



ARQUITECTURA DEL SOFTWARE

2025-26

Pablo González

Jose Labra

Diego Martín

Celia Melendi



Escuela de
Ingeniería
Informática



Universidad de Oviedo

Laboratorio 8

BDD

Pruebas de aceptación

Pruebas de aceptación y BDD

Pruebas que pueden ejecutarse con el cliente

Si pasan, el producto es aceptado

Behaviour-Driven-Development (BDD)

Variante de TDD (test driven development)

Pruebas basadas en comportamiento

Relacionado con ATDD (Acceptance-Test Driven Development)

Comportamiento = Historias de usuario

También conocido como: Especificación por ejemplos

Objetivo: Especificaciones ejecutables

Algunas herramientas:

cucumber, jBehave, concordion

BDD - Historias de usuario

Breves (en teoría se escriben en tarjetas)

Deben ser legibles (y aprobadas) por expertos de dominio (negocio)

Otros atributos deseables:

- Independientes (sin interrelaciones fuertes)

- Negociables (sin detalles concretos)

- Con valor para el usuario

- Estimables (para incluirlas en Sprints)

- Pequeñas (en otro caso considerar dividir las)

- Se pueden validar (automatizar)

BDD - Historias de usuario

Característica: *Título (descripción de historia)*

Como [rol]

Quiero [característica]

Para [beneficio]

COMO <rol>

QUIERO <evento>

PARA <funcionalidad>

Varios escenarios

Dado [Contexto]

Y [un poco más de contexto]

Cuando [Suceso]

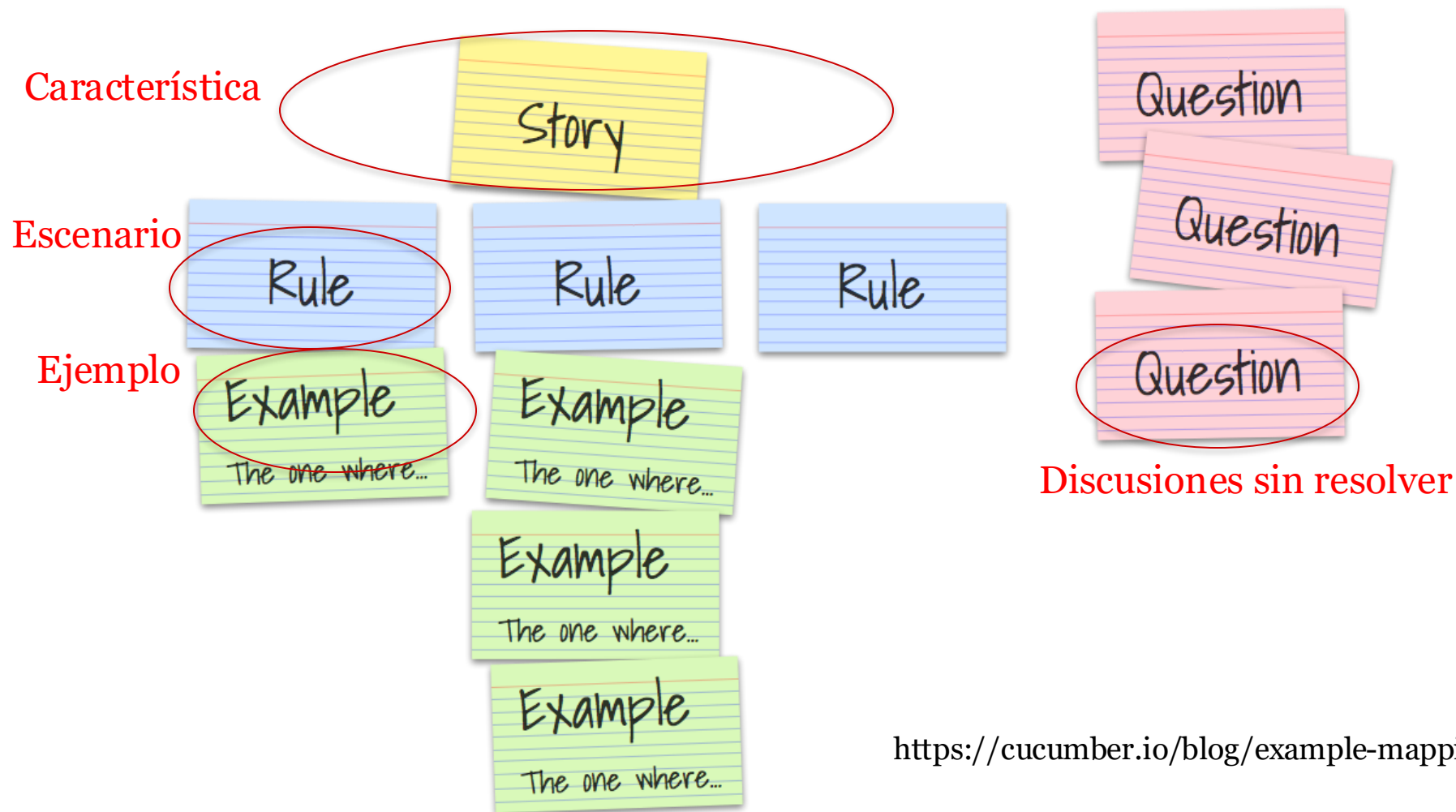
Entonces [Resultado]

Y [otro resultado]

AS ADMIN USER
I WANT TO LOCK A USER ACCOUNT
SO THAT I CAN PREVENT ABUSE OF THE SITE

SCENARIO ADMIN LOCKS A USER ACCOUNT
GIVEN THE USER IS LOGGED ON AS ADMIN
AND THE USER IS ON THE ADMIN PAGE
AND THE TARGET ACCOUNT EXISTS
AND THE TARGET ACCOUNT IS A USER LEVEL ACCOUNT
AND THE TARGET ACCOUNT IS UNLOCKED
WHEN THE USER CLICKS LOCK ACCOUNT
AND THE USER CLICKS CONFIRM
THEN THE TARGET ACCOUNT IS LOCKED
AND THE ADMIN USER RECEIVES A SUMMARY
AND THE USER OF THE TARGET ACCOUNT RECEIVES AN EMAIL

BDD - Example Mapping





BDD con Cucumber

- Cucumber = desarrollado en Ruby (2008)
 - Rspec (Ruby), jbehave (Java)
- Se basa en Lenguaje Gherkin (lenguaje interno para definir las historias)
<https://cucumber.io/docs/gherkin/>
Puede usarse en diferentes idiomas (asturiano, ...)
- Historias de usuarios enlazadas con definiciones de los pasos
 - Definiciones de pasos se ejecutan para validar las historias de usuario



BDD con Cucumber

Característica: Describe una funcionalidad

Una funcionalidad puede tener varios escenarios

Escenario:

Comportamiento del sistema en un contexto determinado

Given: Contexto

When: Interacción con el sistema

Then: Comprueba el resultado

Examples: Datos concretos



BDD con Cucumber

Referencias:

Java: cucumber-jvm

<https://github.com/cucumber/cucumber-jvm>

Eclipse support:

<http://cucumber.github.io/cucumber-eclipse/>

Visual Studio Code: Cucumber (Gherkin) Full Support

<https://marketplace.visualstudio.com/items?itemName=alexkrechik.cucumberautocomplete>

Ejemplo BDD

- Crear una historia de usuario (paso a paso)
 - Instalar Cucumber
 - Escribir el primer escenario en Gherkin
 - Escribir las definiciones de los pasos en el lenguaje de programación escogido
 - Ejecutar Cucumber

Ejemplo BDD con Cucumber

- Depende del lenguaje de programación o entorno
 - Java/Javascript/Python/...
 - Instalación en: <https://cucumber.io/>
- React ejemplo: https://github.com/Arquisoft/yovi_0_cucumber: Módulo que permite crear historias con Gherkin .
 - `$ npm install --save-dev cucumber`
 - [playwright](#): Módulo que permitirá ejecutar los test en un navegador por defecto.
 - Hay otras alternativas como puppeteer.
 - `$ npm install --save-dev playwright`
 - [Start-server-and-test](#): Module for launching the infrastructure
 - `$ npm install --save-dev start-server-and-test`

Ejemplo BDD

- Historia de usuario usando Node.js

Feature: Register. Validate the register form

Característica

Scenario: The user is not registered in the site

Given the register page is open

When I enter "Alice" as the user name and submit

I should see a welcome message containing "Hello Alice"

Escenario

webapp/test/e2e/features/register.feature

Ejemplo BDD

Webapp/src/test/e2e/steps/register.steps.js

```
import { Given, When, Then } from '@cucumber/cucumber'
import assert from 'assert'

Given('the register page is open', async function () {
  const page = this.page
  if (!page) throw new Error('Page not initialized')
  await page.goto('http://localhost:5173')
})

When('I enter {string} as the username and submit', async function (username) {
  const page = this.page
  if (!page) throw new Error('Page not initialized')
  await page.fill('#username', username)
  await page.click('.submit-button')
})

Then('I should see a welcome message containing {string}', async function (expected) {
  const page = this.page
  if (!page) throw new Error('Page not initialized')
  await page.waitForSelector('.success-message', { timeout: 5000 })
  const text = await page.textContent('.success-message')
  assert.ok(text && text.includes(expected), `Expected success message to include "${expected}", got: "${text}"`)
})
```

Ejemplo BDD [Configuración navegador]

- webapp/test/e2e/support.mjs
 - Configura cómo iniciar el navegador para realizar las pruebas
 - También se puede configurar con otros navegadores.
 - Usamos **headless=true** para ejecutar las pruebas en el sistema CI, pero podemos cambiarlo a falso para ejecutarlas localmente
 - El parámetro **slowMo** es útil para ralentizar las pruebas y ver qué está sucediendo.

```
Before(async function () {  
  // Allow turning off headless mode and enabling slow motion/devtools via env vars  
  const headless = true  
  const slowMo = 0  
  const devtools = false  
  
  this.browser = await chromium.launch({ headless, slowMo, devtools })  
  this.page = await this.browser.newPage()  
})
```

Ejemplo BDD [Configuración - lanzamiento del sistema]

- webapp/package.json
 - Configura cómo lanzar el sistema
 - Para probar esta aplicación necesitamos restapi y webapp
 - Vamos a utilizar la librería start-server-and-test Esta librería acepta pares de parámetros. (**clave valor**) o **comandos de npm explicitos**.
 - Para ejecutar los tests e2e hay que ejecutar el comando **npm run test:e2e**

```
"test:e2e:run": "cucumber-js --import test/e2e/support/setup.mjs --import test/e2e/steps/**/*.mjs test/e2e/features --format progress",  
"test:e2e": "start-server-and-test start:all http://localhost:5173 test:e2e:run",  
"test:e2e:install-browsers": "npx playwright install",  
"start:all": "concurrently \"npm run dev\" \"npm --prefix ../users start\""
```

Ejemplo cucumber + selenium + Java (Spring-boot) de años anteriores

<https://github.com/arquisoft/votingSystem0>



Pruebas para navegadores

- Automatización con Navegadores
 - <https://cucumber.io/docs/guides/browser-automation/>
 - Otros herramientas
 - Selenium WebDriver - <http://docs.seleniumhq.org/>
 - Capybara - <http://teamcapybara.github.io/capybara/>
 - Watir - <https://watir.com/>
 - Serenity - <http://serenity-bdd.info>

Selenium



Selenium IDE: permite registrar acciones

Firefox y Chrome plugins

Genera código para ejecutar esas acciones.

Bibliografía y enlaces de interés

User Story Mapping by Jeff Patton

User Story Mapping: Discover the Whole Story, Build the Right Product, 1st Edition

<https://www.amazon.com/User-Story-Mapping-Discover-Product/dp/1491904909>

Historias de Usuario

Scrum. Historias de Usuario (Fernando Llopis, Universidad de Alicante)

<https://fernandollopis.dlsi.ua.es/?p=39>

User stories with Gherkin and Cucumber (Michael Williams)

<https://medium.com/@mwwi/story-writing-with-gherkin-and-cucumber-1878124c284c>

Cucumber 10 minutes tutorial (JS)

<https://docs.cucumber.io/guides/10-minute-tutorial/>

Pruebas basadas en navegador

Automated UI Testing with Selenium and JavaScript

<https://itnext.io/automated-ui-testing-with-selenium-and-javascript-90bbe7ca13a3>