

Apiary

Juan González Iglesias
Andrés Filipe Oliveira da Silva
Lucas Rega Posada



UNIVERSIDADE DA CORUÑA

Integración de Aplicacións
10 de Decembro de 2020

- 1 Introducción
- 2 Instalación
- 3 Características
- 4 API Blueprint
- 5 Caso de estudio 1
- 6 Caso de estudio 2
- 7 Conclusiones



- Permítenos desarrollar diseños de API para logo implementalos.
- Usa un Server Mock para probar as nosas APIs.

Instalación

	Free	Standard	Pro
Pricing Apiary is a part of Oracle Cloud. You need to create or link your existing Oracle Cloud account for upgrade to Standard and Pro.	Free	Per user in team	Per user in team
	Read Oracle Cloud Pricing		
Core Toolset <ul style="list-style-type: none">• API Description Editor• Mock Server• Interactive Documentation• Testing Debugger• Automated Implementation Testing (using Dredd)• RSS feeds for API changes	✓	✓	✓
API Projects Create as many API Projects as you want — Apiary supports API Blueprint and OpenAPI (Swagger) formats.	Unlimited	Unlimited	Unlimited
Collaborators Number of participants in an API Project or team.	5 per API Project No team	10 per API Project Unlimited team members	Custom

- O uso das funcionalidades máis básicas é gratuito
- Permite a colaboración entre varias personas divididas en equipos si se paga unha subscripción.



- É necesario instalar a extensión Browser Console



DREDD

No more outdated API Documentation



node



php



- Para a realización dos tests é convinte instalar Dredd


Creación da API

- Crear unha nova API

New API


Personal API

✓



Andres

Team API



Create a team

New API name

☐ Private API

Start your API in [API Blueprint](#)

You can change selected format anytime.

Create API

1ª Característica: Documentación

The screenshot displays the Apify API documentation for the 'IA-TT' project. The interface is divided into two main sections: a left sidebar and a right main panel.

Left Sidebar:

- Top: 'Download' button and 'API-Busquet' text.
- Menu: 'REFERENCE' (highlighted), 'Pedidos', and 'Products'.
- Section Headers: 'Pedidos' and 'Products'.
- Actions: Under 'Pedidos', 'Devolver los pedidos de un cliente'. Under 'Products', 'Devolver un producto' and 'Crear un novo product'.

Right Main Panel:

- Section: 'Products - Devolver un producto'.
- Method: 'GET'.
- URL: 'http://localhost:7001/v1/ia/tt/services/products/id'.
- Parameters: A table with one parameter: 'id' (Number), with a description 'ID del producto a buscar' and an example 'Ejemplo: 1'.
- Request: A dropdown menu showing 'Production' and 'Raw', with a 'Try' button.
- Response: A 'JSON' tab showing a sample response structure with fields like 'id', 'name', 'description', 'price', 'category', 'status', and 'created_at'.

- Human Column

IA-TT

REFERENCE

Pedidos

Devolver los pedidos de un cliente



Devuelve los pedidos de un cliente.

Products

Devolver un producto



Devuelve un código 200 con el body JSON

Crear un novo product



Devuelve un 201 y el nuevo producto creado

Modificar un product



Devuelve un 201 y el nuevo producto creado

Eliminar un product



Machine Column

The screenshot displays the Apify API client interface for a specific endpoint. At the top, there's a close button (X) and a 'Switch to Console' button. The main header shows 'Products' and the endpoint name 'Devolver un producto'. Below this, the HTTP method is set to 'GET' and the URL is 'http://localhost:7070/rs-fairstutorial-service/products/id'. The 'Parameters' section shows a single parameter 'id' of type 'Number' with a description 'ID del producto a buscar Example: 1'. The 'Request' section has two dropdown menus for 'Production' and 'Raw', and a 'Try' button. The 'Response' section shows a '200' status code, 'HEADERS' with 'Content-Type: application/xml', and the 'BODY' containing an XML document.

Products Devolver un producto

GET http://localhost:7070/rs-fairstutorial-service/products/id

Parameters

id ID del producto a buscar Example: 1 Number

Request

Production Raw Try

Response

200

HEADERS

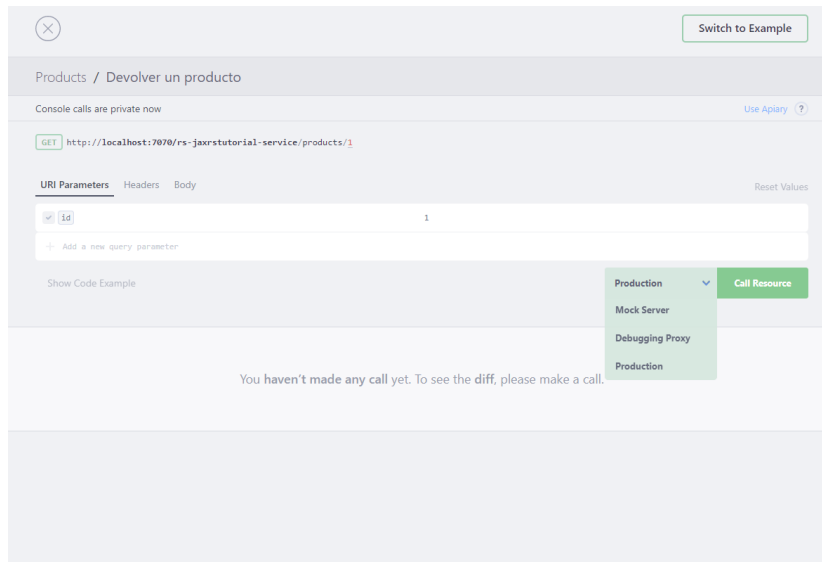
Content-Type: application/xml

BODY

```
01: <?xml version="1.0" encoding="UTF-8"?>
02: <product xmlns="http://rs.udc.es/products/xml">
03:   <id:1/id>
04:   <name>Product 1</name>
05:   <price>10.0</price>
06:   <description>Description of Product 1</description>
07: </product>
```

Características - Console

● Consola



The screenshot displays the Apify Console interface for a REST client. At the top right, there is a button labeled "Switch to Example". The main header shows the path "Products / Devolver un producto". Below this, a message states "Console calls are private now" with a link to "Use Apify". The URL bar shows a GET request to "http://localhost:7070/rs-jaxrstutorial-service/products/1". Underneath, there are tabs for "URI Parameters", "Headers", and "Body", with "URI Parameters" currently selected. A "Reset Values" link is also present. The "URI Parameters" section shows a single parameter "id" with a value of "1". Below this is a button to "Add a new query parameter". On the right side, there is a dropdown menu for selecting the environment, currently set to "Production". The dropdown options are "Production", "Mock Server", "Debugging Proxy", and "Production". A green button labeled "Call Resource" is next to the dropdown. At the bottom, a message says "You haven't made any call yet. To see the diff, please make a call."

Características - Console Cont.

Switch to Example

Products / Modificar un product

Console calls are private now [Use Apiary ?](#)

PUT `http://localhost:7070/rs-jaxrstutorial-service/products/1`

URI Parameters

Headers

Body

```
<?xml version="1.0" encoding="UTF-8"?>
<product xmlns="http://ws.udc.es/products/xml">
  <name>Product 1 new</name>
  <price>53.0</price>
  <description>Description of Product 1 new</description>
</product>
```

Browser console

----- **Browser** -----

Requests are made directly from the browser to the API. Works only with CORS enabled APIs. [Learn more](#)

Cloud console (default)

---- **via Apiary** ---- -----

Requests are made with help of Apiary infrastructure. Works with all HTTP APIs reachable from the internet. [Learn more](#)

Show Code Example

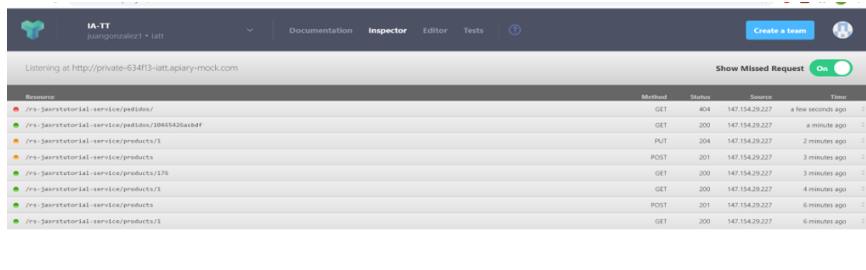
Production

▼

Call Resource

Características - Inspector

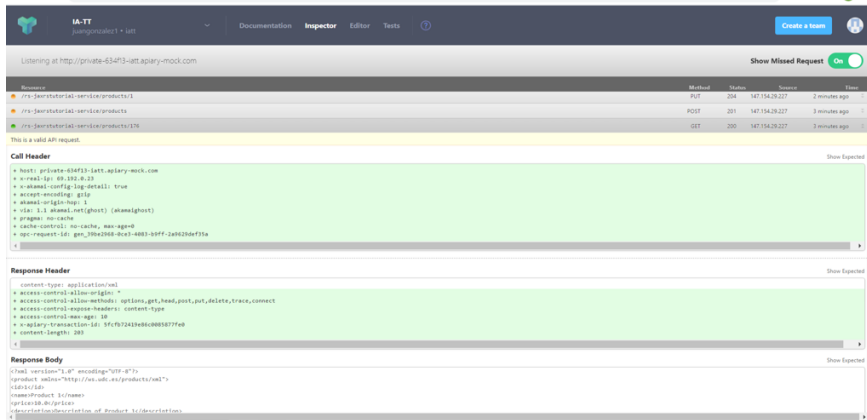
2ª característica: Inspector



The screenshot shows the Apify API Inspector interface. At the top, there's a navigation bar with the Apify logo, the user 'JA-TT', and links for Documentation, Inspector (active), Editor, and Tests. A 'Create a team' button is on the right. Below the navigation bar, it says 'Listening at http://private-634f13-latt.apify-mock.com'. A 'Show Missed Request' toggle is set to 'On'. The main area displays a table of API requests.

Resource	Method	Status	Source	Time
/rs-jaxstutorial-service/pedidos/	GET	404	147.154.29.227	a few seconds ago
/rs-jaxstutorial-service/pedidos/10465420asbdf	GET	200	147.154.29.227	a minute ago
/rs-jaxstutorial-service/products/1	PUT	204	147.154.29.227	2 minutes ago
/rs-jaxstutorial-service/products	POST	201	147.154.29.227	3 minutes ago
/rs-jaxstutorial-service/products/176	GET	200	147.154.29.227	3 minutes ago
/rs-jaxstutorial-service/products/1	GET	200	147.154.29.227	4 minutes ago
/rs-jaxstutorial-service/products	POST	201	147.154.29.227	6 minutes ago
/rs-jaxstutorial-service/products/1	GET	200	147.154.29.227	6 minutes ago

Características - Inspector Cont.



The screenshot displays the Apify API Inspector interface. At the top, the user is logged in as 'IA-TT' (juangonzalez1 • iatt). The interface shows a list of requests being listened to at the URL 'http://private-634f3-iatt.apify-mock.com'. A 'Show Missed Request' toggle is set to 'On'.

Subresource	Method	Status	Source	Time
/rs-javascript-service/products/1	PUT	204	147.154.29.227	2 minutes ago
/rs-javascript-service/products	POST	201	147.154.29.227	3 minutes ago
/rs-javascript-service/products/176	GET	200	147.154.29.227	3 minutes ago

Below the table, a message states: 'This is a valid API request.'

The 'Call Header' section shows the following details:

- Host: private-634f3-iatt.apify-mock.com
- Real-IP: 69.192.0.23
- X-akamai-config-log-detail: true
- Accept-encoding: gzip
- X-akamai-origin-hop: 1
- Via: 1.1 akamai.net(ghost) (akamai/ghost)
- Pragma: no-cache
- Cache-control: no-cache, max-age=0
- opc-request-id: get_39be266-8ce3-4883-b9ff-2a6629def35a

The 'Response Header' section shows the following details:

- Content-type: application/xml
- Access-control-allow-origin: *
- Access-control-allow-methods: options, get, head, post, put, delete, trace, connect
- Access-control-expose-headers: content-type
- Access-control-max-age: 10
- Apify-request-id: 5fcfb7243e8bc005877fe0
- Content-length: 289

The 'Response Body' section shows the following XML content:

```
<?xml version="1.0" encoding="UTF-8"?>
<product xmlns="http://ws.wu.ch/products/xml">
  <id>1</id>
  <name>Product 1</name>
  <price>10.0</price>
  <description>Description of Product 1</description>
```

3ª característica: Editor

The screenshot displays the Apify API Editor interface. The left pane shows the API definition in a code editor, and the right pane shows a visual representation of the API endpoints.

API Definition (Left Pane):

- Base URL:** `http://ws.abc.es/products`
- Endpoint 1:** `GET /products/{id}`
 - Description:** Devuelve un código 200 con el body JSON
 - Parameters:**
 - `id: '1' (number, required) - ID del producto a buscar`
 - Response 200 (application/xml):**

```
<?xml version="1.0" encoding="UTF-8"?>
<product xmlns="http://ws.abc.es/products/xml">
  <id>1</id>
  <name>Product 1</name>
  <price>10.0</price>
  <description>Description of Product 1</description>
</product>
```
- Endpoint 2:** `POST`
 - Description:** Devuelve un 201 y el nuevo producto creado
 - Request (application/xml):**

```
<?xml version="1.0" encoding="UTF-8"?>
<product xmlns="http://ws.abc.es/products/xml">
  <name>Product 4</name>
  <price>50.0</price>
  <description>Description of Product 4</description>
</product>
```
 - Response 201 (application/xml):**

```
<?xml version="1.0" encoding="UTF-8"?>
<product xmlns="http://ws.abc.es/products/xml">
  <id>4</id>
  <name>Product 4</name>
```

API Reference (Right Pane):

- Reference > Products**
- Products**
- Devolver un producto**
 - Devuelve un código 200 con el body JSON
- Crear un novo product**
 - Devuelve un 201 y el nuevo producto creado
- Modificar un product**
 - Devuelve un 201 y el nuevo producto creado
- Eliminar un product**
 - Eliminamos o customer creado anteriormente

4ª característica: Tests


The screenshot displays the Apify API client interface. On the left, a list of API calls is shown, each with a status icon, host, path, and duration. The selected call is a POST request to `/products` with a status of 201. The main panel shows the details of this call, including the request body (XML) and the response (201 status with XML body).

API Client Interface:

- Local Development** | **Continuous Integration** | **Tutorial**
- API** | **Details**
- Status:** passed
- Hostname:** LAPTOP-0VORDP8Q
- Duration:** 00:00:456
- Results:** Passes: 4, Failures: 0
- API Calls List:**
 - GET /products/1 (Devolver un producto) - 200
 - POST /products (Crear un novo product) - 201
 - PUT /products/1 (Modificar un product) - 204
 - DELETE /products/1 (Eliminar un product) - 204
- POST /products**
 - Duration:** 46 ms
 - Timestamp:** 30 Nov, 2020, 17:57:12 GMT
 - Status:** This is a valid API call.
 - Request:**
 - Location: /products/3
 - Content-Type: application/xml
 - User-Agent: Dredd/13.1.2 (Windows_NT 10.0.19041; x64)
 - ```
<?xml version="1.0" encoding="UTF-8" ?>
<product xmlns="http://ws.udc.es/products/xml">
 <name>Product 4</name>
 <price>50.0</price>
 <description>Description of Product 4</description>
</product>
```
  - Response 201**
    - connection: close
    - date: mon, 30 nov 2020 17:57:12 gmt
    - location: http://localhost:7070/rs-jaxrstutorial-service/products/3
    - content-type: application/xml
    - content-length: 219



# API Blueprint

 **A**

```
1 | FORMAT: 1A
2 | HOST: http://localhost:7070/rs-jaxrstutorial-service
3 |
4 | # IA-TT
5 |
6 | ## Pedidos [/pedidos]
7 |
8 | ### Devolver los pedidos de un cliente [GET /pedidos/{idCliente}]
9 |
10 | Devuelve los pedidos de un cliente.
11 |
12 | + Parameters
13 | + idCliente: `1046` (number, required) - ID del producto a buscar
14 |
15 | + Response 200 (application/xml)
16 |
17 | + Body
18 |
19 | <?xml version="1.0" encoding="UTF-8"?>
20 | <orders xmlns="http://ws.udc.es/products/xml">
21 | <order xmlns="http://ws.udc.es/products/xml">
22 | <id>1</id>
23 | <name>Product 1</name>
24 | <OrderStatus>DELIVERED</OrderStatus>
25 | <description>Description of Product 1</description>
26 | </order>
27 | <order xmlns="http://ws.udc.es/products/xml">
28 | <id>2</id>
29 | <name>Product 2</name>
30 | <OrderStatus>DELIVERED</OrderStatus>
31 | <description>Description of Product 2</description>
32 | </order>
33 | <order xmlns="http://ws.udc.es/products/xml">
34 | <id>3</id>
35 | <name>Product 3</name>
36 | <OrderStatus>DELIVERED</OrderStatus>
37 | <description>Description of Product 3</description>
38 | </order>
39 | </orders>
40 |
```

Referer

IA-TT

REFERENCE


Pedidos

Devolver lo


Devuelve los pe

Products

# API Blueprint Cont.

 **A**

```
46 - ### Devolver un producto [GET /products/{id}]
47
48 Devuelve un codigo 200 con el body JSON
49
50 + Parameters
51 + id: '1' (number, required) - ID del producto a buscar
52
53 + Response 200 (application/xml)
54
55 <?xml version="1.0" encoding="UTF-8"?>
56 <product xmlns="http://ws.udc.es/products/xml">
57 <id>1</id>
58 <name>Product 1</name>
59 <price>10.0</price>
60 <description>Description of Product 1</description>
61 </product>
62
63
64 - ### Crear un novo product [POST]
65
66 Devuelve un 201 y el nuevo producto creado
67
68 + Request (application/xml)
69
70 <?xml version="1.0" encoding="UTF-8"?>
71 <product xmlns="http://ws.udc.es/products/xml">
72 <name>Product 4</name>
73 <price>50.0</price>
74 <description>Description of Product 4</description>
75 </product>
76
77 + Response 201 (application/xml)
78
79 <?xml version="1.0" encoding="UTF-8"?>
80 <product xmlns="http://ws.udc.es/products/xml">
81 <id>3</id>
82 <name>Product 4</name>
83 <price>50.0</price>
84 <description>Description of Product 4</description>
85 </product>
```

 **Referer**

REFERENCE

## Pedidos

[Devolver lo](#)

Devuelve los pr

## Products

[Devolver un](#)

Devuelve un co

[Crear un no](#)



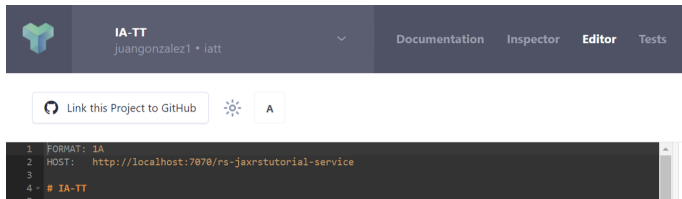
## Ejemplo: ProductService (2)

- El servicio proporciona operaciones para
  - Añadir un Producto
  - Encontrar un Producto a partir de su identificador
  - Actualizar un Producto
  - Eliminar un Producto
  - Encontrar todos los productos que contengan una cierta palabra clave en su nombre
- Un Producto tiene los siguientes atributos
  - Identificador
  - Nombre
  - Precio
  - Descripción

Product
- id: Long - name: String - price: double - description: String
+ Constructor + métodos get / set

# Caso de estudio 1 Cont.

- Pestana **Editor**.
- Declarar *Format*, *Host* e nome da **API**.



# Caso de estudio 1 - Devolver un producto

- Declarar o *Resource* **Products**
- A operación é de tipo **GET** á que se lle pasa un *Id*.
- Indicar a resposta que se espera con código **200**.

```
Products [/products]

Devolver un producto [GET /products/{id}]

Devuelve un codigo 200 con el body JSON

+ Parameters
 + id: `1` (number, required) - ID del producto a buscar

+ Response 200 (application/xml)

 <?xml version="1.0" encoding="UTF-8"?>
 <product xmlns="http://ws.udc.es/products/xml">
 <id>1</id>
 <name>Product 1</name>
 <price>10.0</price>
 <description>Description of Product 1</description>
 </product>
```

# Caso de estudio 1 - Crear un producto

- A operación é de tipo **POST**.
- Corpo da petición.
- Indicar a resposta que se espera con código **201**.

```
Crear un novo product [POST]

Devuelve un 201 y el nuevo producto creado

+ Request (application/xml)

 <?xml version="1.0" encoding="UTF-8"?>
 <product xmlns="http://ws.udc.es/products/xml">
 <name>Product 4</name>
 <price>50.0</price>
 <description>Description of Product 4</description>
 </product>

+ Response 201 (application/xml)

 <?xml version="1.0" encoding="UTF-8"?>
 <product xmlns="http://ws.udc.es/products/xml">
 <id>3</id>
 <name>Product 4</name>
 <price>50.0</price>
 <description>Description of Product 4</description>
 </product>
```

# Caso de estudio 1 - Modificar un producto

- A operación é de tipo **PUT** á que se lle pasa o *Id* do produto.
- Corpo da petición
- Indicar a resposta que se espera con código **204**.

```
Modificar un product [PUT /products/{id}]

Devuelve un 201 y el nuevo producto creado

+ Request (application/xml)

 <?xml version="1.0" encoding="UTF-8"?>
 <product xmlns="http://ws.udc.es/products/xml">
 <name>Product 1 new</name>
 <price>53.0</price>
 <description>Description of Product 1 new</description>
 </product>

+ Parameters
 + id: `1` (number, required) - ID del producto a buscar

+ Response 204 (application/xml)
```

# Caso de estudio 1 - Eliminar un producto

- A operación é de tipo **DELETE** á que se lle pasa o *Id*.
- Corpo da petición
- Indicar a resposta que se espera con código **204**.

```
Eliminar un product [DELETE /products/{id}]

Eliminamos o customer creado anteriormente

+ Parameters
 + id: `1` (number, required) - ID del producto a buscar

+ Response 204 (application/xml)
```



## IA-TT

---

### REFERENCE

### Products

#### Devolver un producto

Devuelve un código 200 con el body JSON

#### Crear un novo product

Devuelve un 201 y el nuevo producto creado


#### Modificar un product

Devuelve un 201 y el nuevo producto creado

#### Eliminar un product

Eliminamos o customer creado anteriormente

# Caso de estudio 1 - Machine Column

 Switch to Console

Products Devolver un producto

GET http://localhost:7070/rs-jaxrsutorial-service/products/*id*

**Parameters**

id	ID del producto a buscar Example: 1	Number
----	-------------------------------------	--------

**Request**

ProductionRaw

Try

**Response**

200

HEADERS

Content-Type: application/xml

BODY

```
01 <?xml version="1.0" encoding="UTF-8"?>
02 <product xmlns="http://ws.udc.es/products/xml">
03 <id>1</id>
04 <name>Product 1</name>
05 <price>10.0</price>
06 <description>Description of Product 1</description>
07 </product>
```

# Caso de estudio 1 - Consola de Apiary

Products / Devolver un producto

Console calls are private now [Use Apiary](#)

**GET** `http://localhost:7070/rs-jaxrstutorial-service/products/1`

**URI Parameters** Headers Body Reset Values

☒ `id` 1

[Add a new query parameter](#)

[Show Code Example](#) Production Call Resource

You haven't made any call yet. To see the diff, please make a call.

# Caso de estudio 1 - Consola de Apiary Cont.

GET http://localhost:7070/rs-jaxrstutorial-service/products/1

URI Parameters Headers Body Reset Values

☒ id 1

+ Add a new query parameter

Show Code Example Production Call Resource

> Request  
GET http://localhost:7070/rs-jaxrstutorial-service/products/1

▼ Response 200

Response Headers Real Diff Specification

- 1 content-length: 219
- 2 content-type: application/xml
- 3 date: sat, 05 dec 2020 15:37:56 gmt
- 4 server: Jetty(9.4.28.v20200408)

Response Body Real Diff Specification

```
1 <?xml version="1.0" encoding="UTF-8"?>\r\n<product xmlns="http://ws.udc.es/products/xml"?>\r\n <id>1</id>\r\n <name>Product 1</name>\r\n <price>10.0</price>\r\n <description>Description of Product 1</description>\r\n</product>\r\n"
```

# Caso de estudio 1 - Creación do recurso MOCK

- Devolver os pedidos dun cliente.
- Operación **GET** á que lle pasamos un *ID* de cliente.
- Devolve unha resposta co código **200**.

```
Pedidos [/pedidos]

Devolver los pedidos de un cliente [GET /pedidos/{idCliente}]

Devuelve los pedidos de un cliente.

+ Parameters
 + idCliente: '1046' (number, required) - ID del producto a buscar

+ Response 200 (application/xml)

 + Body

 <?xml version="1.0" encoding="UTF-8"?>
 <orders xmlns="http://ws.udc.es/products/xml">
 <order xmlns="http://ws.udc.es/products/xml">
 <id>1</id>
 <name>Product 1</name>
 <OrderStatus>DELIVERED</OrderStatus>
 <description>Description of Product 1</description>
 </order>
 <order xmlns="http://ws.udc.es/products/xml">
 <id>2</id>
 <name>Product 2</name>
 <OrderStatus>DELIVERED</OrderStatus>
 <description>Description of Product 2</description>
 </order>
 <order xmlns="http://ws.udc.es/products/xml">
 <id>3</id>
 <name>Product 3</name>
 <OrderStatus>DELIVERED</OrderStatus>
 <description>Description of Product 3</description>
 </order>
 </orders>
```

# Caso de estudio 1 - Proba do recurso Mock

Untitled Request

BUILD

GET https://private-4595dc-latt.apify-mock.com/rs-jaxrs/tutorial-service/pedidos/1046

Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION	***
Key	Value	Description	

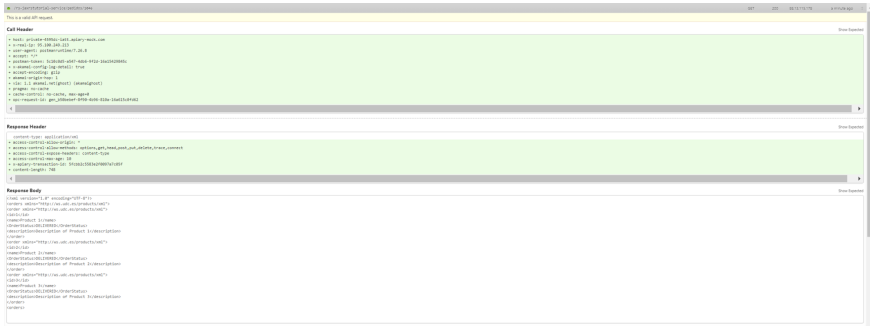
Body Cookies Headers (11) Test Results

Status: 200 OK Time: 1096 ms Size: 1.15 KB Save

Pretty Raw Preview Visualize XML

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <orders xmlns="http://ws.udc.es/products/xml">
3 <order xmlns="http://ws.udc.es/products/xml">
4 <id>1</id>
5 <name>Product 1</name>
6 <orderStatus>DELIVERED</orderStatus>
7 <description>Description of Product 1</description>
8 </order>
9 <order xmlns="http://ws.udc.es/products/xml">
10 <id>2</id>
11 <name>Product 2</name>
12 <orderStatus>DELIVERED</orderStatus>
13 <description>Description of Product 2</description>
14 </order>
15 <order xmlns="http://ws.udc.es/products/xml">
16 <id>3</id>
17 <name>Product 3</name>
18 <orderStatus>DELIVERED</orderStatus>
19 <description>Description of Product 3</description>
20 </order>
21 </orders>
```

# Caso de estudio 1 - Pestana INSPECTOR



## Caso de estudio 2 - Tests con DREDD - Config

- Configuración do *endpoint* no ficheiro **dredd.yml**
- Creación do ficheiro **apiary.apib** donde se inclúe o código blueprint.
- Hai que ter correndo o servico rs-jaxrstutorial correndo

```
color: true
reporter: apiary
custom:
 apiaryApiKey: 56623c21df2568dc026ea3b9e0d8b24b
 apiaryApiName: iatt
dry-run: null
hookfiles: null
language: nodejs
require: null
server: npm start
server-wait: 3
init: false
names: false
only: []
output: []
header: []
sorted: false
user: null
inline-errors: false
details: false
method: []
loglevel: warning
path: []
hooks-worker-timeout: 5000
hooks-worker-connect-timeout: 1500
hooks-worker-connect-retry: 500
hooks-worker-after-connect-wait: 100
hooks-worker-term-timeout: 5000
hooks-worker-term-retry: 500
hooks-worker-handler-host: 127.0.0.1
hooks-worker-handler-port: 61321
config: ./dredd.yml
blueprint: apiary.apib
endpoint: 'http://localhost:7070/rs-jaxrstutorial-service'
```



## Caso de estudio 2 - Execución dos tests

- Para executar os tests escribimos ***dredd*** dentro da carpeta donde están os ficheiros anteriores.

```
C:\Users\juanc>dredd
(node:16988) Warning: Accessing non-existent property 'padLevels' of module exports inside circular dependency
(Use `node --trace-warnings ...` to show where the warning was created)
npm ERR! code ENOENT
npm ERR! syscall open
npm ERR! path C:\Users\juanc\package.json
npm ERR! errno -4058
npm ERR! enoent ENOENT: no such file or directory, open 'C:\Users\juanc\package.json'
npm ERR! enoent This is related to npm not being able to find a file.
npm ERR! enoent

npm ERR! A complete log of this run can be found in:
npm ERR! C:\Users\juanc\AppData\Roaming\npm-cache_logs\2020-11-24T21_52_34_636Z-debug.log
warn: API description parser warning in C:\Users\juanc\apiary.apib:22 (from line 22 column 1 to line 32 column 1): Ignoring messa
ge-body list item, message-body list item is expected to be indented by 4 spaces or 1 tab
warn: API description parser warning in C:\Users\juanc\apiary.apib:48 (from line 48 column 1 to line 51 column 1): the 'headers'
section at this level is deprecated and will be removed in a future, use respective payload header section(s) instead
warn: API description parser warning in C:\Users\juanc\apiary.apib:52 (from line 52 column 1 to line 62 column 1): Ignoring messa
ge-body list item, message-body list item is expected to be indented by 4 spaces or 1 tab
(node:16988) DeprecationWarning: Deep requiring like `const uuidv4 = require('uuid/v4');` is deprecated as of uuid@7.x. Please re
quire the top-level module when using the Node.js CommonJS module or use ECMAScript Modules when bundling for the browser. See ht
tps://github.com/uuidjs/uuid#deep-requires-now-deprecated for more information.
pass: GET (200) /products/1 duration: 104ms
pass: POST (201) /products duration: 33ms
pass: PUT (204) /products/1 duration: 9ms
pass: DELETE (204) /products/1 duration: 7ms
complete: 4 passing, 0 failing, 0 errors, 0 skipped, 4 total
complete: Tests took 531ms
complete: See results in Apiary at: https://app.apiary.io/iatt/tests/run/3797bc12-d2c1-483e-8536-7c1244b42e6d
```

# Caso de estudio 2 - Resultados dos tests

The screenshot displays the Apiary web interface. At the top, there's a navigation bar with tabs for Documentation, Inspector, Editor, and Tests. The Tests tab is active. Below the navigation bar, there are three main sections: Local Development, Continuous Integration, and Tutorial. The Local Development section is selected, showing a list of test results on the left and a detailed view of the selected test on the right.

**Test Results List:**

Test Name	Duration	Time Ago
✓ @LAPTOP-0VORDP8Q	531ms	34 minutes ago
✗ @LAPTOP-0VORDP8Q	483ms	34 minutes ago
✗ @LAPTOP-0VORDP8Q	493ms	34 minutes ago
✗ @LAPTOP-0VORDP8Q	864ms	35 minutes ago
✗ @LAPTOP-0VORDP8Q	461ms	43 minutes ago
✗ @LAPTOP-0VORDP8Q	484ms	43 minutes ago
✗ @LAPTOP-0VORDP8Q	511ms	44 minutes ago
✗ @LAPTOP-0VORDP8Q	...	...

**Test Details:**

**Apiary** [Details](#)

**Status** ✓ passed  
**Hostname** LAPTOP-0VORDP8Q  
**Duration** 00:00.531  
**Results** Passes: 4, Failures: 0


Method	Endpoint	Status
GET	/products/1	200
POST	/products	201
PUT	/products/1	204
DELETE	/products/1	204

Devolver un producto  
Crear un novo product  
Modificar un product  
Eliminar un product

# Caso de estudio 2 - Resultados dos tests Cont

Apiary



Details

Status  passed



Hostname LAPTOP-0VORDP8Q

Duration 00:00.531



Results Passes: 4, Failures: 0

 GET /products/1 



Devolver un producto

 POST /products 

Crear un novo product

 PUT /products/1 

Modificar un product


 DELETE /products/1 

Eliminar un product

Crear un novo product

POST /products

∞ 33 ms © 24 Nov, 2020; 21:52:37 GMT

 This is a valid API call.

▼ Request

Location: /products/3

Content-Type: application/xml

User-Agent: Dredd/13.1.2 (Windows\_NT 10.0.19041; x64)

```
<?xml version="1.0" encoding="UTF-8" ?>
<product xmlns="http://ws.udc.es/products/xml">
 <name>Product 4</name>
 <price>50.0</price>
 <description>Description of Product 4</description>
</product>
```

▼ Response 201 Diff • Real

connection: close

date: tue, 24 nov 2020 21:52:37 gmt

location: http://localhost:7070/rs-jaxrstutorial-service/products/3

content-type: application/xml

content-length: 219

server: jetty(9.4.28.v20200408)

```
<?xml version="1.0" encoding="UTF-8" ?>
```

- Intuitiva e fácil de empregar.
- Curva de aprendizaxe moi pequena.
- Documentación interactiva e tests.

# Apiary

Juan González Iglesias  
Andrés Filipe Oliveira da Silva  
Lucas Rega Posada



UNIVERSIDADE DA CORUÑA

Integración de Aplicacións  
10 de Decembro de 2020