

Diseñando API's de Micro-Servicios Para Mobile.



Capítulo 1 Temporada 2
Track: Java, Apps
Móviles y su nube

Jorge Vargas

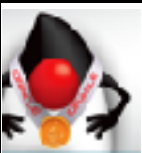
CTO **weex**

SUN/Oracle Java Champion

@edivargas

Jorge Vargas García
SUN/Oracle Java Champion
@Edivargas
@JavaUP

- ✓ CTO en weex (Operador Virtual Móvil en México)
- ✓ Arquitecto para proyecto Calmecac
- ✓ Instructor Java en CertificaTIC
- ✓ Consultor Freelancer en Java y tecnologías asociadas desde 2000.
- ✓ Miembro activo en las comunidades de Java en México y Latinoamérica.
- ✓ Promueve y usa el software libre, tanto en sus actividades de desarrollo y consultoría como las implementa con sus clientes.
- ✓ Cree en la agilidad y su importancia en el desarrollo de software.
- ✓ Orgulloso graduado del IPN.

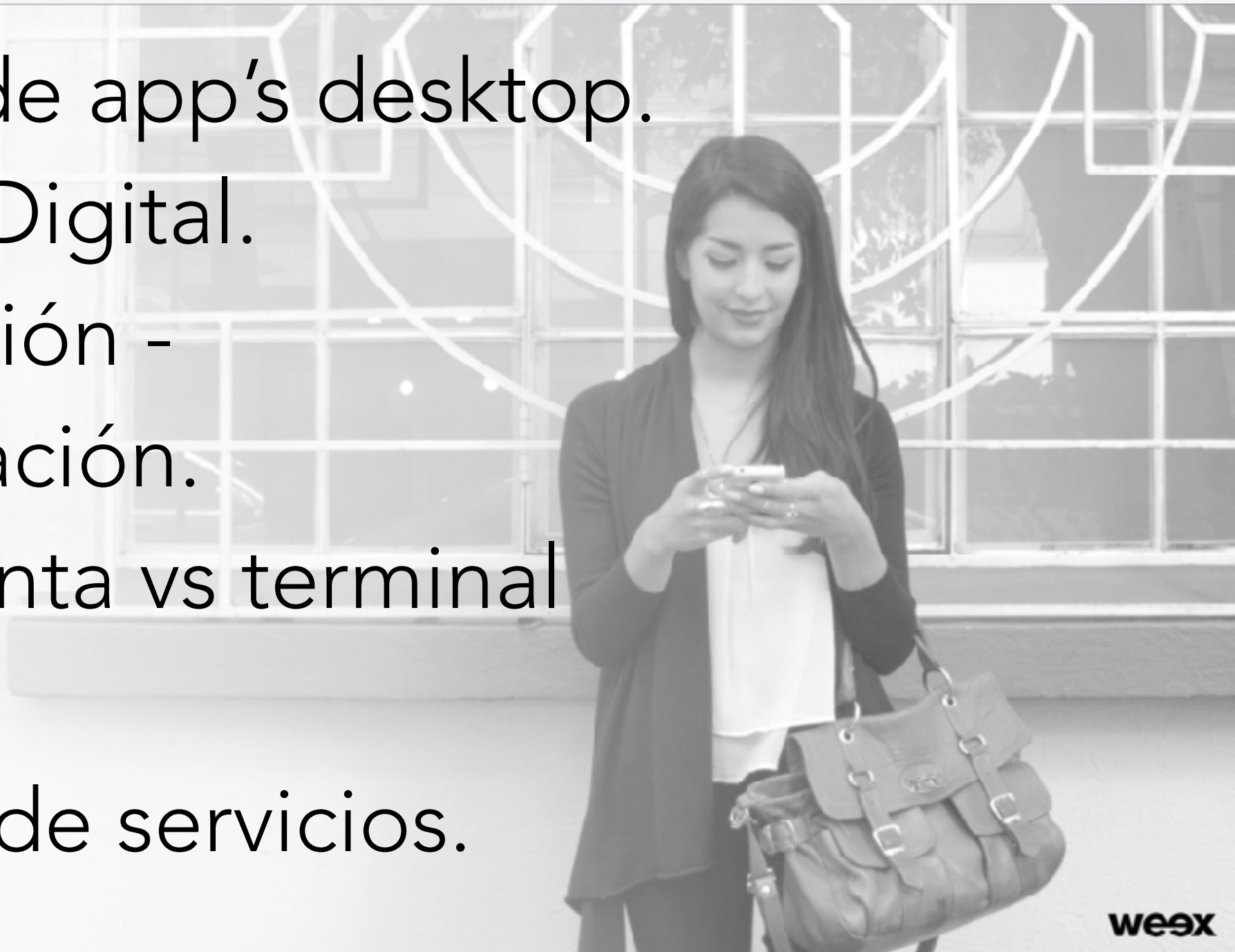


weex



Aplicaciones Móviles

- Extensión de app's desktop.
- Marketing Digital.
- Segmentación - Geolocalización.
- Terminal tonta vs terminal inteligente.
- Un mundo de servicios.



we9x

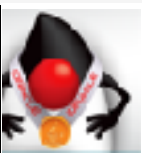
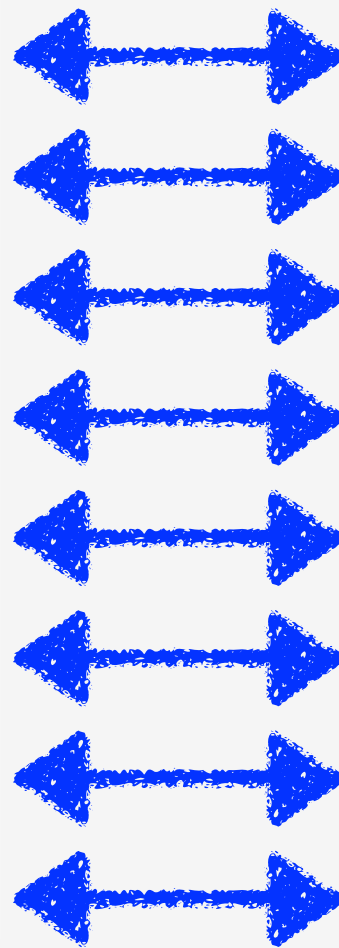


we9x



El Front-End y el Back-End

Llamados a servicios



Opciones de Desarrollo - Front

- Nativas (Objective-C, Swift, Java, Kotlin).
- Framework cross-platform (PhoneGap, Xamarin, Ionic, Titanium, SenchaTouch, etc).
- HTML 5 responsive (Bootstrap, Kendo, etc).



we9x

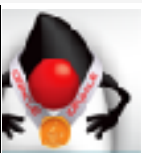


we9x



Opciones de Desarrollo - Back

- Java.
- Java.
- Existen otras???.
Claro pero
recuerden que ...



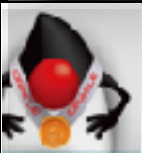
La Nube ... !Qué Confusión!!!

PaaS

SaaS

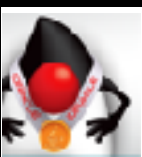
IaaS

weex



Servicios o Microservicios

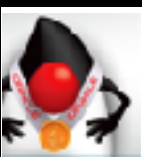
- El otro dia me dijeron SOA o Microservicios!!! ¿Están bromeando verdad?
- ¿Qué son?
- ¿Los necesito?
- ¿Incrementa costo o mantenimiento?
- ¿Facilita el deployment?
- Les cuento una vieja historia?



Servicios vs. Microservicios

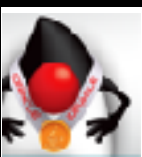
- Ligado a conceptos Legacy.
- No son mecanismos monolíticos.
- Siempre han existido, pero no se identifican fácilmente.
- Puede escalarse y monitorizarse de forma muy fácil.

- Principio de responsabilidad única.
- Multi-tecnología (Legacy).
- Ámbito de control en desarrollo.
- Escalado horizontal más fácil.
- Nó, no facilita el deployment.



Retos de los Microservicios

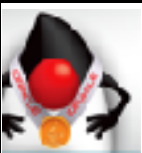
- Localización (HA).
- Tolerancia a fallos (Resilencia).
- Gestión de la configuración (HA).
- Gestión de Logs (Monitoreo).
- Gestión de Despliegue (Monitoreo).
- Seguridad (HA).



Algo Importantísimo!!!

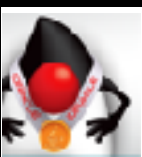
- Tú, si Tú !!!
- NO ERES NETFLIX

NETFLIX | **OSS**



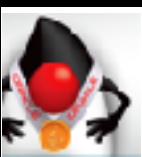
API - Definición

- La interfaz de programación de aplicaciones, abreviada como API del inglés: Application Programming Interface. Es un conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.
- Es el contrato a cumplir por ambas partes: Servidor vs Cliente de consumo.

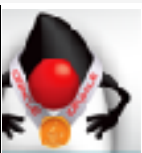
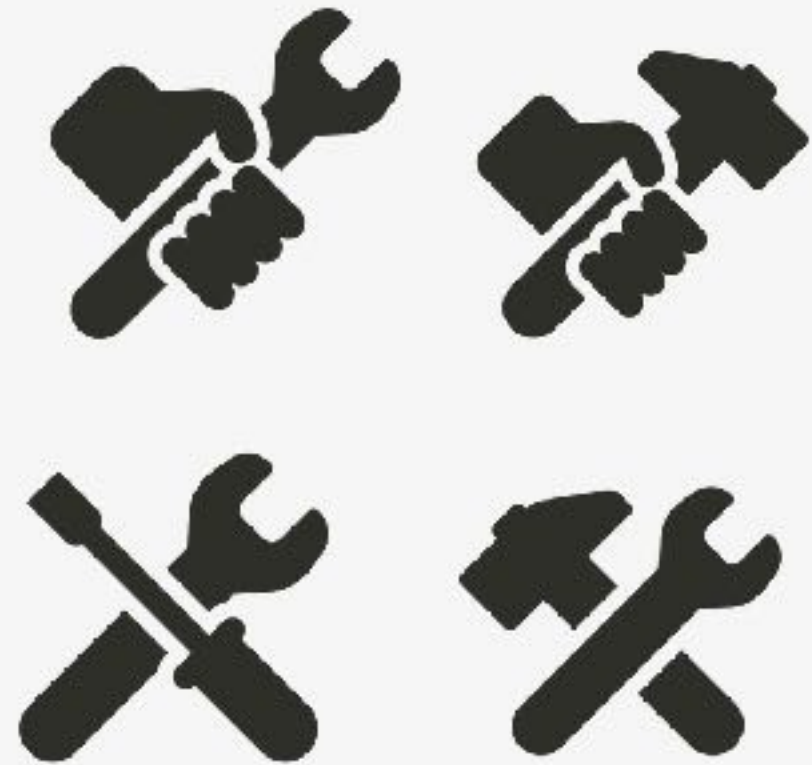


¿Porqué Debes Diseñar una Correcta API?

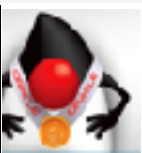
- Asegura la correcta utilización de la implementación. Seguro y Confiable.
- Minimiza el tiempo de desarrollo de una aplicación.
- Minimiza la cantidad de código a desarrollar. Y como cada línea de código es un bug potencial ...
- A veces el término contrato no es claro. Pensemos un ejemplo.



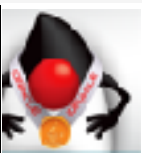
Servicios (Legados y Nuevos)



Defining End-Points.



Herramientas.



Simple, Cualquiera Puede...

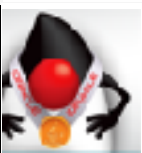


Download from
Dreamstime.com

This is a low-resolution sample image for previewing purposes only.

ID 7951050

Avi | Dreamstime.com



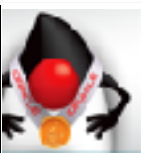
Java
Champions



we9x



Soy Muy Inteligente



La Solución Perfecta —> Un Contrato Correcto

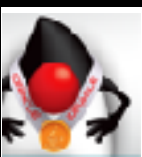


Al Final Es un Todo



Definir Mensaje

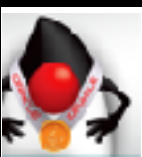
- Forma de Envío:
 - Soap (JAX-WS)
 - REST (JAX-RS) - Modelo de Maduración de Richardson.
- Formato del mensaje.
 - XML.
 - JSON.
 - PlainText.



Formato Mensaje - XML

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  <SOAP-ENV:Body>
    <ns1:cotizacion xmlns:ns1="Petroleo">
      <op1 xsi:type="xsd:string">Brent</op1>
      <op2 xsi:type="xsd:integer">400</op2>
    </ns1:cotizacion>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

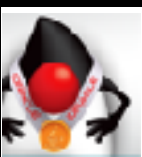
- JAXB



Formato Mensaje - JSON

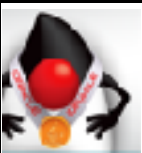
```
{  
  "Cotizacion":{  
    "op1 ":"Brent",  
    "op2 ":"400"  
  }  
}
```

- JAXP - GSON



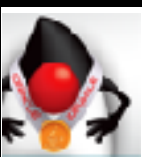
Formato Mensaje - Plain Text

- Cotizacion|Brent|400
- Simplemente split



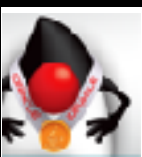
Http Codes ?

- ¿Se recomiendan basarse en los códigos de respuesta?
 - **200** OK - **201** Created - **202** Accepted - **301** Moved Permanently- **302** Found - **303** See Other - **400** Bad Request - **401** Unauthorized - **403** Forbidden - **404** Not Found -**500** Internal Server Error.
- ¿O usar códigos de error internos?
 - {"code":**0**,"httpCode":**200**,"message":"**OK**","jsonResponse":""}



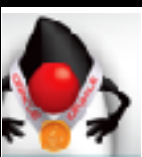
Http Methods ?

- ¿Se recomiendan basarse en los métodos de http? **SI**
 - **GET** - Trae datos de un recurso específico.
 - **POST** - Envía datos para que sea procesado por un recurso.
 - **PUT** - Carga una representación de una URI.
 - **DELETE** - Elimina un recurso específico.
 - **OPTIONS** - Regresa los métodos que soporta un server HTTP.



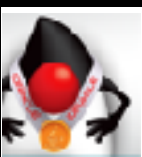
¿Usar Verbos y Versión en el URL?

- ¿Se recomiendan agregar verbos en el URL? **Si, no es indispensable, y debe analizarse el caso.**
 - `https://miservidor:8080/app/producto/add`
- ¿Se recomiendan agregar versión en el URL? **Si, no es indispensable, y debe analizarse el caso.**
 - `https://miservidor:8080/app/producto/v1/add`



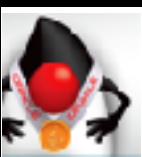
Cuida el Tamaño del Mensaje

```
{"code":0,"httpCode":200,"jsonResponse":{"objs":[{"id":400,"title":"Combo Papas","desc":"\u003cp\u003e- Cup\u0026acute;n para un hotdog y un refresco mediano de cualquier sabor.\u003c/p\u003e\u003cp\u003e- Cin\u0026acute;polis Tradicional de lunes a domingo.\u003c/p\u003e","otherInfo":"\u003cp\u003eAl confirmar tu compra recibir\u0026acute;s un correo con el \u003cstrong\u003ec\u0026acute;digo\u003c/strong\u003e para canjearlo en la sucursal MiCinito de tu elecci\u0026acute;n. Recuerda que tambi\u0026acute;n puedes consultar tu c\u0026acute;digo ingresando a la secci\u0026acute;n de \u003cem\u003eCompras en misitio\u003c/em\u003e" desde el men\u0026uacute;. \u003cp\u003e\u003c/n\u003cp\u003eLos c\u0026acute;digos tienen una validez de \u003cstrong\u003e15 d\u0026iacute;as\u003c/strong\u003e a partir de la fecha de compra.\u003c/p\u003e","icon":"imagenes/Resources/MiCinitoStore/pack_Papas.png","price":0.05,"typeCurrency":"MXN","stock":27},{id":500,"title":"Combo Hotdog","desc":"\u003cp\u003e- Cup\u0026acute;n para una Galleta dulce de dos ingredientes y caf\u0026acute; americano mediano (16 oz).\u003c/p\u003e\u003cp\u003e- Cin\u0026acute;polis Tradicional de lunes a domingo.\u003c/p\u003e","otherInfo":"\u003cp\u003eAl confirmar tu compra recibir\u0026acute;s un correo con el \u003cstrong\u003ec\u0026acute;digo\u003c/strong\u003e para canjearlo en la sucursal MiCinito de tu elecci\u0026acute;n. Recuerda que tambi\u0026acute;n puedes consultar tu c\u0026acute;digo ingresando a la secci\u0026acute;n de \u003cem\u003eCompras en misitio\u003c/em\u003e" desde el men\u0026uacute;. \u003cp\u003e\u003c/n\u003cp\u003eLos c\u0026acute;digos tienen una validez de \u003cstrong\u003e15 d\u0026iacute;as\u003c/strong\u003e a partir de la fecha de compra.\u003c/p\u003e","icon":"imagenes/Resources/MiCinitoStore/pack_hotdog.png","price":0.02,"typeCurrency":"MXN","stock":185},{id":106,"title":"Combo Galleta","desc":"\u003cp\u003e- Cup\u0026acute;n para unos Papas con queso y un refresco mediano de cualquier sabor.\u003c/p\u003e\u003cp\u003e- Cin\u0026acute;polis Tradicional de lunes a domingo.\u003c/p\u003e","otherInfo":"\u003cp\u003eAl confirmar tu compra recibir\u0026acute;s un correo con el \u003cstrong\u003ec\u0026acute;digo\u003c/strong\u003e para canjearlo en la sucursal MiCinito de tu elecci\u0026acute;n. Recuerda que tambi\u0026acute;n puedes consultar tu c\u0026acute;digo ingresando a la secci\u0026acute;n de \u003cem\u003eCompras en misitio\u003c/em\u003e" desde el men\u0026uacute;. \u003cp\u003e\u003c/n\u003cp\u003eLos c\u0026acute;digos tienen una validez de \u003cstrong\u003e15 d\u0026iacute;as\u003c/strong\u003e a partir de la fecha de compra.\u003c/p\u003e","icon":"imagenes/Resources/MiCinitoStore/pack_Galletas.png","price":0.06,"typeCurrency":"MXN","stock":214}], "error":{"code":0,"retry":0}}
```



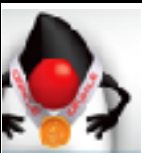
¿Y la Seguridad?

- Mecanismos de Token.
- Encriptación de request y response.
- Uso de canales seguros (https).
- Uso de medios seguros (VPN).
- Frameworks de autenticación (OAuth, AccessToken, etc).



Aplicaciones Para Diseñar y Construir

- APIARY
- SWAGGER
- VI, SUBLIME, EMACS, ETC.
- WORD.



APIary

ORACLE + apiary

[How Apiary Works](#) [Product](#) [Plans](#) [Company](#) [Help](#) [Sign In](#)

[Sign Up](#)

BUILDING AN API

Fast-track your API Design

Write an API in 30 minutes. Share it with your teammates or customers. Let them use the API mock to take your API for a spin--without writing any code. Iterate, rinse & repeat. Coding can wait until you know what your developers really need.



ORACLE + apiary

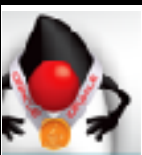
[How Apiary Works](#) [Product](#) [Plans](#) [Company](#) [Help](#) [Sign In](#)

[Sign Up](#)

SHARING AN API

An API Is Only As Good As Its Documentation

Time and time again, developers list quality documentation as the most important API feature. Good news! While you were building your API in Apiary, we've been building the documentation all along. Interactive, rich documentation your developers will love.



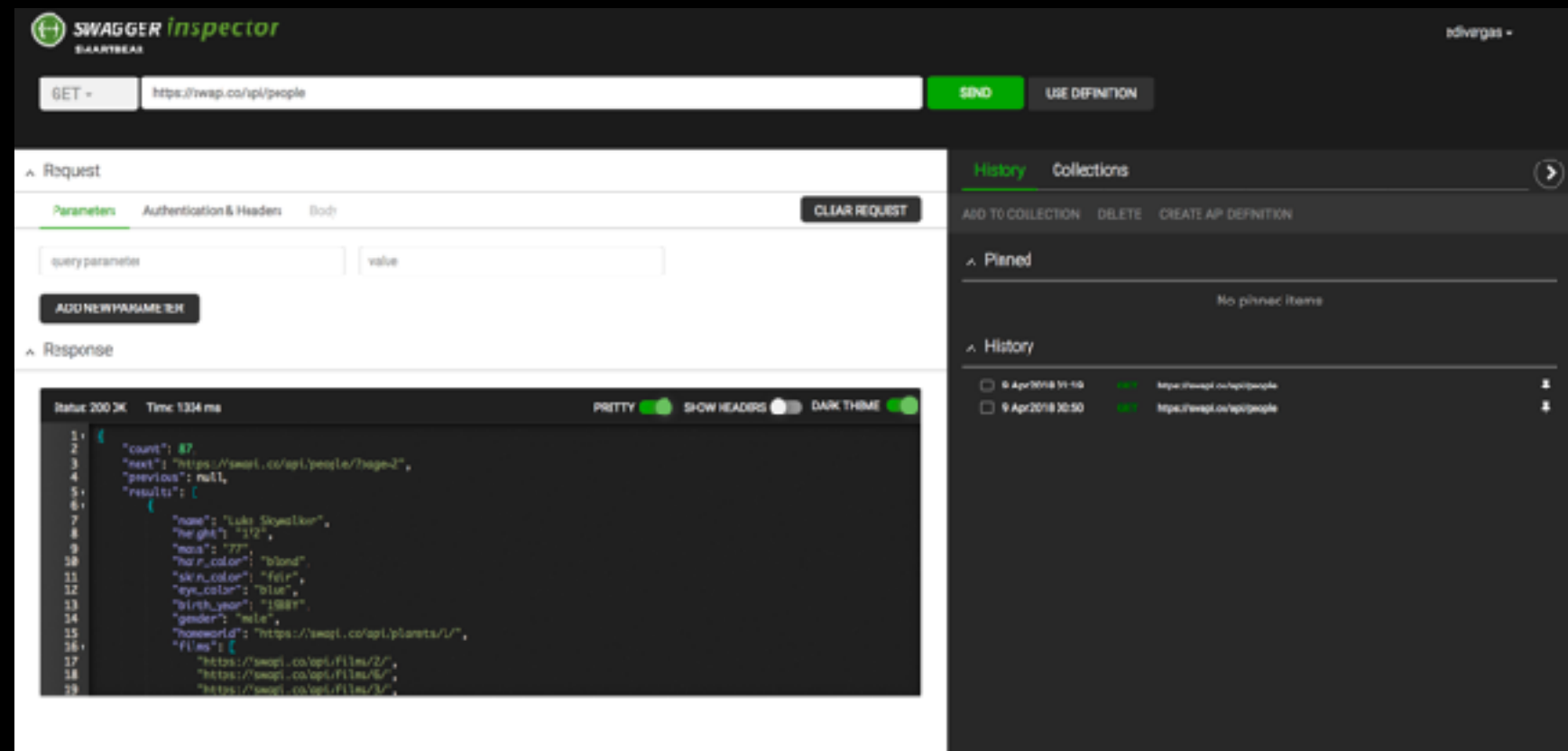
Java
Champions



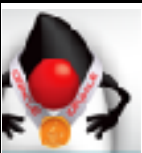
weax



SWAGGER



- <https://swagger.io/tools/>
- <https://inspector.swagger.io/builder>
- <https://editor.swagger.io/>
- <https://swagger.io/swagger-codegen/>



Swagger Editor

FileEditGenerate ServerGenerate ClientBuild back to swagger editor

```
1 swagger: "2.0"
2 info:
3   description: "This is a sample server Petstore server. You can find out more about Swagger at [http://swagger
4     .io](http://swagger.io) or on [irc.freenode.net, #swagger](http://swagger.io/irc/). For this sample, you can
5     use the api key 'special-key' to test the authorization filters."
6   version: "1.0.0"
7   title: "Swagger Petstore"
8   termsOfService: "http://swagger.io/terms/"
9   contact:
10     email: "apiteam@swagger.io"
11   license:
12     name: "Apache 2.0"
13     url: "http://www.apache.org/licenses/LICENSE-2.0.html"
14 host: "petstore.swagger.io"
15 basePath: "/v2"
16 tags:
17   - name: "pet"
18     description: "Everything about your Pets"
19     externalDocs:
20       description: "Find out more"
21       url: "http://swagger.io"
22   - name: "store"
23     description: "Access to Petstore orders"
24   - name: "user"
25     description: "Operations about user"
26   externalDocs:
27     description: "Find out more about our store"
28     url: "http://swagger.io"
29 schemes:
30   - "http"
31 paths:
32   /pet:
33     post:
34       tags:
35       - "pet"
36       summary: "Add a new pet to the store"
37       description: ""
38       operationId: "addPet"
39       consumes:
40       - "application/json"
41       - "application/xml"
42       produces:
43       - "application/xml"
44       - "application/json"
45       parameters:
46       - in: "body"
47         name: "body"
48         description: "Pet object that needs to be added to the store"
49         required: true
50         schema:
51           $ref: "#/definitions/Pet"
52       responses:
53       405:
54         description: "Invalid input"
55       security:
56       - petstore_auth:
57         - "write:pets"
58         - "read:pets"
59   out:
```

Swagger Petstore^{1.0.0}

[Base URL: petstore.swagger.io/v2]

This is a sample server Petstore server. You can find out more about Swagger at <http://swagger.io> or on irc.freenode.net, #swagger. For this sample, you can use the api key [special-key](#) to test the authorization filters.

[Terms of service](#)

[Contact the developer](#)

[Apache 2.0](#)

[Find out more about Swagger](#)

Schemes

HTTP

Authorize

pet

Everything about your Pets

Find out more: <http://swagger.io>

POST

/pet

Add a new pet to the store

PUT

/pet

Update an existing pet

GET

/pet/findByStatus

Finds Pets by status

GET

/pet/findByTags

Finds Pets by tags

GET

/pet/{petId}

Find pet by ID

POST

/pet/{petId}

Updates a pet in the store with form data

DELETE

/pet/{petId}

Deletes a pet

POST

/pet/{petId}/uploadImage

uploads an image

store

Access to Petstore orders

GET

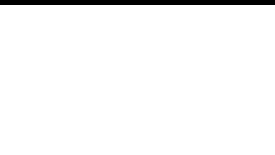
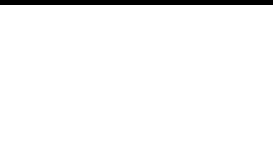
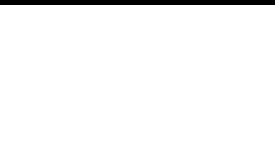
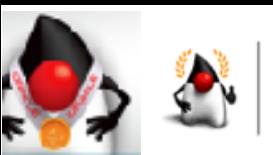
/store/inventory

Returns pet inventories by status

POST

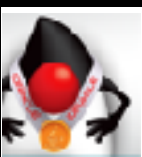
/store/order

Place an order for a pet



Conclusión Rápida

- Las aplicaciones móviles es una necesidad imperante.
- Seleccionar la arquitectura de backend depende de la naturaleza de la app, presupuesto y experiencia.
- Arquitectura de despliegue a seleccionar depende de más que la moda. Se debe seleccionar el nivel de granularidad correcto.
- Necesidad de construir una API consistente, robusta y mantenible.
- Se debe decidir, el como se hará el **CONTRATO**.
- Utilización de herramientas disponibles: Apiary y Swagger.



Q & a

Capítulo 1 Temporada 2
Track: Java, Apps
Móviles y su nube

Jorge Vargas

CTO **weex**

SUN/Oracle Java Champion

@edivargas



Diseñando API's de Microservicios Para Mobile.

Gracias!!!

Jorge Vargas

CTO **weex**

SUN/Oracle Java Champion

@edivargas

Capítulo 1 Temporada 2
Track: Java, Apps
Móviles y su nube



Arquitectura de Microservicios

- Configuration Service (Archaius / Netflix).
- Discovery Service (Eureka / Netflix).
- Load Balancing (Ribbon / Netflix).
- Circuit Breaker (Hystrix / Netflix).
- Edge Service (Zuul / Netflix)..
- Log Management (Graylog / OS).

