BACALLAO PROPERTY-BASED TESTING

Rodrigo López-Romero Guijarro

@rodriguetelrg

iOS by the sea, June 2019

NETWORKING !
LET'S INTRODUCE EACH OTHER !

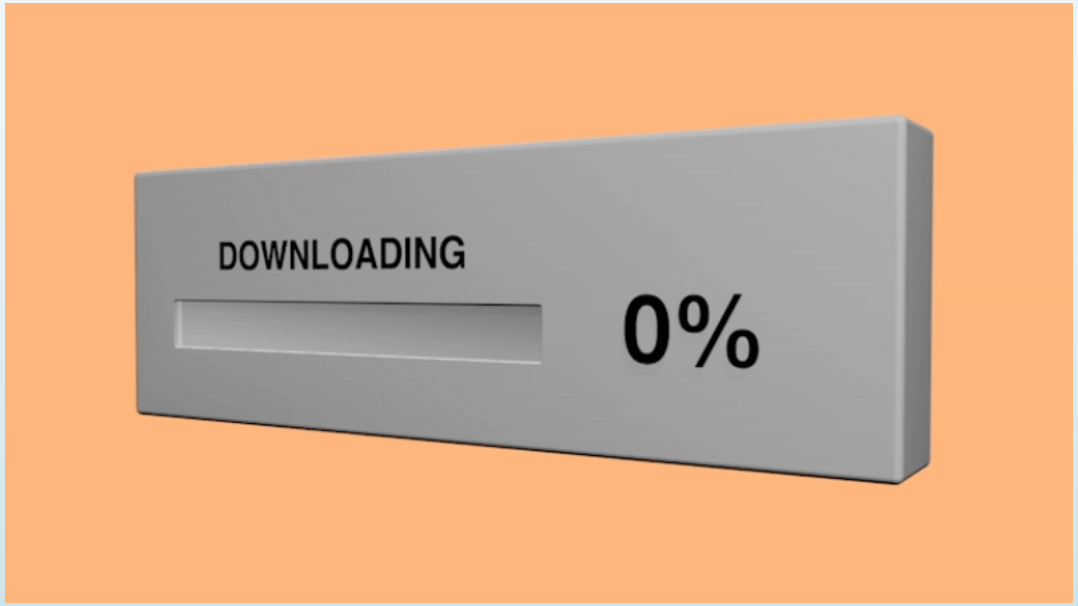
- Who am I ?
- What's my main role ?
- Where do I work ?
- Free time ?

AGENDA FOR TODAY!

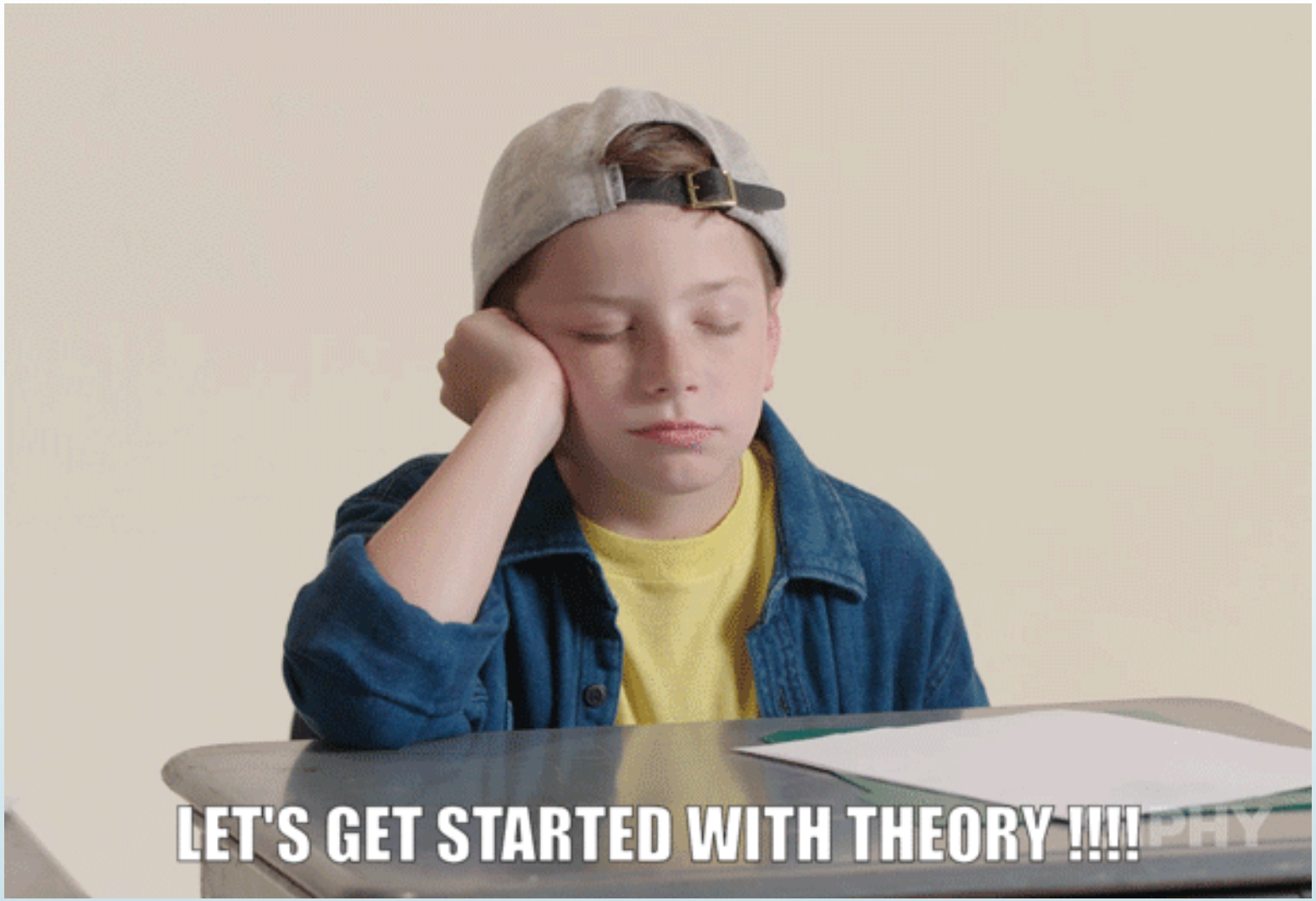
- A LITTLE BIT OF THEORY
- SUM PROPERTIES
- PLAYING WITH SWIFTCHECK
- REAL PROJECT: BACALLAO HOTEL BUFFET
- QUIZ TIME !!

FOR NOW … 1 MINUTE TO CHECKOUT!

https://github.com/RLRG/SwiftAveiro_PropertyBasedTesting.git



PROPERTY-BASED TESTING



PROPERTY-BASED TESTING

WHAT IS IT THEN ?

WHAT IS THE PURPOSE ?

PROPERTY-BASED TESTING

ALSO KNOWN AS "GENERATIVE TESTING"

PROPERTY-BASED TESTING

- LITTLE EFFORT
- DIFFERENT INPUTS EACH TIME
- IN-DEPTH UNDERSTANDING

PROPERTY-BASED TESTING

HOW DO WE PROGRAM THOSE BEAUTIFUL TESTS ?

PROPERTY-BASED TESTING

WE RELY ON THE PROPERTIES OF A SOFTWARE ITEM - SUT.

BUT WHAT'S A PROPERTY THEN ?

PROPERTY-BASED TESTING

- A PROPERTY CAN BE SOMETHING LIKE:

FOR ALL (X,Y, ...)

SUCH AS PRECONDITION (X,Y, ...) HOLDS

PROPERTY (X,Y, ...) IS TRUE

- OK... I STILL DON'T GET IT ? CAN WE SEE AN EXAMPLE ?

PROPERTY-BASED TESTING

- EXAMPLE:

FOR ALL (X,Y, ...)

SUCH AS PRECONDITION (X,Y, ...) HOLDS

PROPERTY (X,Y, ...) IS TRUE

FOR ALL STRINGS (A,B,C)

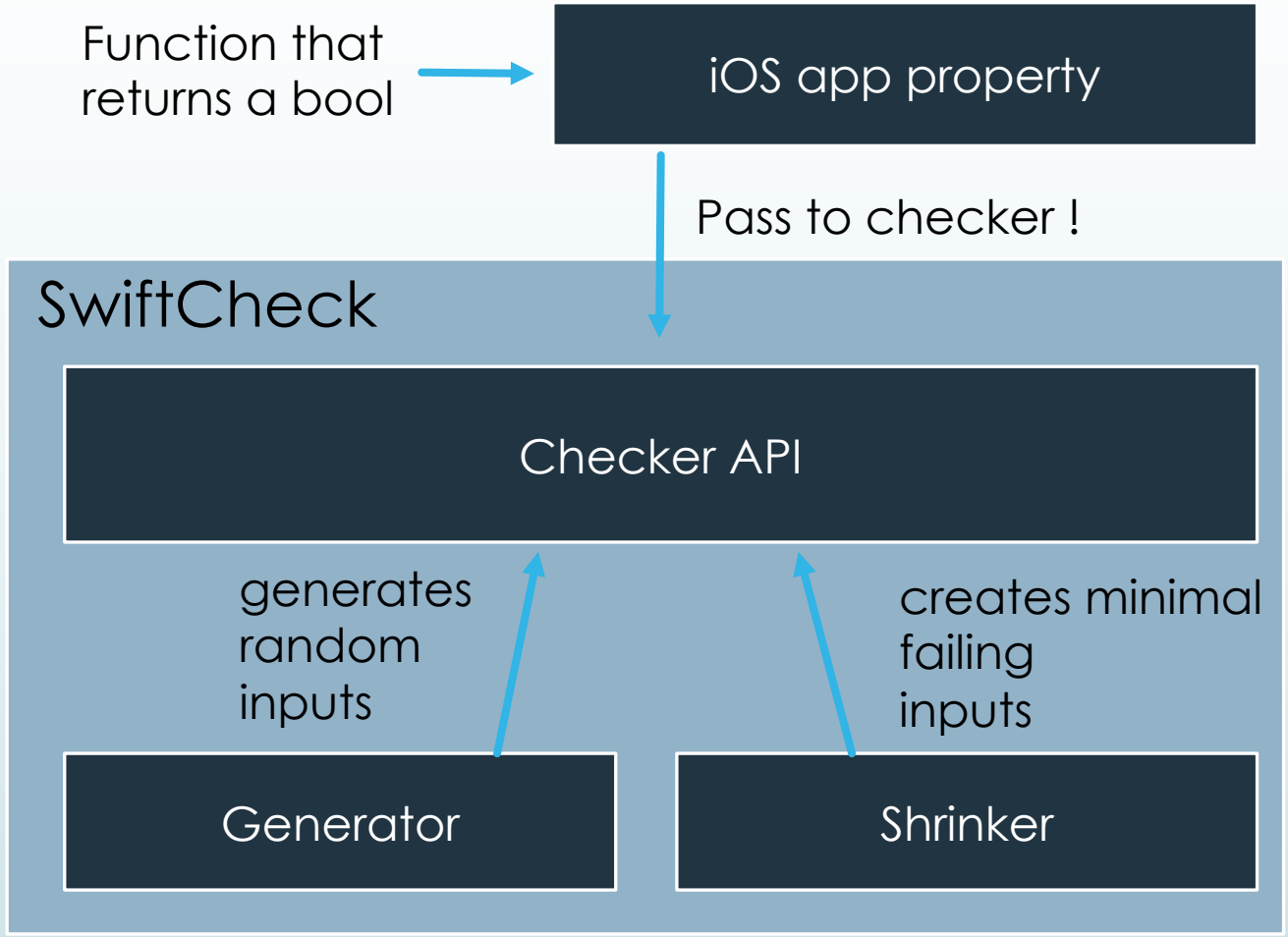
THE CONCATENATION OF A,B AND C HOLDS

"IT ALWAYS CONTAINS B" IS TRUE

PROPERTY-BASED TESTING. COMMON PROPERTIES

- MATH PROPERTIES:
  - ASSOCIATIVE
  - COMMUTATIVE
  - DISTRIBUTIVE
  - …
- array = array.reversed().reversed()
- array.length = array.sorted().length
- Function with two array parameters, the output is a merge of those two arrays.
- …

PROPERTY-BASED TESTING - SWIFTCHECK



PROPERTY-BASED TESTING - BENEFITS

COVER THE SCOPE OF ALL POSSIBLE INPUTS

PROPERTY-BASED TESTING - BENEFITS

SHRINK THE INPUT IN CASE OF FAILURE

"xkxcjbddgjhdhj@knfjfnfkfkfj109288383" →
"xhj@knfjf383" → "hj@kn" → "@"

PROPERTY-BASED TESTING - BENEFITS

REPRODUCIBLE AND REPLAYABLE

```
Proposition: make a test fail in order to see how shrinking works
Falsifiable (after 67 tests):
59
*** Passed 66 tests
.
/Users/lopezror/Desktop/SwiftAveiro_PropertyBasedTesting/SwiftAveiroTheHotelBuffet_SolutionComplete/
    SwiftAveiroTheHotelBuffetTests/2. Properties Exercises/PropertiesExercises.swift:38: error:
    -[SwiftAveiroTheHotelBuffetTests.PropertiesExercises testShrinkByMakingATestFail] : failed -
    Falsifiable; Replay with 1657277901 8907 and size 66
```

PROPERTY-BASED TESTING - BENEFITS

CUSTOM GENERATORS ARE ALLOWED

App with classes: Client, Waiter and Hotel

We can generate random Clients for a test !

PROPERTY-BASED TESTING - BENEFITS

FIND DISCREPANCIES

SPECIFICATION VS IMPLEMENTATION

PROPERTY-BASED TESTING

DOES IT REPLACE OTHER TESTING TECHNIQUES SUCH AS UNIT TESTING ?

PROPERTY-BASED TESTING

NO.

IT IS AN ADDITIONAL LAYER!

ENOUGH THEORY MAN !!

LET'S PROGRAM TOGETHER !

- SUM PROPERTIES
  - Commutative property:
    - 4 + 2 = 2 + 4
  - Associative property:
    - (2 + 3) + 4 = 2 + (3 + 4)
  - Neutral element:
    - 5 + 0 = 5
  - Distributive property:
    - 4 * (6 + 3) = (4 * 6) + (4 * 3)

LET'S PLAY WITH SWIFTCHECK !

- EXERCISE 1:
  - Add logs to see what is being tested.
- EXERCISE 2:
  - Modify default number of executions.
- EXERCISE 3:
  - Analyze shrinking: Make a test fail (example: "any INT is < 50") in order to see in the logs how shrinking works.
- EXERCISE 4:
  - Create a custom generator which generate Ints between 25 and 50.

SOLUTION

FILE:

SwiftAveiroTheHotelBuffet_SolutionExercises.dmg

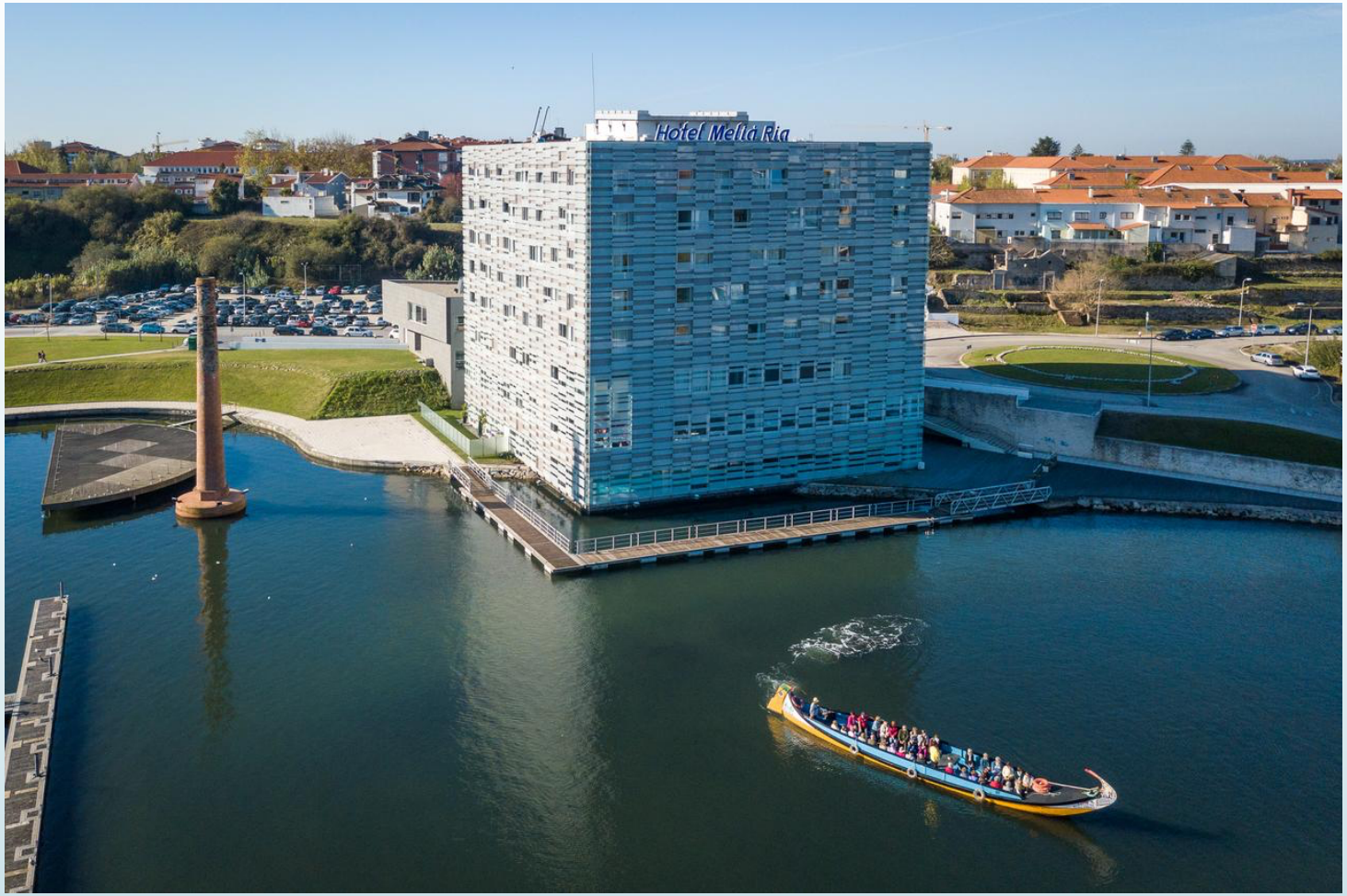
PASSWORD:

"PropertyBasedTesting"

5-10 MINUTES TO CHECK THE SOLUTION GUYS !!

REAL SIMPLE PROJECT: BACALLAO HOTEL BUFFET

REAL SIMPLE PROJECT: BACALLAO HOTEL BUFFET



REAL SIMPLE PROJECT: BACALLAO HOTEL BUFFET

Friends

CHARLIE

REAL SIMPLE PROJECT: BACALLAO HOTEL BUFFET

Friends

CHARLIE

THOMAS

REAL SIMPLE PROJECT: BACALLAO HOTEL BUFFET

Friends

CHARLIE

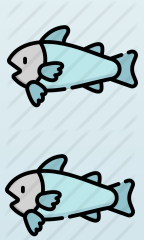
THOMAS

DAVID

REAL SIMPLE PROJECT: BACALLAO HOTEL BUFFET

Friends

CHARLIE



THOMAS

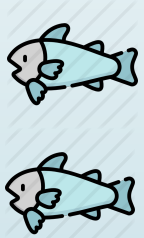
DAVID

ALICE

REAL SIMPLE PROJECT: BACALLAO HOTEL BUFFET

Friends

CHARLIE

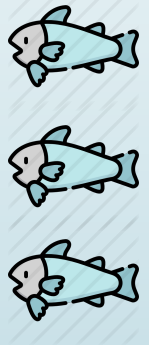


THOMAS

DAVID

ALICE

EMILY



REAL SIMPLE PROJECT: BACALLAO HOTEL BUFFET

Waiters

MARTINA

+ 10

REAL SIMPLE PROJECT: BACALLAO HOTEL BUFFET

Waiters

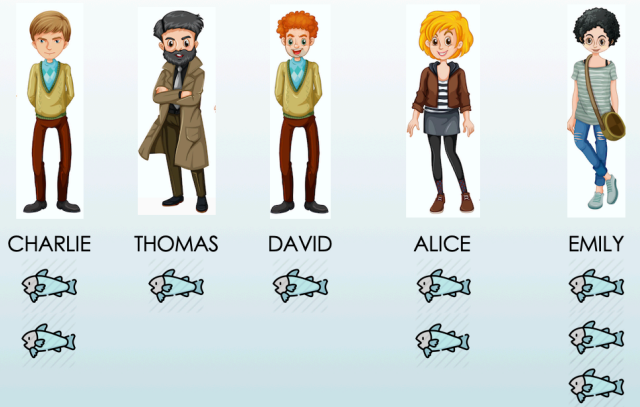
MARTINA

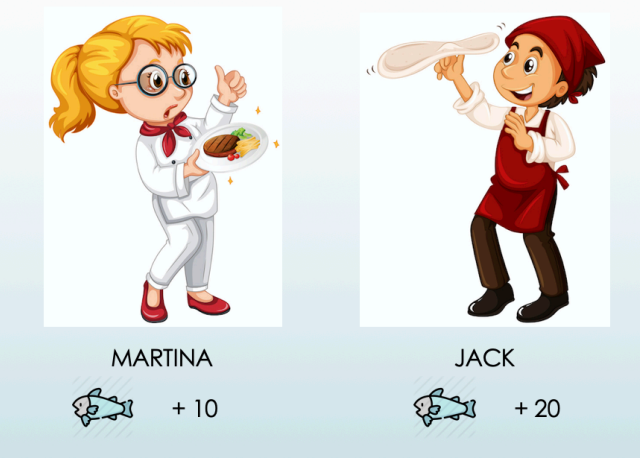
+ 10

JACK

+ 20

REAL SIMPLE PROJECT: BACALLAO HOTEL BUFFET





LOGIC SUMMARY:

Friend

→ eat X pieces or the remaining ones.

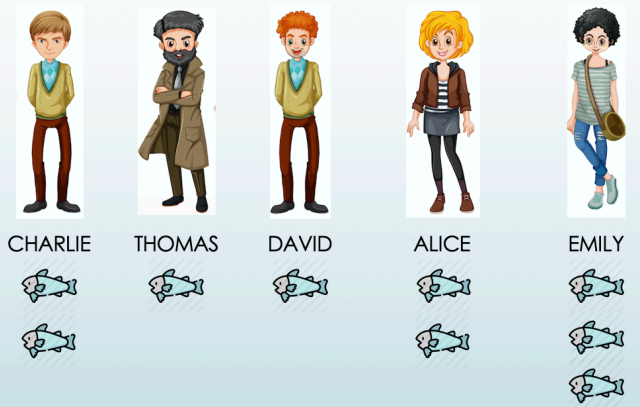
If < 2 pieces of Bacallao

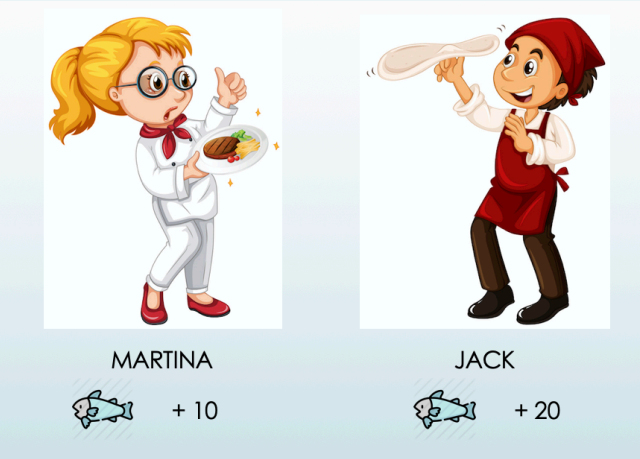
→ Tell random waiter

Waiter

→ Add X Bacallao pieces

REAL SIMPLE PROJECT: BACALLAO HOTEL BUFFET





OUR OBJECTIVE:

- ClientTests
- WaiterTests
- HotelBuffetTests

HINT:

- THINK ABOUT THE PROPERTIES EACH CLASS HAS FIRST !!
- Example property for client:
  - "Number of bacallao rations to consume cannot be negative"

REAL SIMPLE PROJECT: BACALLAO HOTEL BUFFET

Client / Waiter properties:

- A string name is assigned in initialization phase

```
func testNameIsAssignedToClient() {
    property("A string name is assigned in initialization phase") <- forAll { (name: String) in
        let client = Client(name: name, bacallaoUnitsToEat: self.testBacallaoRationsToEat)
        return client.name == name
    }
}
```

- Number of bacallao rations to consume cannot be negative
- Number of bacallao rations is assigned if is positive or zero

REAL SIMPLE PROJECT: BACALLAO HOTEL BUFFET

HotelBuffet properties:

- Number of bacallao rations in the buffet at the beginning cannot be negative

```
func testHotelBuffetStartsWithNonNegativeBacallaoRations() {
    property("Number of bacallao rations to add cannot be negative")
        <- forAll { (integerToTest: Int) in
            let hotelBuffet = HotelBuffet(bacallaoRationsInBuffet: integerToTest)
            return hotelBuffet.bacallaoRationsInBuffet >= 0
    }
}
```

- If one or many clients eat bacallao, the number of bacallao rations left cannot be lower than two
- If < 2 bacalao units → bacalao warning generated
- If > 2 bacalao units → bacalao warning is not generated
- ….

SOLUTION

FILE:

SwiftAveiroTheHotelBuffet_SolutionComplete.dmg

PASSWORD:

"SwiftAveiro2019"

5-10 MINUTES TO CHECK THE SOLUTION GUYS !!

SUM UP ! NEXT STEPS

- PROPERTY-BASED TESTING CAN BE APPLIED FOR REAL USE CASES.
- REAL USE CASES EXAMPLES:
  - RETAIL APP - CHECKOUT DIFFERENT PRODUCTS RANDOMLY.
  - LOGIN WITH USERS WITH DIFFERENT PERMISSIONS TO AN APP RANDOMLY AND TEST ANY SPECIFIC FEATURE.
  - ...
- MIXING PROPERTY BASED TESTING
  - INTEGRATION TESTS
  - UI TESTS
  - UNIT TESTS
  - ...

KAHOOT TIME !

Download the app!

or… https://kahoot.com/

SOME USEFUL REFERENCES

- "The lazy programmer's guide to writing 1000's of tests. An introduction to property based testing" https://es.slideshare.net/ScottWlaschin/an-introduction-to-property-based-testing
- John Hughes "Don't write tests!" https://www.youtube.com/watch?v=hXnS_Xjwk2Y
- https://dev.to/jdsteinhauser/intro-to-property-based-testing-2cj8
- Some icons were taken from www.freepik.es (brgfx)

THANK YOU ! QUESTIONS ?

@rodriguetelrg

Rodrigo López-Romero Guijarro

https://github.com/RLRG

Feedback is more than welcome ! ☺
https://es.surveymonkey.com/r/QGCR5JK