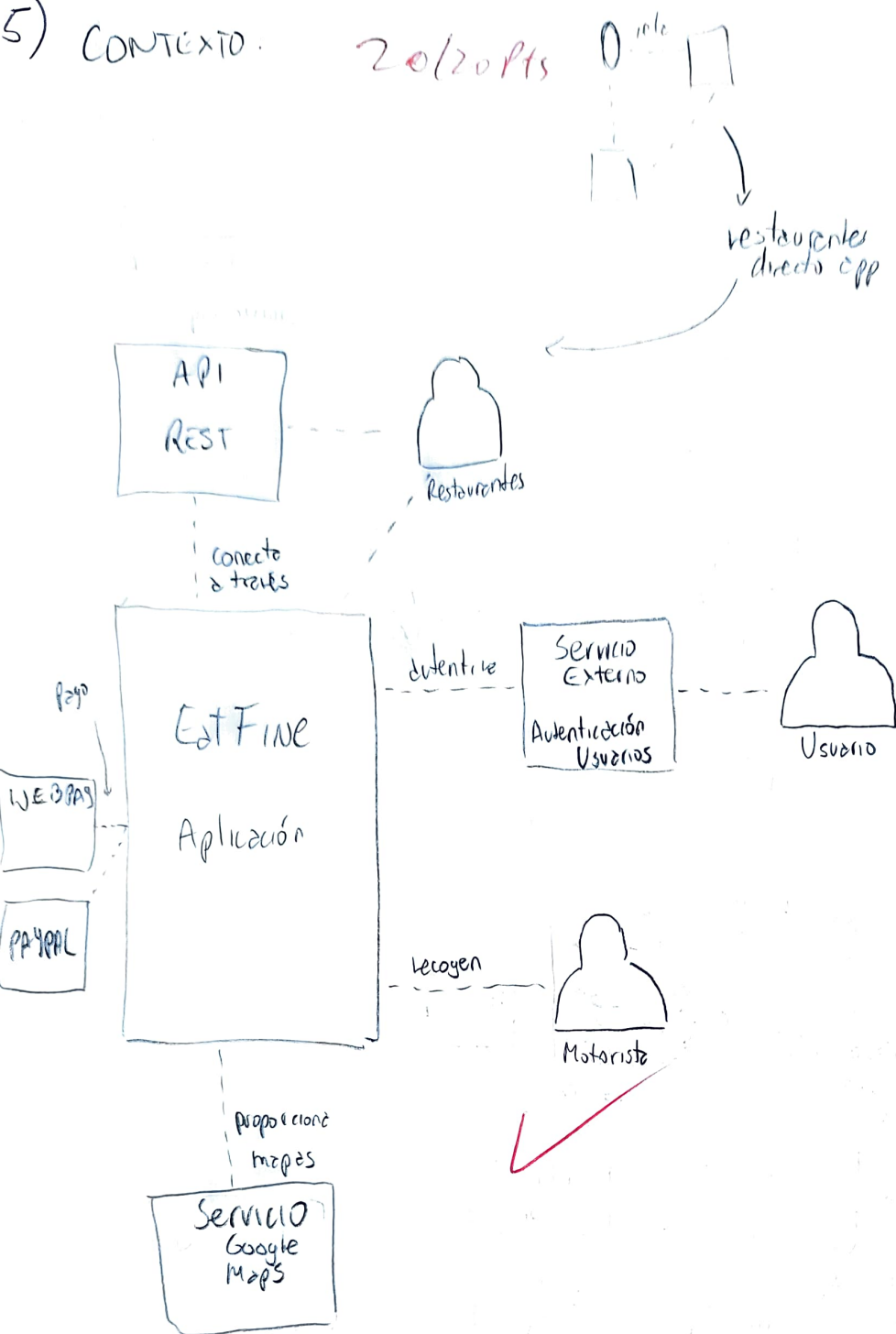
```



5) CONTEXTO.

20/20 Pts

JORGE DE ESCOBAR  
SECCION 1  
20624719  
33



# CONTAINERS

