TESTING CON KARATE DSL: UP & RUNNING

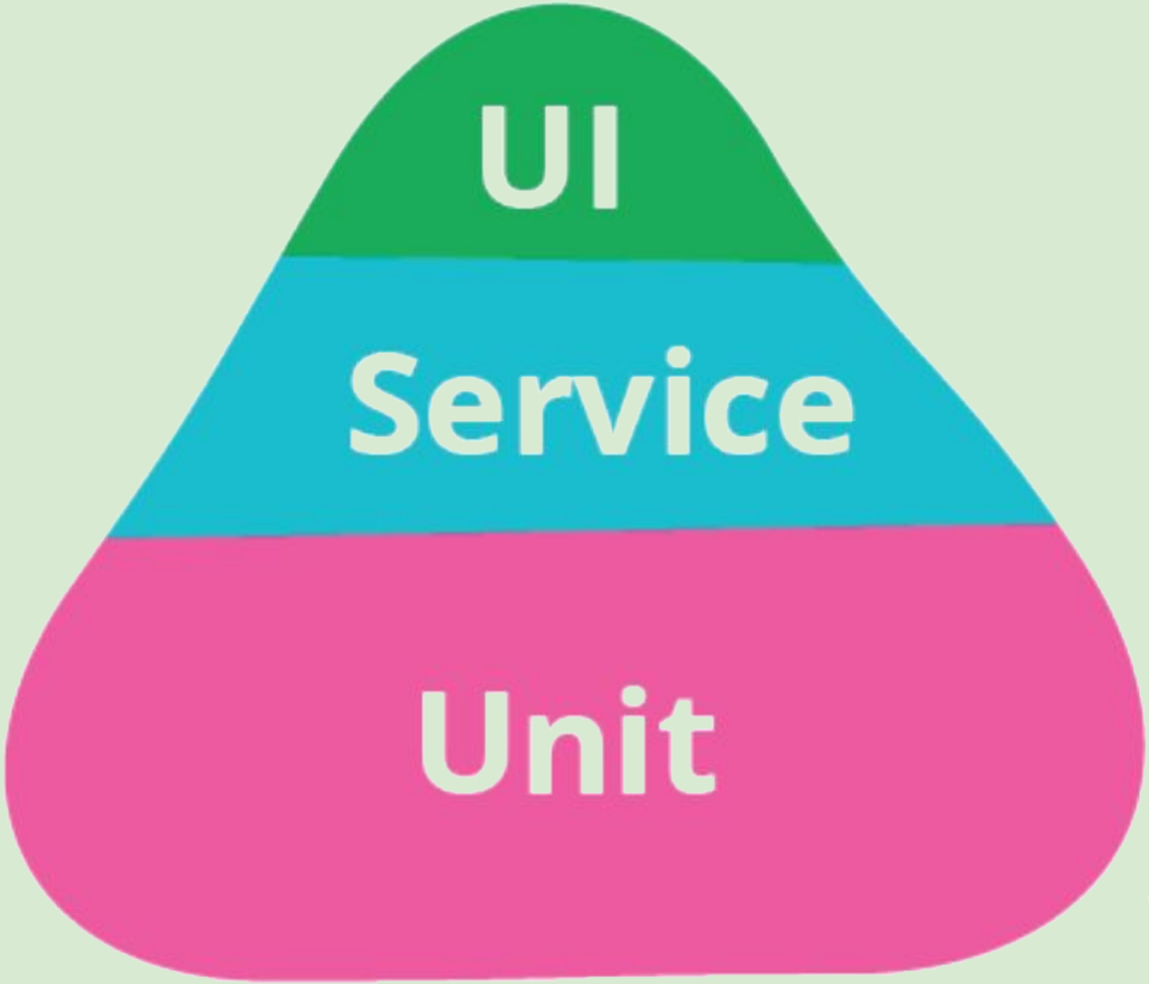
github.com/arcones/karate-meetup

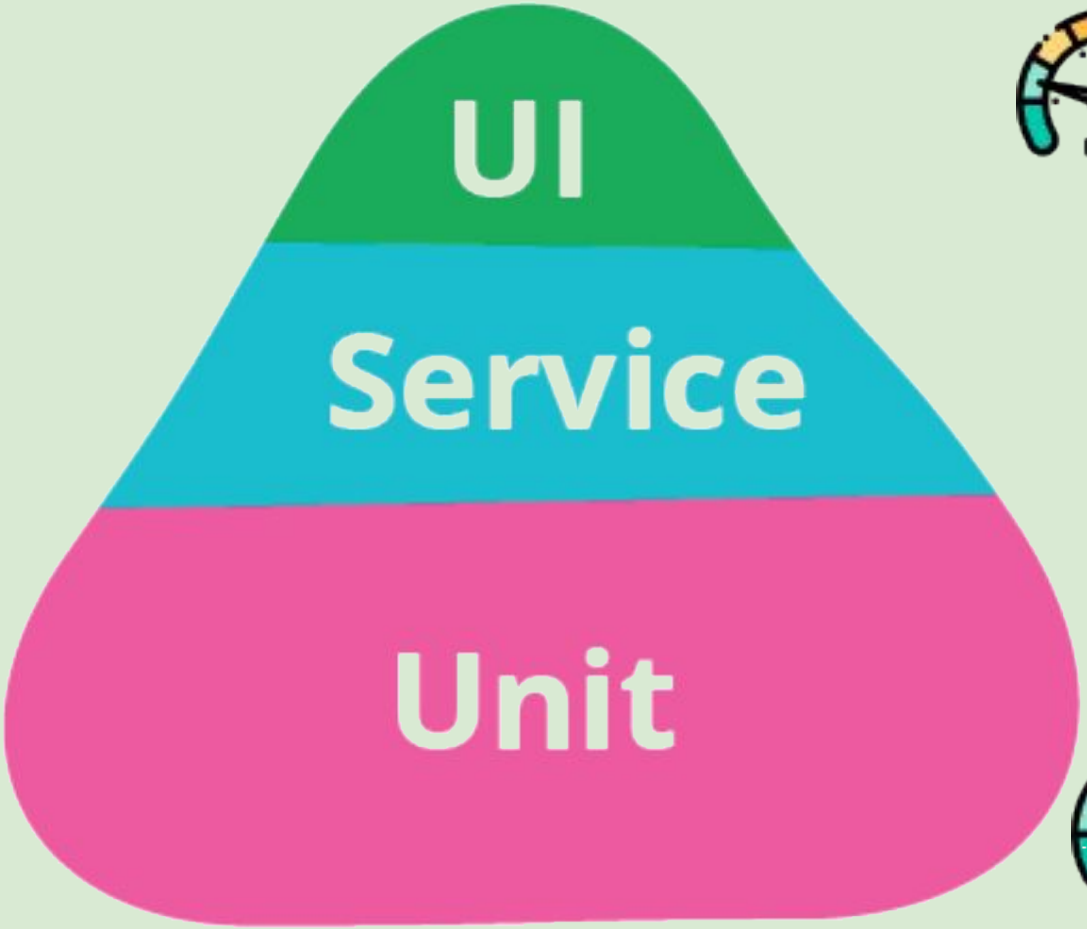
SOBRE NOSOTROS

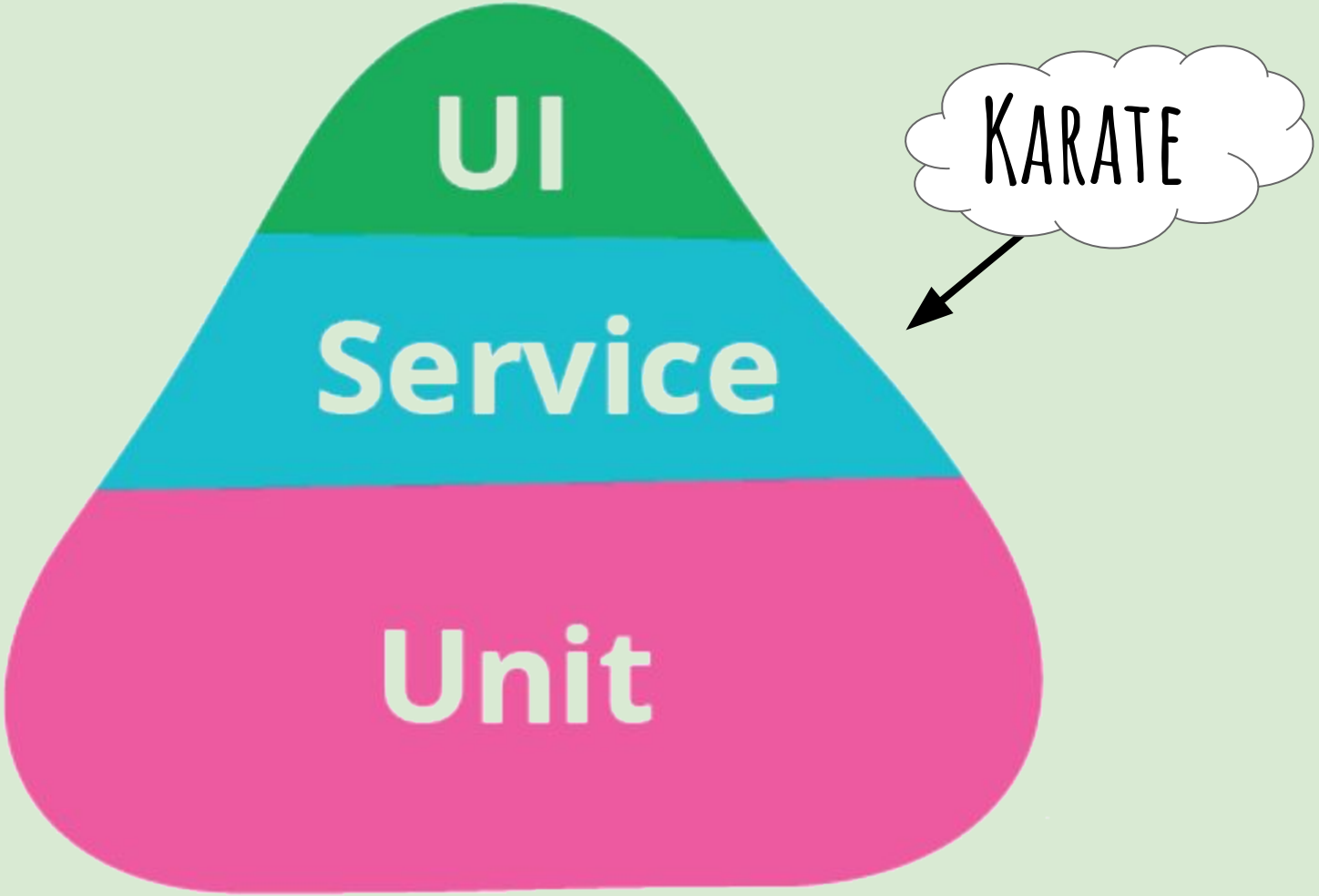
zooplus

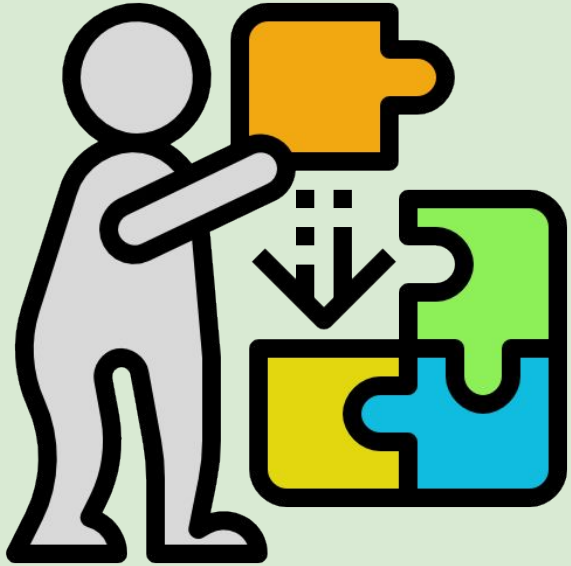

I HAVE NO IDEA
WHAT I'M DOING

ALGUNOS CONCEPTOS DE TESTING


UI
Service
Unit

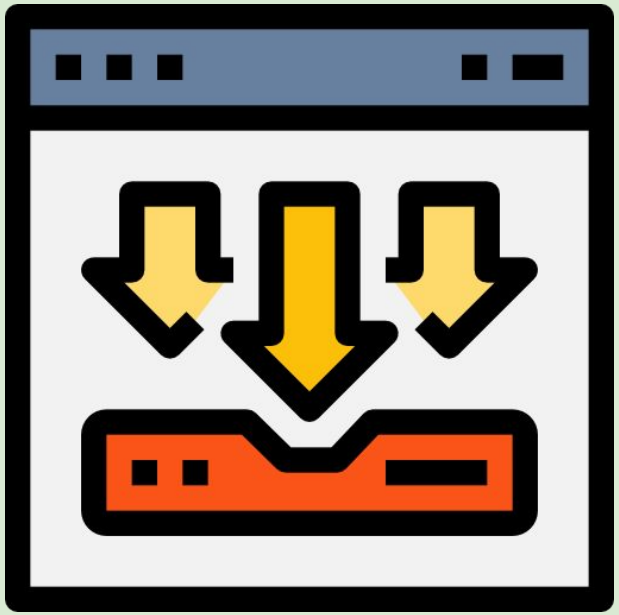

UI
Service
Unit


UI
Service
Unit
Karate



TEST DE INTEGRACIÓN

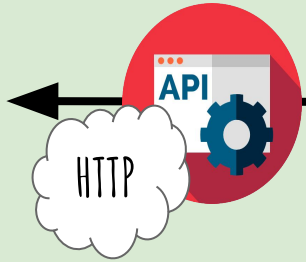
TEST DE CARGA



CAT AS A SERVICE


```
{
  "id": 15748,
  "name": "Lucifer",
  "age": 8,
  "parentId": 264
}
```

```
{
  "id": 15748,
  "catId": 1,
  "brand": "Cosma"
}
```

KARATE

HELLO "CAT"

```
Scenario: Create a cat
  Given url 'http://localhost:4567/v1/cats'
  And request { name: 'Chiquito', age: 8 }
  When method PUT
  Then status 201
  And match response == { id: '#notnull', name: 'Chiquito', age: 8, parentId: 0}
  And match response $.name == 'Chiquito'
```

```
Scenario: Create, retrieve and delete a cat
  Given url 'http://localhost:4567/v1/cats'
  And request { name: 'Satan', age: 2 }
  When method PUT
  Then status 201
  And match response == { id: '#notnull', name: 'Satan', age: 2, parentId: 0}

  Given path response.name
  When method GET
  Then status 200
  And match response == { id: '#notnull', name: 'Satan', age: 2, parentId: 0}
```

```
Scenario: Create, retrieve and delete a cat
  Given url 'http://localhost:4567/v1/cats'
  And request { name: 'Satan', age: 2 }
  When method PUT
  Then status 201
  And match response == { id: '#notnull', name: 'Satan', age: 2, parentId: 0}

  Given path response.name
  When method GET
  Then status 200
  And match response == { id: '#notnull', name: 'Satan', age: 2, parentId: 0}

  Given path response.name
  When method DELETE
  Then status 204

  Given path 'Satan'
  When method GET
  Then status 404
```

ESTRUCTURA DE LAS FEATURES

```
Feature: Try out basic Karate syntax

  Background: Area to define variables used in the scenarios of this file
    * def catName = 'Neo'

  Scenario: Greet the cat
    Given def greet = 'Guten morgen '
    Then print greet + catName
```

```
Scenario Outline: Greet several cats
  Given def cat = { name: '<visigothicName>', age: '<age>'}
  Then print 'Hello Visigothic cat ' + cat.name + ' who is ' + cat.age + ' years old'
  Examples:
    | visigothicName | age |
    | Ataulfo        | 12  |
    | Teodorico      | 2   |
    | Chindasvinto   | 9   |
    | Leovigildo     | 6   |
```

```
Scenario Outline: [1.1:12] Greet several cats
Test 1 : * def catName = 'Neo'
Test 2 : Given def cat = { name: 'Ataulfo', age: '12'}
Test 3 : Then print 'Hello Visigothic cat ' + cat.name + ' who is ' + cat.age + ' years old'
23:53:31.859 [print] Hello Visigothic cat Ataulfo who is 12 years old

Scenario Outline: [1.2:13] Greet several cats
Test 4 : * def catName = 'Neo'
Test 5 : Given def cat = { name: 'Teodorico', age: '2'}
Test 6 : Then print 'Hello Visigothic cat ' + cat.name + ' who is ' + cat.age + ' years old'
23:53:31.865 [print] Hello Visigothic cat Teodorico who is 2 years old

Scenario Outline: [1.3:14] Greet several cats
Test 7 : * def catName = 'Neo'
23:53:31.870 API URL: http://localhost:4567
Test 8 : Given def cat = { name: 'Chindasvinto', age: '9'}
Test 9 : Then print 'Hello Visigothic cat ' + cat.name + ' who is ' + cat.age + ' years old'
23:53:31.872 [print] Hello Visigothic cat Chindasvinto who is 9 years old

Scenario Outline: [1.4:15] Greet several cats
Test 10 : * def catName = 'Neo'
Test 11 : Given def cat = { name: 'Leovigildo', age: '6'}
Test 12 : Then print 'Hello Visigothic cat ' + cat.name + ' who is ' + cat.age + ' years old'
23:53:31.879 [print] Hello Visigothic cat Leovigildo who is 6 years old
```

SINTAXIS BÁSICA

```
Scenario: Print a greeting for my favourite cat
  Given def catName = 'Neo'
  Then print 'Hello ' + catName
```

Scenario: [3:17] **Print a greeting for my favourite cat**

Test 9 : * def catName = 'Neo'

Test 10 : Given def catName = 'Neo'

Test 11 : Then print 'Hello ' + catName

```
14:37:07.326 [print] Hello Neo
```

```
Background:
  * def catName = 'Neo'

Scenario: Assert that my favourite cat is happy
  Given def myCat = { name: '#(catName)', isHappy: true}
  Then assert myCat.isHappy
  And assert myCat.name == 'Neo'
```

Scenario: [1:6] **Assert that my favourite cat is happy**

Test 1 : * def catName = 'Neo'

Test 2 : Given def myCat = { name: '#(catName)', isHappy: true}

Test 3 : Then assert myCat.isHappy

Test 4 : And assert myCat.name == 'Neo'

```
Background:
  * def catName = 'Neo'

Scenario: Print my cat that is old and happy
  Given def age = 10
  And def myCat = { name: '#(catName)', age: '#(age)', isHappy: true}
  Then print myCat
```

Scenario: [1:6] **Print my cat that is old and happy**
Test 1 : * def catName = 'Neo'
Test 2 : Given def age = 10
Test 3 : And def myCat = { name: '#(catName)', age: '#(age)', isHappy: true}
Test 4 : Then print myCat

```
14:33:36.204 [print] {
  "name": "Neo",
  "age": 10,
  "isHappy": true
}
```

```
Background: Area to define variables used in the scenarios of this file
  * def catName = 'Neo'

Scenario: Print my cat that is old and grumpy
  Given def age = 10
  And def myCat = { name: '#(catName)', age: '#(age)', isHappy: false}
  And match myCat.name == 'Neo'
  But assert !myCat.isHappy
```

```
Scenario: Use a table to fill my list of favourite cats
  Given table catList
    | position | name     |
    | 1        | 'Neo'    |
    | 2        | 'Panchi' |
    | 3        | 'Nami'   |
  Then match catList == [{position:1, name: 'Neo'},{position:2, name: 'Panchi'},{position:3, name: 'Nami'}]
```

Scenario: [5:26] **Use a table to fill my list of favourite cats**

Test 17 : * def catName = 'Neo'

Test 18 : Given table catList

Test 19 : Then match catList == [{position:1, name: 'Neo'},{position:2, name: 'Panchi'},{position:3, name: 'Nami'}]

```
Scenario: Use math to calculate age of my cat
  Given def months = 72
  And def myCat = {name: 'Nami', age: '#(months / 12)'}
  Then assert myCat.age == 6
```

Scenario: [6:34] **Use math to calculate age of my cat**

Test 20 : * def catName = 'Neo'

Test 21 : Given def months = 72

Test 22 : And def myCat = {name: 'Nami', age: '#(months / 12)'}

Test 23 : Then assert myCat.age == 6

```
Scenario: Calculate the age of my cat without decimals
  Given def months = 73
  And def myCat = {name: 'Nami', age: '#(parseInt(months / 12))'}
  Then assert myCat.age == 6
```

```
Scenario: Calculate the age of my cat without decimals
  Given def months = 73
  And def myCat = {name: 'Nami', age: '#(~~(months / 12))'}
  Then assert myCat.age == 6
```

Test 27 : Then assert myCat.age == 6

```
Scenario: Store my cat and the last time of she asked me for food
  Given def lastRequestTime = java.lang.System.currentTimeMillis()
  And def myCat = { name: 'Itzel', lastRequestTime: '#(lastRequestTime)'}
  Then assert myCat.lastRequestTime < java.lang.System.currentTimeMillis()
```

```
Background:
  * def catName = 'Neo'

Scenario: Use function to baptise my cats
  Given def catPriest = function(arg) {return arg + ' De todos los Santos' }
  When def newbornCats =
  """
  ([
    { name: catPriest('Felix'), age: 0},
    { name: catPriest('Lucifer'), age: 0},
    { name: catPriest(catName), age: 0},
  ])
  """
  Then match newbornCats.[0].name == 'Felix De todos los Santos'
  Then match newbornCats.[1].name == 'Lucifer De todos los Santos'
  Then match newbornCats.[2].name == 'Neo De todos los Santos'
```

Scenario: [1:6] **Use function to baptise my cats**

Test 1 : * def catName = 'Neo'

Test 2 : Given def catPriest = function(arg) {return arg + ' De todos los Santos' }

Test 3 : When def newbornCats =

Test 4 : Then match newbornCats.[0].name == 'Felix De todos los Santos'

Test 5 : Then match newbornCats.[1].name == 'Lucifer De todos los Santos'

Test 6 : Then match newbornCats.[2].name == 'Neo De todos los Santos'

```
Scenario Outline: Extract a cat from file
  Given def cat = read('<catFile>')
  Then match cat contains { name: '#notnull' }
  And match cat !contains { behaviour: 'Good' }
  Examples:
    | catFile       |
    | clander.json  |
    | fistro.json   |
    | jander.json   |
```

```
Scenario Outline: [12.3:82] Extract a cat from file
Test 55 : * def catName = 'Neo'
Test 56 : Given def cat = read('jander.json')
Test 57 : Then match cat contains { name: '#notnull' }
Test 58 : And match cat !contains { behaviour: 'Good' }
```

```
{
  "name": "Clander",
  "age": 4,
  "behaviour": "Always grumpy",
  "isHappy": false
}
```

```
{
  "name": "Fistro",
  "age": 2,
  "behaviour": "Intolerably charming",
  "isHappy": true
}
```

```
{
  "name": "Jander",
  "age": 9,
  "behaviour": "Flamenco lover",
  "isHappy": true
}
```

HTTP KEYWORDS

```
Background: Store common variables used accross the scenarios
  * def port = 4567

Scenario: Create, retrieve and delete a cat
  Given url 'http://localhost:' + port + '/v1/cats'
  And request { name: 'Billie', age: 2 }
  When method PUT
  Then status 201
  And match response.parentId == 0
```

```
Background: Store common variables used accross the scenarios
  * def port = 4567

Scenario: Create, retrieve and delete a cat
  Given url 'http://localhost:' + port + '/v1/cats'
  And request { name: 'Billie', age: 2 }
  When method PUT
  Then status 201
  And match response.parentId == 0

  Given path response.name
  When method GET
  Then status 200
  And match response == { id: '#notnull', name: 'Billie', age: 2, parentId: 0}
```

```
Scenario: Create, retrieve and delete a cat
  Given url 'http://localhost:4567/v1/cats'
  And request { name: 'Satan', age: 2 }
  When method PUT
  Then status 201
  And match response == { id: '#notnull', name: 'Satan', age: 2, parentId: 0}

  Given path response.name
  When method GET
  Then status 200
  And match response == { id: '#notnull', name: 'Satan', age: 2, parentId: 0}

  Given path response.name
  When method DELETE
  Then status 204

  Given path 'Satan'
  When method GET
  Then status 404
```

```
Scenario: Create, retrieve and delete a cat
  * def catName = 'Satan'

  Given url 'http://localhost:4567/v1/cats'
  And request { name: '#(catName)', age: 2 }
  When method PUT
  Then status 201
  And match response == { id: '#notnull', name: '#(catName)', age: 2, parentId: 0}

  Given path response.name
  When method GET
  Then status 200
  And match response == { id: '#notnull', name: '#(catName)', age: 2, parentId: 0}
  And print catName

  Given path response.name
  When method DELETE
  Then status 204

  Given path '#(catName)'
  When method GET
  Then status 404
```

```
Scenario: Retrieve favourite food of cat's parent
  Given url 'http://localhost:4567/v1/cats'
  And path 'Felix/parent/food'
  When method GET
  Then status 200
  And match response.brand == 'Cosma'
```

```
Scenario: Retrieve favourite food of cat's parent
  Given url 'http://localhost:4567/v1/cats'
  And path '/Felix/parent/food'
  When method GET
  Then status 200
  And match response.brand == 'Cosma'
```

```
Scenario: Retrieve favourite food of cat's parent
  Given url 'http://localhost:4567/v1/cats'
  And path 'Felix', 'parent', 'food'
  When method GET
  Then status 200
  And match response.brand == 'Cosma'
```

```
Scenario: Create and update a cat
  Given url 'http://localhost:4567/v1/cats'
  And request { name: 'Samu', age: 2 }
  When method PUT
  Then status 201
  And match response == { id: '#notnull', name: 'Samu', age: 2, parentId: 0}

  Given request read('samuUpdate.json')
  When method POST
  Then status 200
  And match response.age == 3
```

```
{
  "name": "Samu",
  "age": 3
}
```

```
Scenario: Retrieve cat by name
  * def cat = { name: 'Samu', age: 2 }

  Given url 'http://localhost:4567/v1/cats'
  And request cat
  When method PUT
  Then status 201
  And match response == { id: '#notnull', name: 'Samu', age: 2, parentId: 0}
```

```
Scenario: Create, retrieve, update and delete a cat
  * def catName = 'Anacleto'

  Given url 'http://localhost:4567/v1/cats'
  And request { name: '#(catName)', age: 4 }
  When method PUT
  Then status 201
  And match response == { id: '#notnull', name: '#(catName)', age: 4, parentId: 0}

  Given request { id: '#notnull', name: '#(catName)', age: 1, parentId: 1}
  When method POST
  Then status 200
  And match response.age == 1
  And match response.parentId == 1
```

```
Scenario: Retrieve an cat or only its header
  * def environment = 'dev'
  * def getOrHead = (environment == 'dev' ? 'GET':'HEAD')

  Given url 'http://localhost:4567/v1/cats'
  And path 'Felix'
  When method getOrHead
  Then assert responseStatus == 404 || responseStatus == 200
```

```
Background:
  * def endpoint = 'http://localhost:4567/v1/cats'

Scenario: Retrieve filtered cats

  Given url endpoint
  And param age = 9
  When method GET
  Then status 200
  And match response == read('catsAged9.json')

  Given params { age: 9, name: 'Jacobo'}
  When method GET
  Then status 200
  And match response.[0].name == 'Jacobo'
  And match response.[0].age == 9
```

http://localhost:4567/v1/cats?age=9

http://localhost:4567/v1/cats?age=9&name=Jacobo

```
Background:
  * def testId = java.util.UUID.randomUUID()
  * def catName = 'Pozi'
  * configure headers = { 'Annoying-Header' : 'I should be always there' }

Scenario: Try to create a cat with no luck
  Given url 'http://localhost:4567/v1/cats'
  And request { name: '#(catName)' , age: 8 }
  And header Soy-El-Doctor = 'Grijando'
  When method PUT
  Then status 403

Scenario: Create a cat correctly and the delete it
  Given url 'http://localhost:4567/v1/cats'
  And request ({ name: catName , age: 8 })
  And header Test-Execution-Info = 'AWS_DEV_' + testId
  When method PUT
  Then status 201
  And match response.name == '#(catName)'

  Given path response.name
  And def myHeaders = {Adios: 'muy buenas', Gato: 'Molesto'}
  And headers myHeaders
  When method DELETE
  Then status 204
  And match header Cat-Reply == 'hasta luego Mari Carmen'
```

```
Scenario: Create another cat correctly and the delete it
  Given url 'http://localhost:4567/v1/cats'
  And request { name: Jon , age: 8 }
  When method PUT
  Then status 201
  And match response.name == 'Jon'

  Given path response.name
  And def myHeaders = {Adios: 'muy buenas', Gato: 'Molesto'}
  And headers myHeaders
  When method DELETE
  Then status 204
  And match responseHeaders['Cat-Reply'][0] == 'hasta luego Mari Carmen'
  And match responseHeaders['Cat-Reply'][1] == 'no me puedes importar menos'
```

```
Background: Remove cookies between scenarios
  * configure cookies = null

Scenario: Check that my cat will not accept specific cookie
  Given url 'http://localhost:4567/v1/cats'
  And path 'Felix'
  And cookie Food = 'RoyalCanin'
  When method GET
  Then status 406
  And assert responseCookies['Food'].value == 'RoyalFeline'

Scenario: Retrieve cat
  Given url 'http://localhost:4567/v1/cats'
  And path 'Felix'
  And cookies { Food : 'RoyalFeline', Please: 'ForgiveMe'}
  When method GET
  Then status 200
  And assert responseCookies['Forgiven'].value == 'DoNotDoItAgain'
  And assert responseTime < 1000
```

```
Scenario: Retrieve cat's picture
  Given url 'http://localhost:4567/v1/cats'
  And path 'Felix/portrait'
  When method GET
  Then status 200
  Then match responseBytes == read('Felix.jpeg')
```

VALIDACIONES REBUSCADAS

```
Given url 'http://localhost:4567/v1/cats/master'
When method GET
Then status 200

And match response == read('master.json')
And match response != { id: '#notnull', name: 'Ezequiel', age: 1}
And match response contains
"""
{
  id: #uuid,
  age: #number,
  judo: #notpresent,
  isTheBest: #boolean,
  name: #string,
  favouriteFoods: #array,
  defeats: ##null,
  optional: ##number
}
"""
```


```
{
  "id": "a9f7a56b-8d5c-455c-9d13-808461d17b91",
  "name": "Miyagi",
  "age": 99,
  "isTheBest": true,
  "defeats" : null,
  "favouriteFoods": [
    "Cosma",
    "Tigerino",
    "Purizon"
  ],
  "pupils": [
    {
      "id": 68473264,
      "name": "Inari"
    },
    {
      "id": 14,
      "name": "Axel"
    },
    {
      "id": 23434343,
      "name": "Slash"
    }
  ]
}
```

```
Given url 'http://localhost:4567/v1/cats/master'
When method GET
Then status 200
And match response.favouriteFoods contains ['Tigerino','Purizon']
And match response.favouriteFoods contains any ['Cosma','Royal Canin']
And match response.favouriteFoods contains only ['Cosma','Tigerino','Purizon']
And match response !contains { food: 'Royal Canin'}

And assert response.isTheBest
And match $.pupils.[0] == '#object'
And match $.pupils.[1].name == '#regex A.*l'
And match response contains { age: '#? _ > 0 && _ < 100' }
```


```
{
  "id": "a9f7a56b-8d5c-455c-9d13-808461d17b91",
  "name": "Miyagi",
  "age": 99,
  "isTheBest": true,
  "defeats" : null,
  "favouriteFoods": [
    "Cosma",
    "Tigerino",
    "Purizon"
  ],
  "pupils": [
    {
      "id": 68473264,
      "name": "Inari"
    },
    {
      "id": 14,
      "name": "Axel"
    },
    {
      "id": 23434343,
      "name": "Slash"
    }
  ]
}
```

TRUCOS PARA REUTILIZAR

karate-config.js

```
1  function () {
2    karate.configure('connectTimeout', 5000);
3    karate.configure('readTimeout', 5000);
4  }
```

```
function () {
    var URL = 'http://localhost:4567'
    karate.configure('connectTimeout', 5000);
    karate.configure('readTimeout', 50000);
    karate.log("API URL: " + URL)
    return {
       baseURL: URL
    };
}
```

```
Scenario: Retrieve cat's picture
  Given url baseURL
  And path 'v1', 'cats', 'Felix', 'portrait'
  When method GET
  Then status 200
  Then match responseBytes == read('classpath:fixtures/Felix.jpeg')
```

```
@Common
Feature: Intended to be called from another features

  Scenario: Retrieve the token
    Given url baseURL
    And path 'auth'
    And form field username =  'arcones@karate-meetup.com',
    And form field password = 'CambiameDeUnaVez123'
    And method GET
    * def token = response.accessToken
    And print response.accessToken
    And status 200
```

```
@CallAnotherFeature
Feature: Show how to call common feature

  Scenario: Use the token retrieved by another feature
    * def callForToken = call read('classpath:com/zooplus/auth/Common.feature')
    * def myToken = callForToken.response.accessToken
    And assert myToken == 'paco-paco-paco-que-mi-paco-paco-paco-paco'
```

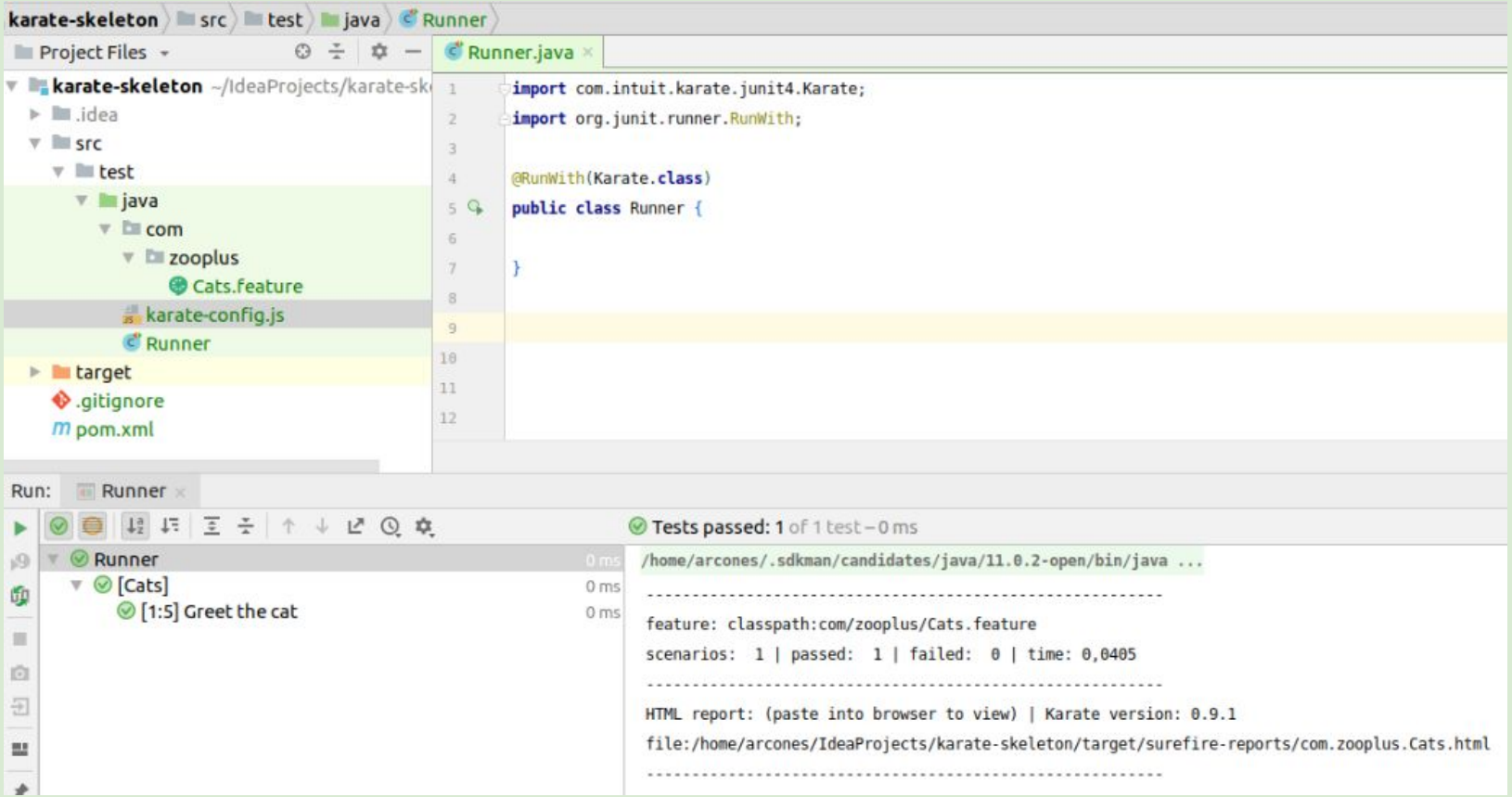
INTEGRACIÓN CON KARATE

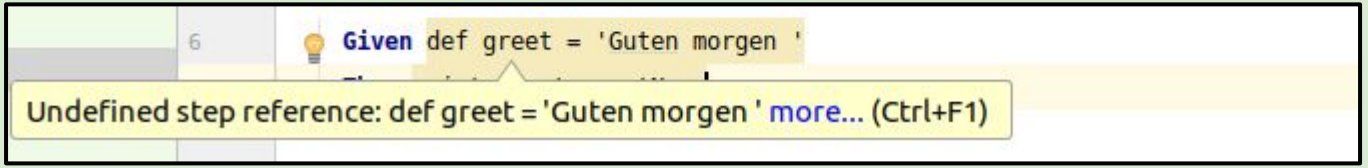

```
➜  ~ mvn archetype:generate \
-DarchetypeGroupId=com.intuit.karate \
-DarchetypeArtifactId=karate-archetype \
-DarchetypeVersion=0.9.1 \
-DgroupId=com.zooplus \
-DartifactId=karate-skeleton
```

```
➜  karate-skeleton git:(master) ✗ tree
.
├── pom.xml
└── src
    └── test
        └── java
            ├── com
            │   └── zooplus
            │       └── Cats.feature
            ├── karate-config.js
            └── Runner.java
```

```java
import com.intuit.karate.junit4.Karate;
import org.junit.runner.RunWith;

@RunWith(Karate.class)
public class Runner {

}
```

```
→ karate-skeleton git:(master) ✗ mvn test -Dtest=Runner
-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running Runner
19:47:40.047 [main] INFO com.intuit.karate - [print] Guten morgen Neo
-------------------------------------------------------
feature: classpath:com/zooplus/Cats.feature
scenarios:  1 | passed:  1 | failed:  0 | time: 0,0253
-------------------------------------------------------
HTML report: (paste into browser to view) | Karate version: 0.9.1
file:/home/arcones/IdeaProjects/karate-skeleton/target/surefire-reports/com.zooplus.Cats.html
-------------------------------------------------------

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.978 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  2.291 s
[INFO] Finished at: 2019-02-23T19:47:40+01:00
[INFO] ------------------------------------------------------------------------
```


karate-skeleton
src
test
java
Runner
Project Files
karate-skeleton ~/IdeaProjects/karate-sk
.idea
src
test
java
com
zooplus
Cats.feature
karate-config.js
Runner
target
.gitignore
pom.xml
Runner.java
1 import com.intuit.karate.junit4.Karate;
2 import org.junit.runner.RunWith;
3
4 @RunWith(Karate.class)
5 public class Runner {
6
7 }
8
9
10
11
12
Run:
Runner
Tests passed: 1 of 1 test – 0 ms
Runner 0 ms
[Cats] 0 ms
[1:5] Greet the cat 0 ms
/home/arcones/.sdkman/candidates/java/11.0.2-open/bin/java ...
feature: classpath:com/zooplus/Cats.feature
scenarios: 1 | passed: 1 | failed: 0 | time: 0,0405
HTML report: (paste into browser to view) | Karate version: 0.9.1
file:/home/arcones/IdeaProjects/karate-skeleton/target/surefire-reports/com.zooplus.Cats.html


6
Given def greet = 'Guten morgen '
Undefined step reference: def greet = 'Guten morgen ' more... (Ctrl+F1)

Test Suite Navigation

of failed tests: 0/17
(0.00%)

# of skipped tests: 0/17
(0.00%)

# of passed tests: 17/17
(100.00%)

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	

Scenario: [1:3] **Create, retrieve and delete a cat**

Test 1 : Given url 'http://localhost:4567/v1/cats/master'	**0.003571**
Test 2 : When method GET	**0.006683**
Test 3 : Then status 200	0.000466
Test 4 : And match response == read('master.json')	0.003951
Test 5 : And match response != { id: '#notnull', name: 'Ezequiel', age: 1}	0.000339
Test 6 : And match response contains	**0.000938**

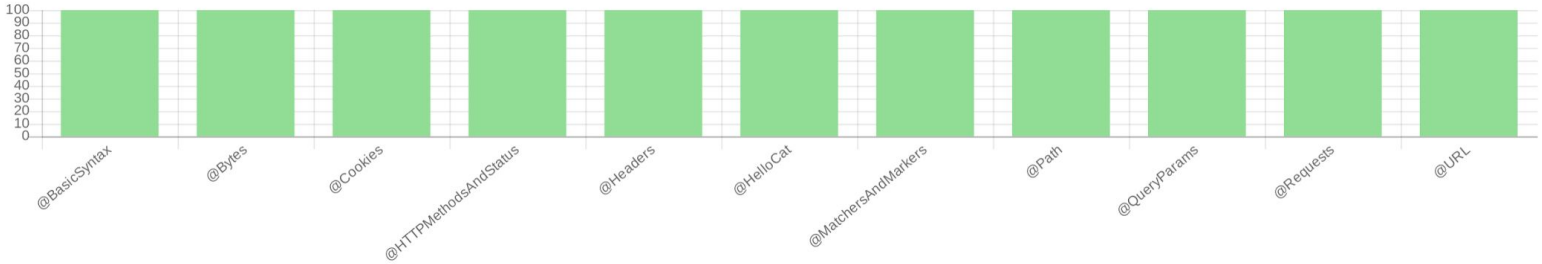
```
{
  id: #uuid,
  age: #number,
  judo: #notpresent,
  isTheBest: #boolean,
  name: #string,
  favouriteFoods: #array,
  defeats: ##null,
  optional: ##number
}
```

Test 7 : Given url 'http://localhost:4567/v1/cats/master'	0.000132
Test 8 : When method GET	**0.00447**
Test 9 : Then status 200	0.000021
Test 10 : And match response.favouriteFoods contains ['Tigerino','Purizon']	0.000641
Test 11 : And match response.favouriteFoods contains any ['Cosma','Royal Canin']	0.000227
Test 12 : And match response.favouriteFoods contains only ['Cosma','Tigerino','Purizon']	0.000749
Test 13 : And match response !contains { food: 'Royal Canin'}	0.000197
Test 14 : And assert response.isTheBest	0.004194
Test 15 : And match $.pupils.[0] == '#object'	0.004059
Test 16 : And match $.pupils.[1].name == '#regex A.*l'	0.002902
Test 17 : And match response contains { age: '#? _ > 0 && _ < 100' }	0.008526

Project	Date
CatsApi	24 feb. 2019, 12:43

Tags Statistics

The following graph shows passing and failing statistics for tags



	Steps						Scenarios			Features	
Tag	Passed	Failed	Skipped	Pending	Undefined	Total	Passed	Failed	Total	Duration	Status
@BasicSyntax	52	0	0	0	0	52	18	0	18	0.397	Passed
@Bytes	5	0									
@Cookies	13	0									
@HTTPMethodsAndStatus	27	0									

Tag Report

	Steps						Scenarios			Features	
Tag	Passed	Failed	Skipped	Pending	Undefined	Total	Passed	Failed	Total	Duration	Status
@Bytes	5	0	0	0	0	5	1	0	1	0.033	Passed

Feature: com/zooplus/cats/http/keywords/Bytes.feature

Scenario Retrieve cat's picture 0.033

Steps

Given url 'http://localhost:4567/v1/cats' 0.000

Doc string

And path 'Felix/portrait' 0.004

When method GET 0.022

Doc string

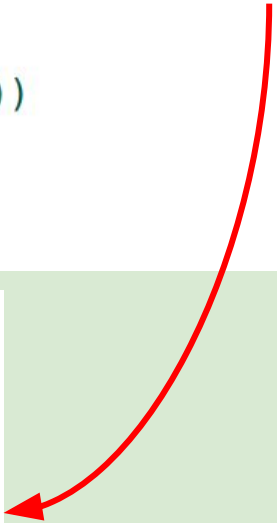
Then status 200 0.000

Then match responseBytes == read('Felix.jpeg') 0.007

TEST DE CARGA

```
class LoadTestRunner extends Simulation {

  val feature: ScenarioBuilder = scenario( scenarioName = "Mad german boy retrieving cats")
    .exec(karateFeature( name = "src/test/java/com/zooplus/cats/load/Load.feature"))

  setUp(feature.inject(rampUsers( users = 100) over (1 seconds)))
    .assertions(global.failedRequests.count.is( value = 0))
}
```

```
Feature: Intended to be run from Gatling suite

  Scenario: Retrieve a cat
    Given url 'http://localhost:4567/v1/cats'
    And path 'Perry'
    And method GET
    Then status 200
```



COMPARATIVA CON CUCUMBER

LEGIBILIDAD Y ÁMBITO DE LAS FEATURES

LEGIBILIDAD Y ÁMBITO DE LAS FEATURES

DOCUMENTACIÓN Y COMUNIDAD

DOCUMENTACIÓN Y COMUNIDAD

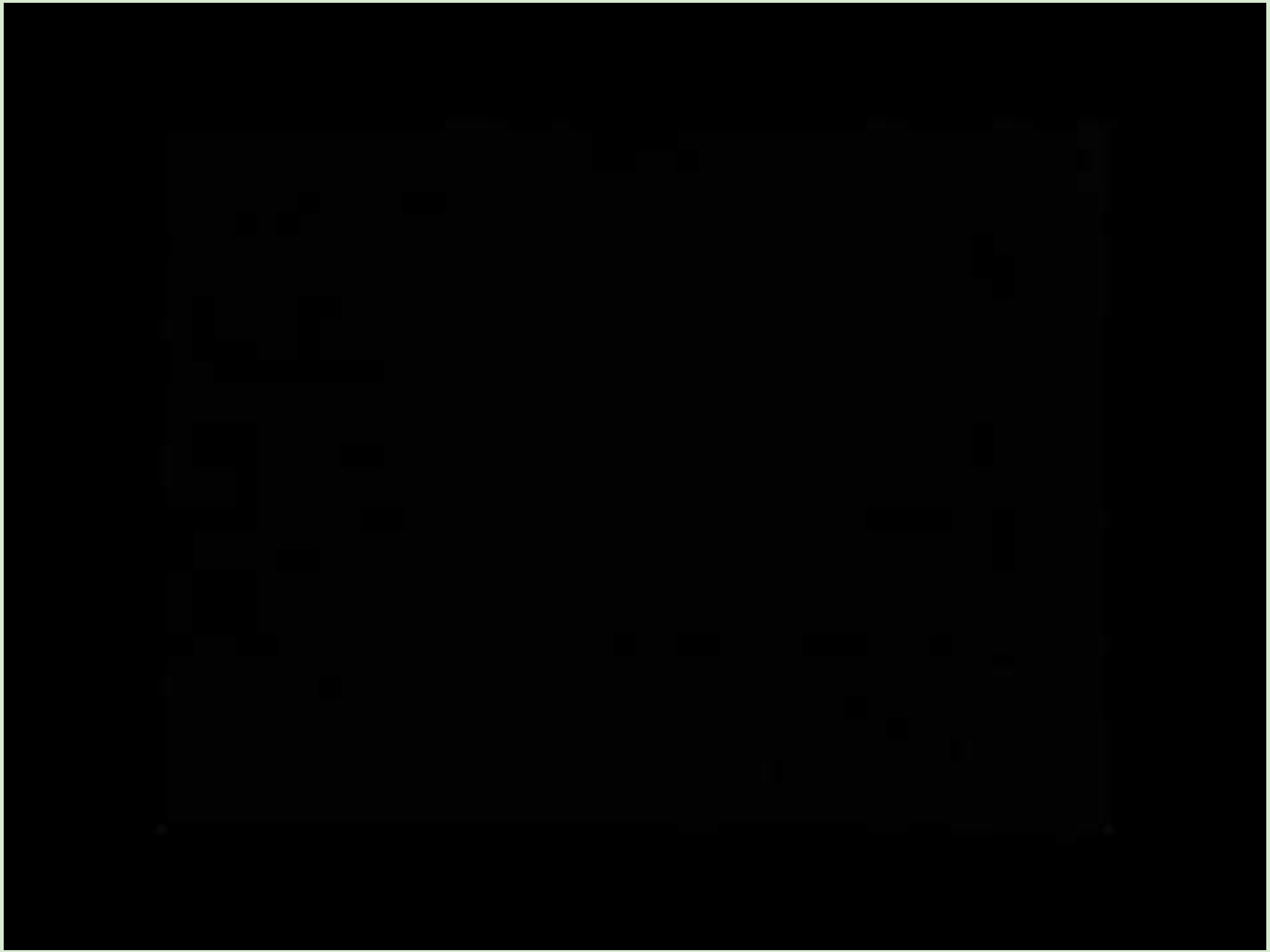
CANTIDAD DE CÓDIGO A MANTENER

CANTIDAD DE CÓDIGO A MANTENER

CONCLUSIONES


PERO ESTO YA NO ES BDD...
ME LA PELA PEPINILLOS
imgflip.com

Y DE POSTRE...



REFERENCIAS Y AGRADECIMIENTOS

- Proyecto Karate en GitHub
- Artículos sobre testing del blog de Martin Fowler
- Libro “Building Microservices” de Sam Newman
- Interesante entrada de blog para estructurar el proyecto de Karate
- Buen artículo acerca de mocking con karate-netty
- Karate cheatsheet muy praćtica
- Comparativa extensa de frameworks para hacer test de servicio, Karate entre ellos
- Iconos de flaticon
- Haters de Stackoverflow. Sin ellos no sería lo mismo
- **Gente de Zooplus, con especial mención a mi equipo, los QAs**