Coding UI Automated Tests

Diego Bosin & Juan Berdugo

Quienes somos

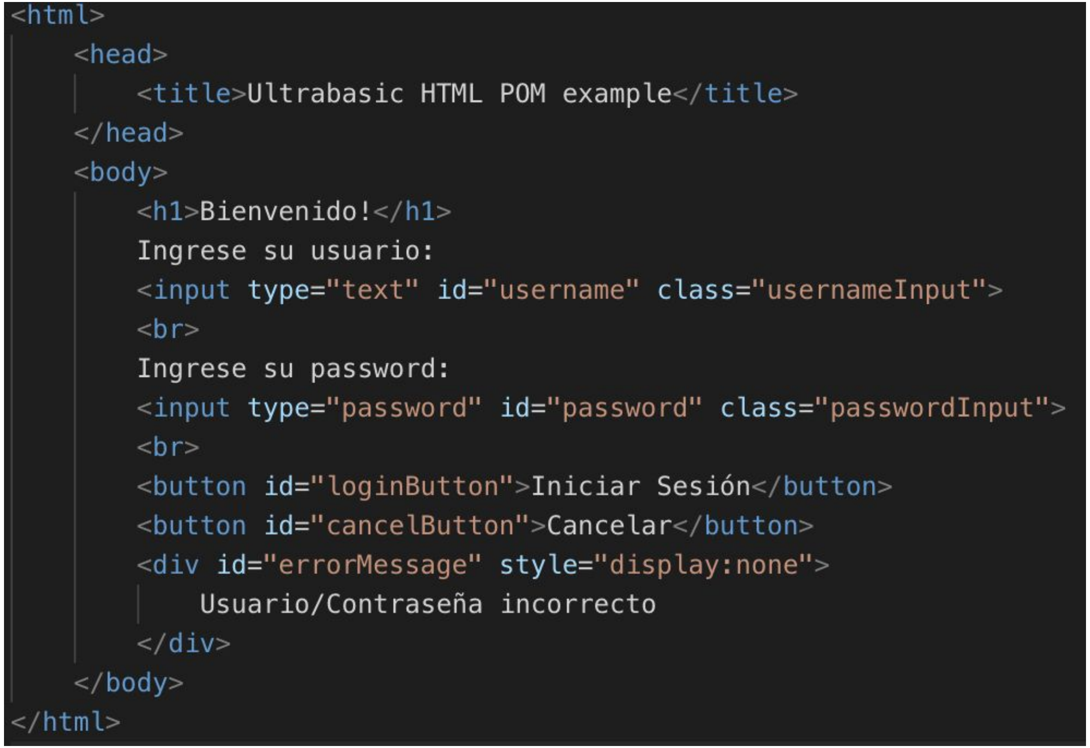
GlobalLogic®

Agenda

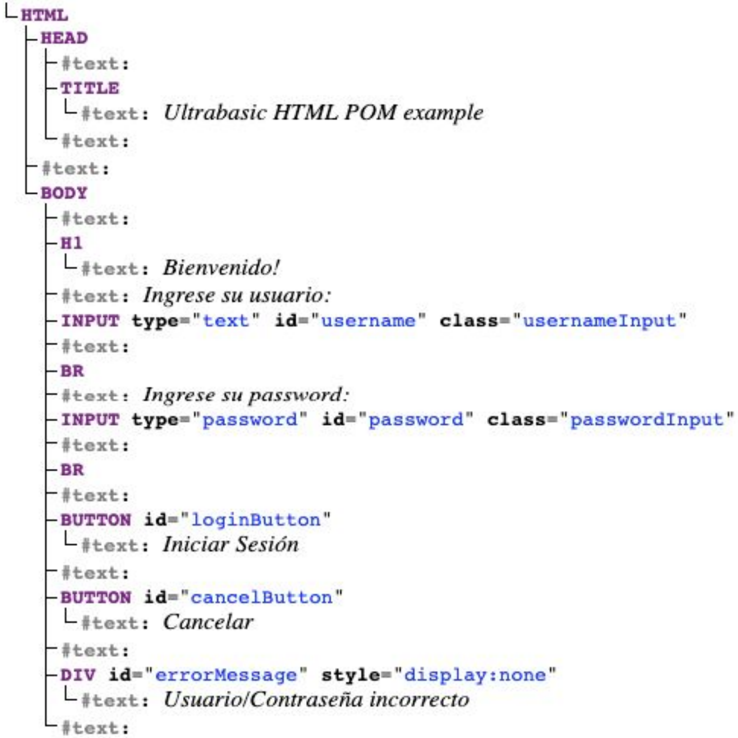
- Conceptos básicos
- Planteo del framework
- BDD y Cucumber
- Arquitectura del framework
- Test Steps

Conceptos básicos

DOM

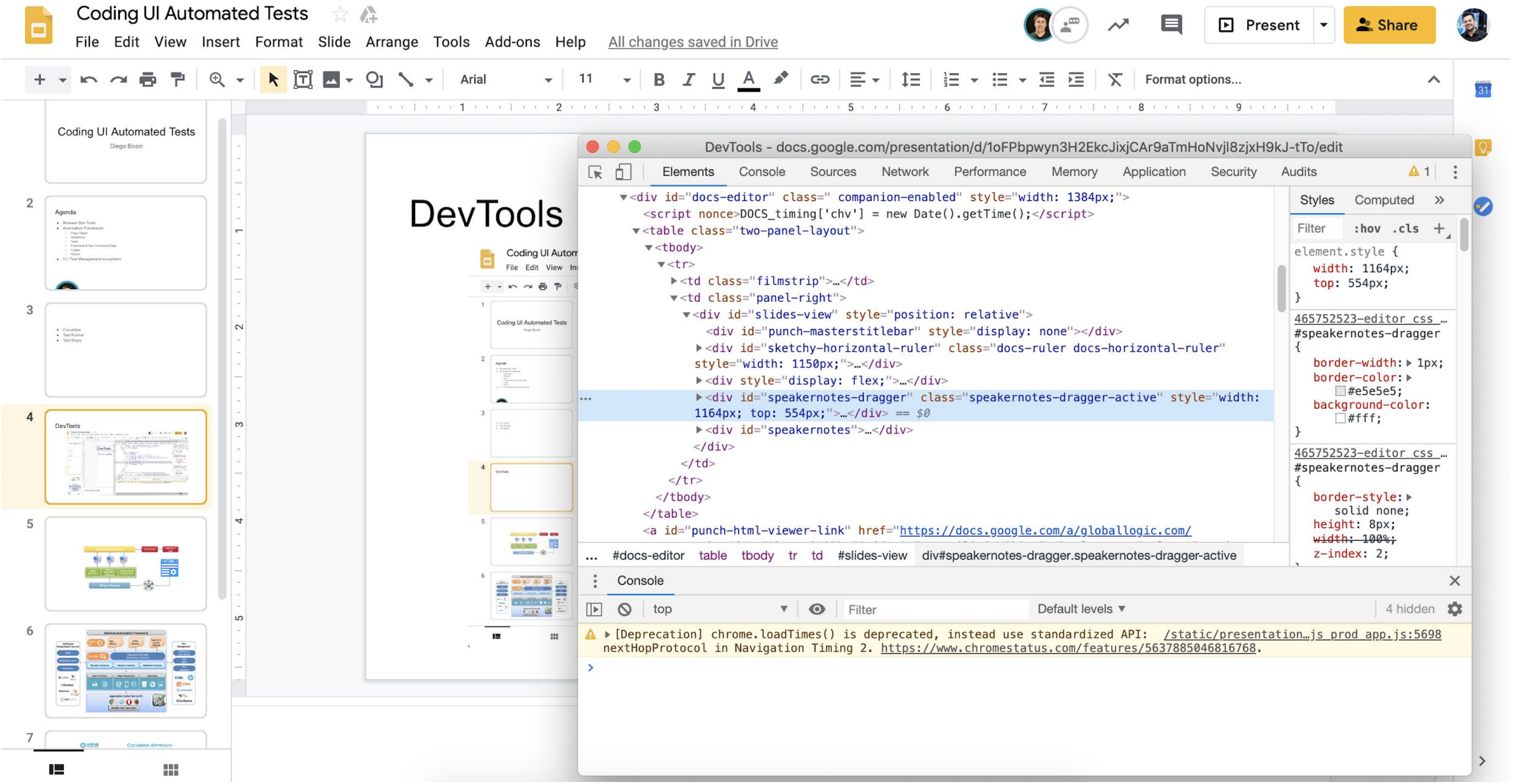


```
<html>
    <head>
        <title>Ultrabasic HTML POM example</title>
    </head>
    <body>
        <h1>Bienvenido!</h1>
        Ingrese su usuario:
        <input type="text" id="username" class="usernameInput">
        <br>
        Ingrese su password:
        <input type="password" id="password" class="passwordInput">
        <br>
        <button id="loginButton">Iniciar Sesión</button>
        <button id="cancelButton">Cancelar</button>
        <div id="errorMessage" style="display:none">
            Usuario/Contraseña incorrecto
        </div>
    </body>
</html>
```



```
└HTML
 ├HEAD
 │├#text:
 │├TITLE
 ││└#text: Ultrabasic HTML POM example
 │└#text:
 ├#text:
 └BODY
  ├#text:
  ├H1
  │└#text: Bienvenido!
  ├#text: Ingrese su usuario:
  ├INPUT type="text" id="username" class="usernameInput"
  ├#text:
  ├BR
  ├#text: Ingrese su password:
  ├INPUT type="password" id="password" class="passwordInput"
  ├#text:
  ├BR
  ├#text:
  ├BUTTON id="loginButton"
  │└#text: Iniciar Sesión
  ├#text:
  ├BUTTON id="cancelButton"
  │└#text: Cancelar
  ├#text:
  ├DIV id="errorMessage" style="display:none"
  │└#text: Usuario/Contraseña incorrecto
  └#text:
```

DevTools



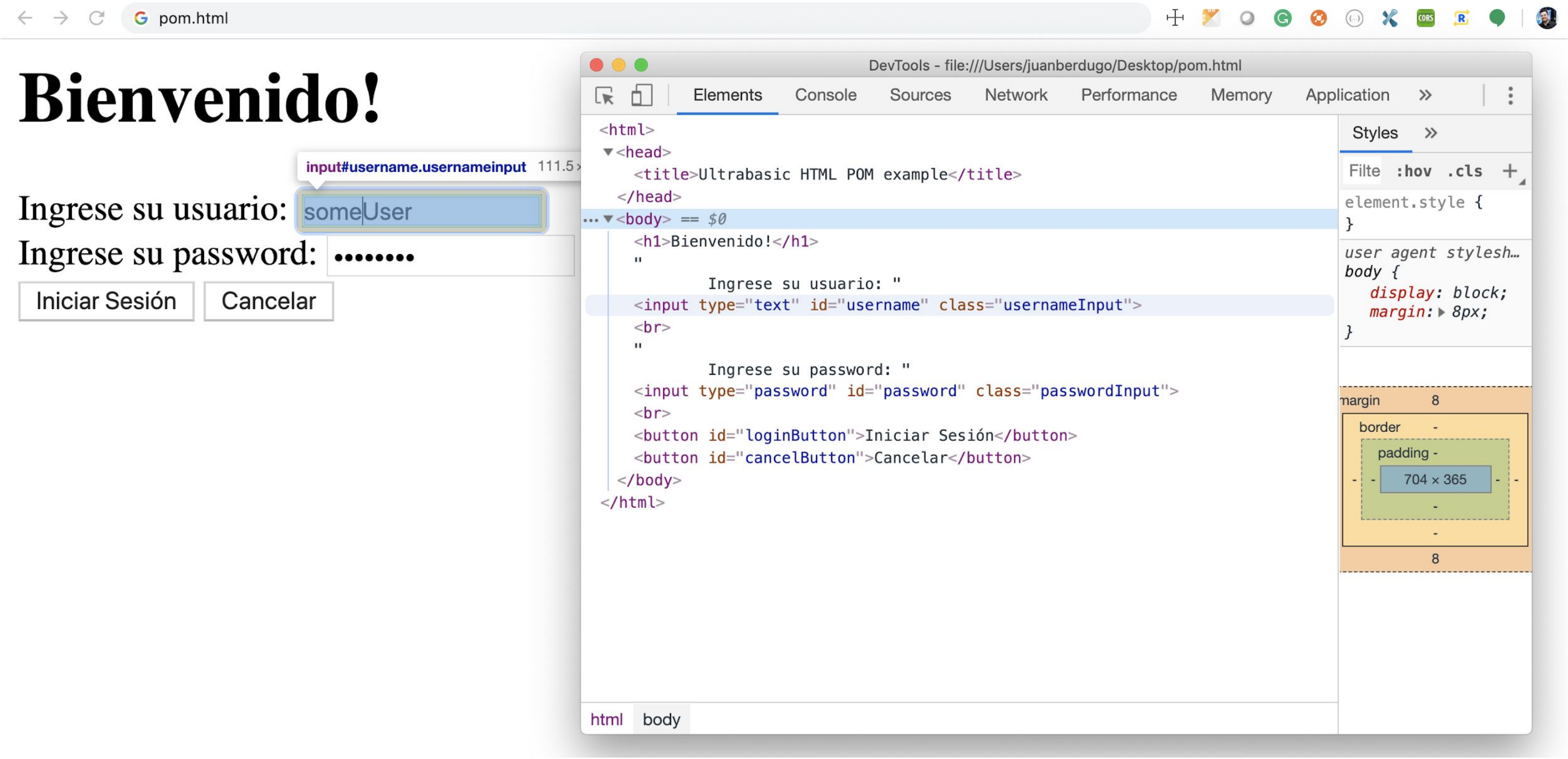
```
▼<div id="docs-editor" class=" companion-enabled" style="width: 1384px;">
   <script nonce>DOCS_timing['chv'] = new Date().getTime();</script>
  ▼<table class="two-panel-layout">
    ▼<tbody>
      ▼<tr>
        ▶<td class="filmstrip">…</td>
        ▼<td class="panel-right">
          ▼<div id="slides-view" style="position: relative">
             <div id="punch-masterstitlebar" style="display: none"></div>
            ▶<div id="sketchy-horizontal-ruler" class="docs-ruler docs-horizontal-ruler"
            style="width: 1150px;">…</div>
            ▶<div style="display: flex;">…</div>
...         ▶<div id="speakernotes-dragger" class="speakernotes-dragger-active" style="width:
            1164px; top: 554px;">…</div> == $0
            ▶<div id="speakernotes">…</div>
            </div>
          </td>
        </tr>
      </tbody>
    </table>
   <a id="punch-html-viewer-link" href="https://docs.google.com/a/globallogic.com/
```

... #docs-editor table tbody tr td #slides-view div#speakernotes-dragger.speakernotes-dragger-active

```
element.style {
   width: 1164px;
   top: 554px;
}
465752523-editor css …
#speakernotes-dragger
{
   border-width: ▶ 1px;
   border-color: ▶
      □#e5e5e5;
   background-color:
      □#fff;
}
465752523-editor css …
#speakernotes-dragger
{
   border-style: ▶
      solid none;
   height: 8px;
   ~~width: 100%;~~
   z-index: 2;
```

```
⚠ ▶[Deprecation] chrome.loadTimes() is deprecated, instead use standardized API:  /static/presentation…js prod app.js:5698
nextHopProtocol in Navigation Timing 2. https://www.chromestatus.com/features/5637885046816768.
>
```

DevTools



pom.html

Bienvenido!

Ingrese su usuario: someUser

Ingrese su password: ••••••••

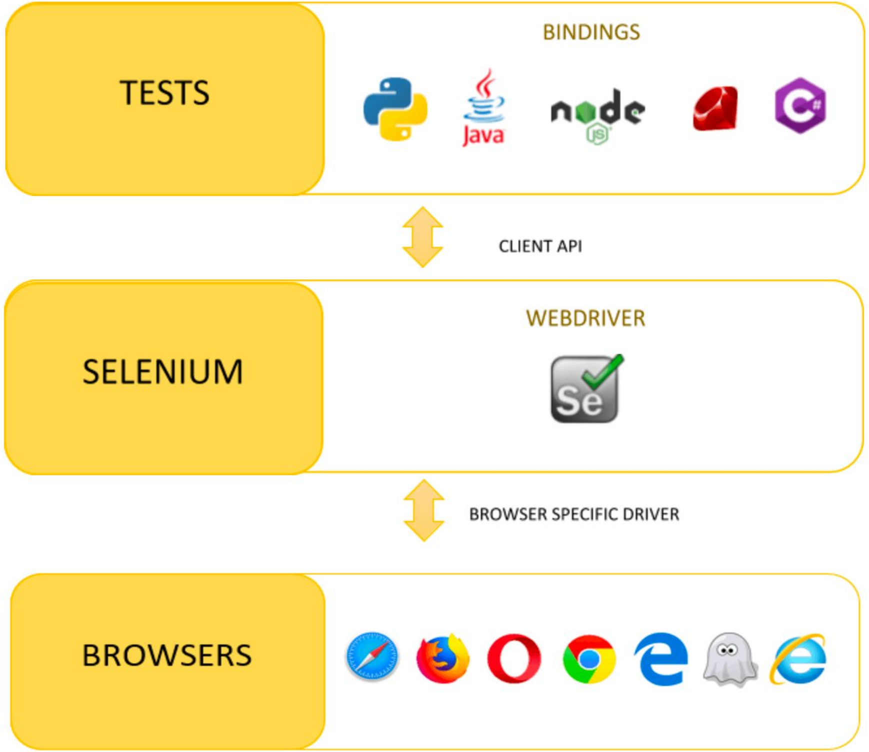
Iniciar Sesión Cancelar

DevTools - file:///Users/juanberdugo/Desktop/pom.html

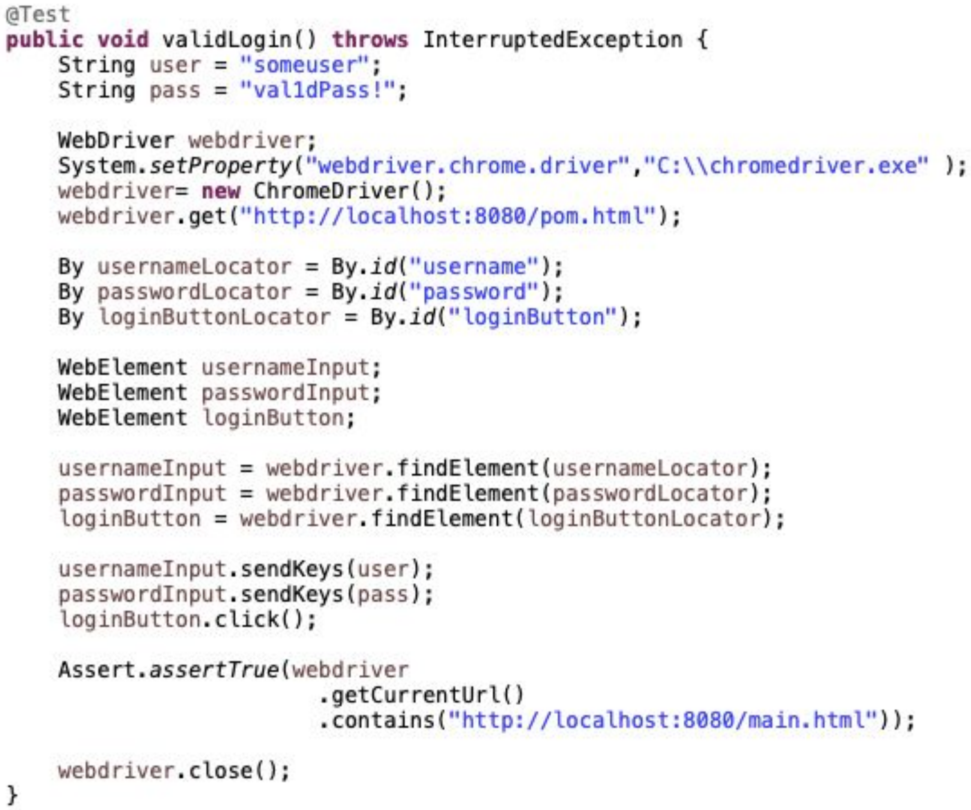
```
<html>
▼<head>
    <title>Ultrabasic HTML POM example</title>
  </head>
…▼<body> == $0
    <h1>Bienvenido!</h1>
    "
          Ingrese su usuario: "
    <input type="text" id="username" class="usernameInput">
    <br>
    "
          Ingrese su password: "
    <input type="password" id="password" class="passwordInput">
    <br>
    <button id="loginButton">Iniciar Sesión</button>
    <button id="cancelButton">Cancelar</button>
  </body>
</html>
```

```
element.style {
}
user agent stylesh…
body {
    display: block;
    margin:▸ 8px;
}
```

WebDriver


TESTS
BINDINGS
Java
node JS
CLIENT API
SELENIUM
WEBDRIVER
Se
BROWSER SPECIFIC DRIVER
BROWSERS

Testing with WebDriver



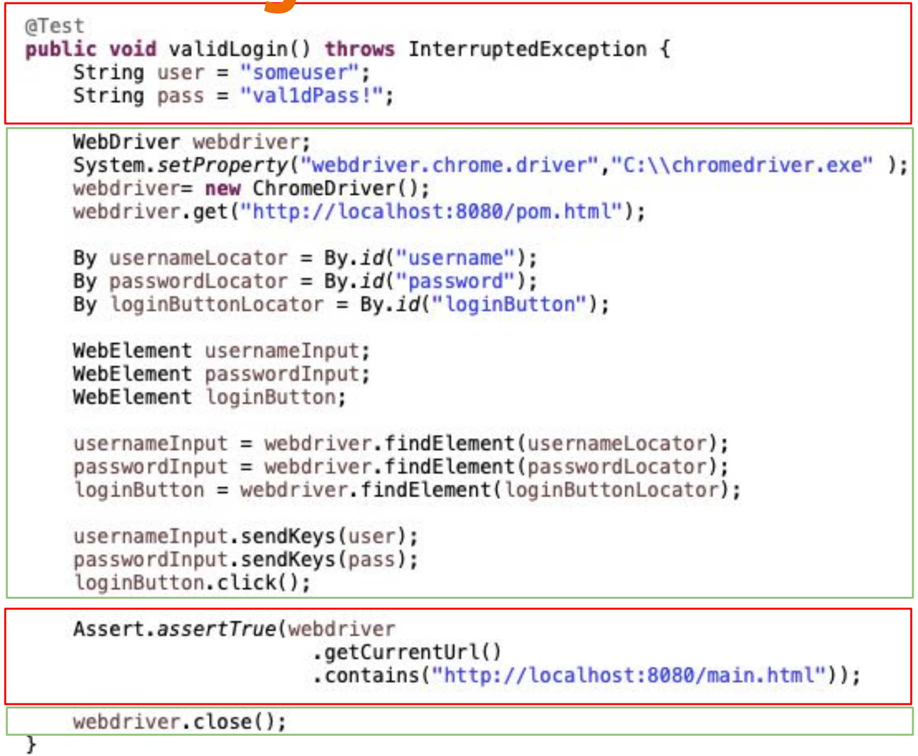
```
@Test
public void validLogin() throws InterruptedException {
    String user = "someuser";
    String pass = "val1dPass!";

    WebDriver webdriver;
    System.setProperty("webdriver.chrome.driver","C:\\chromedriver.exe" );
    webdriver= new ChromeDriver();
    webdriver.get("http://localhost:8080/pom.html");

    By usernameLocator = By.id("username");
    By passwordLocator = By.id("password");
    By loginButtonLocator = By.id("loginButton");

    WebElement usernameInput;
    WebElement passwordInput;
    WebElement loginButton;

    usernameInput = webdriver.findElement(usernameLocator);
    passwordInput = webdriver.findElement(passwordLocator);
    loginButton = webdriver.findElement(loginButtonLocator);

    usernameInput.sendKeys(user);
    passwordInput.sendKeys(pass);
    loginButton.click();

    Assert.assertTrue(webdriver
                        .getCurrentUrl()
                        .contains("http://localhost:8080/main.html"));

    webdriver.close();
}
```

Diseno del framework

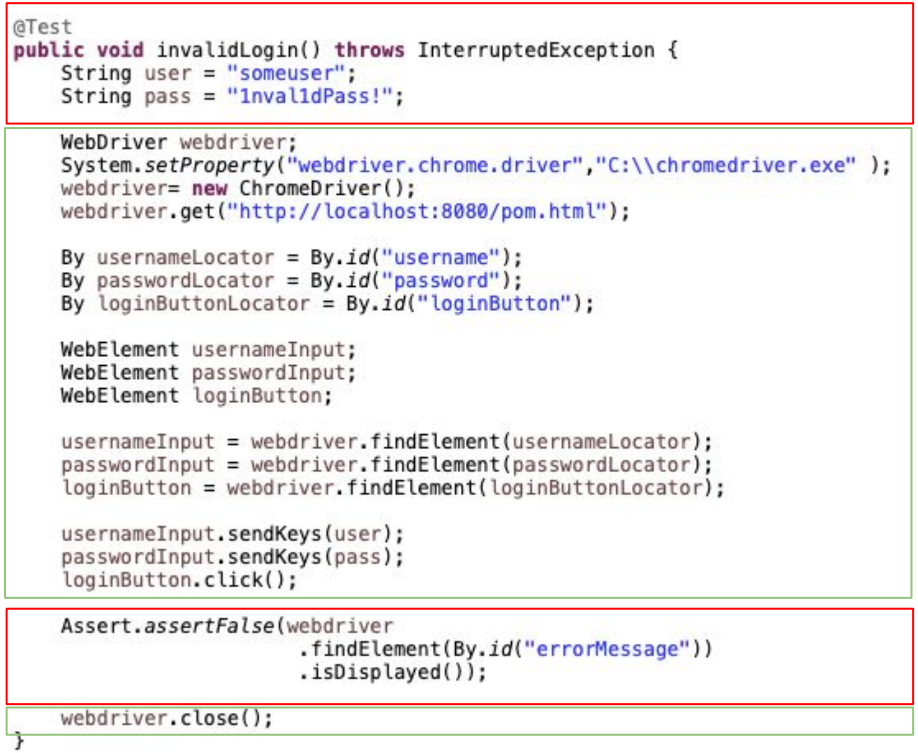
División de responsabilidades

- ¿Qué probar?
  - Tests (scripts)
- ¿Cómo probarlo?
  - Page Objects
  - Webdriver
  - Librerías
- ¿Cómo veo los resultados?
  - Logs
  - Reportes
- ¿Cómo los ejecuto automáticamente?
  - Integración contínua

Testing with WebDriver

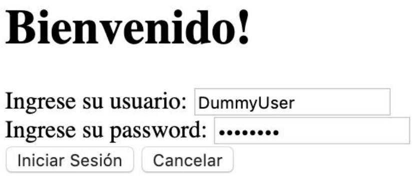


```
@Test
public void validLogin() throws InterruptedException {
    String user = "someuser";
    String pass = "val1dPass!";

    WebDriver webdriver;
    System.setProperty("webdriver.chrome.driver","C:\\chromedriver.exe" );
    webdriver= new ChromeDriver();
    webdriver.get("http://localhost:8080/pom.html");

    By usernameLocator = By.id("username");
    By passwordLocator = By.id("password");
    By loginButtonLocator = By.id("loginButton");

    WebElement usernameInput;
    WebElement passwordInput;
    WebElement loginButton;

    usernameInput = webdriver.findElement(usernameLocator);
    passwordInput = webdriver.findElement(passwordLocator);
    loginButton = webdriver.findElement(loginButtonLocator);

    usernameInput.sendKeys(user);
    passwordInput.sendKeys(pass);
    loginButton.click();

    Assert.assertTrue(webdriver
                        .getCurrentUrl()
                        .contains("http://localhost:8080/main.html"));

    webdriver.close();
}
```



```
@Test
public void invalidLogin() throws InterruptedException {
    String user = "someuser";
    String pass = "1nval1dPass!";

    WebDriver webdriver;
    System.setProperty("webdriver.chrome.driver","C:\\chromedriver.exe" );
    webdriver= new ChromeDriver();
    webdriver.get("http://localhost:8080/pom.html");

    By usernameLocator = By.id("username");
    By passwordLocator = By.id("password");
    By loginButtonLocator = By.id("loginButton");

    WebElement usernameInput;
    WebElement passwordInput;
    WebElement loginButton;

    usernameInput = webdriver.findElement(usernameLocator);
    passwordInput = webdriver.findElement(passwordLocator);
    loginButton = webdriver.findElement(loginButtonLocator);

    usernameInput.sendKeys(user);
    passwordInput.sendKeys(pass);
    loginButton.click();

    Assert.assertFalse(webdriver
                        .findElement(By.id("errorMessage"))
                        .isDisplayed());

    webdriver.close();
}
```

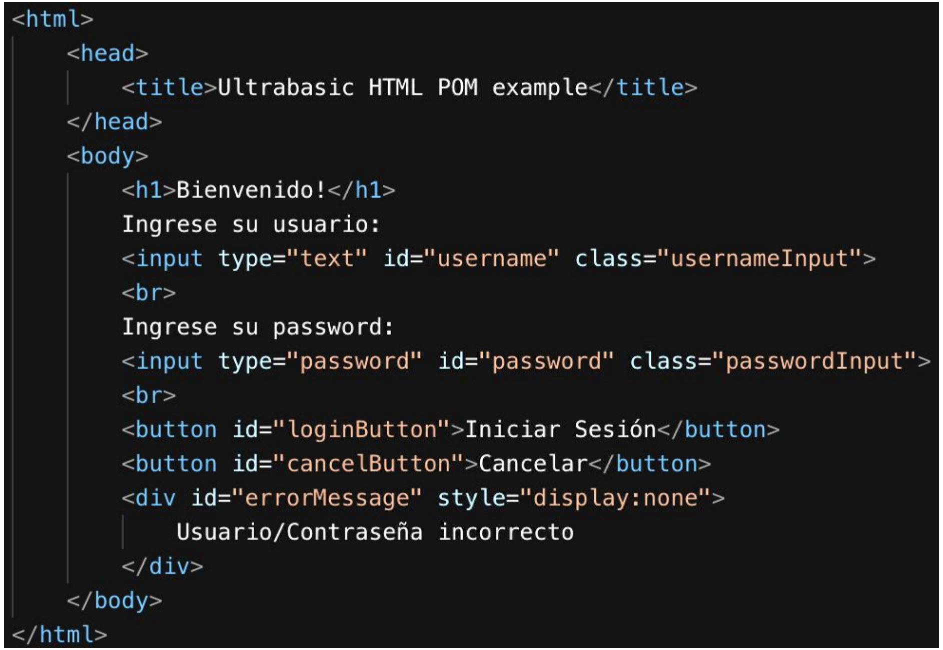
Page Object Model



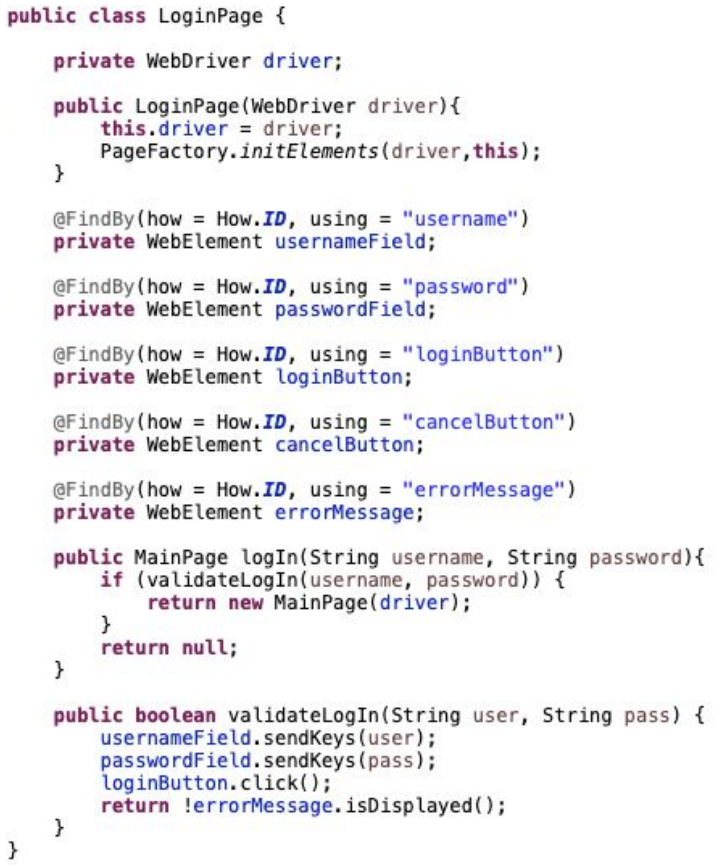
Bienvenido!

Ingrese su usuario: DummyUser

Ingrese su password: ••••••••

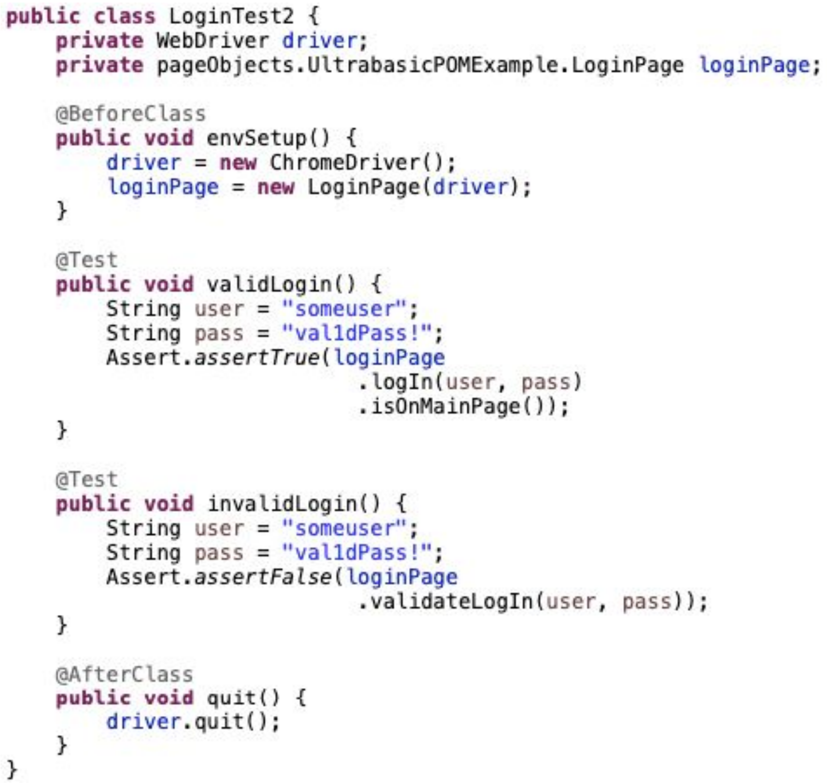


```
<html>
    <head>
        <title>Ultrabasic HTML POM example</title>
    </head>
    <body>
        <h1>Bienvenido!</h1>
        Ingrese su usuario:
        <input type="text" id="username" class="usernameInput">
        <br>
        Ingrese su password:
        <input type="password" id="password" class="passwordInput">
        <br>
        <button id="loginButton">Iniciar Sesión</button>
        <button id="cancelButton">Cancelar</button>
        <div id="errorMessage" style="display:none">
            Usuario/Contraseña incorrecto
        </div>
    </body>
</html>
```

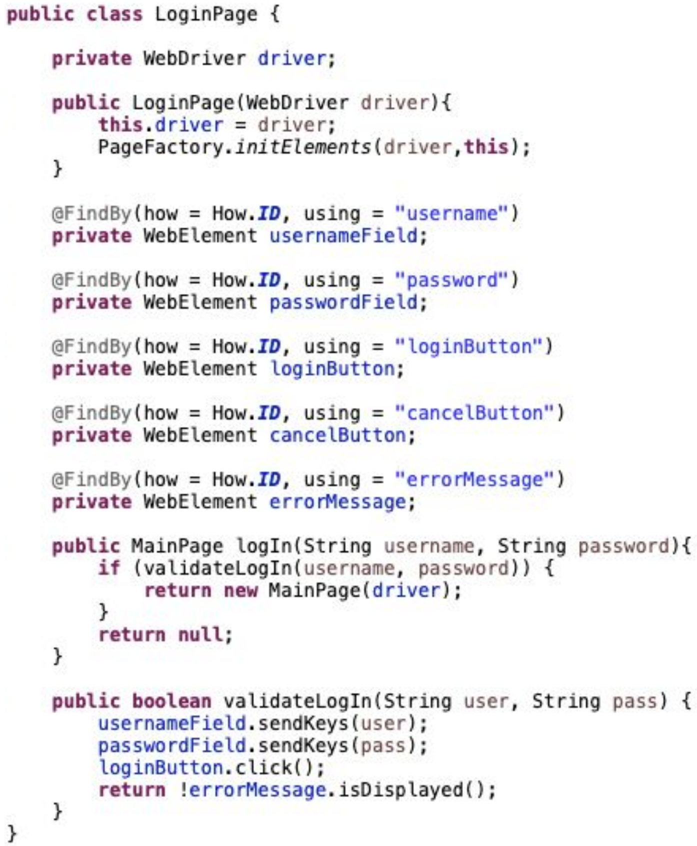


```
public class LoginPage {

    private WebDriver driver;

    public LoginPage(WebDriver driver){
        this.driver = driver;
        PageFactory.initElements(driver,this);
    }

    @FindBy(how = How.ID, using = "username")
    private WebElement usernameField;

    @FindBy(how = How.ID, using = "password")
    private WebElement passwordField;

    @FindBy(how = How.ID, using = "loginButton")
    private WebElement loginButton;

    @FindBy(how = How.ID, using = "cancelButton")
    private WebElement cancelButton;

    @FindBy(how = How.ID, using = "errorMessage")
    private WebElement errorMessage;

    public MainPage logIn(String username, String password){
        if (validateLogIn(username, password)) {
            return new MainPage(driver);
        }
        return null;
    }

    public boolean validateLogIn(String user, String pass) {
        usernameField.sendKeys(user);
        passwordField.sendKeys(pass);
        loginButton.click();
        return !errorMessage.isDisplayed();
    }
}
```

Page Object Model



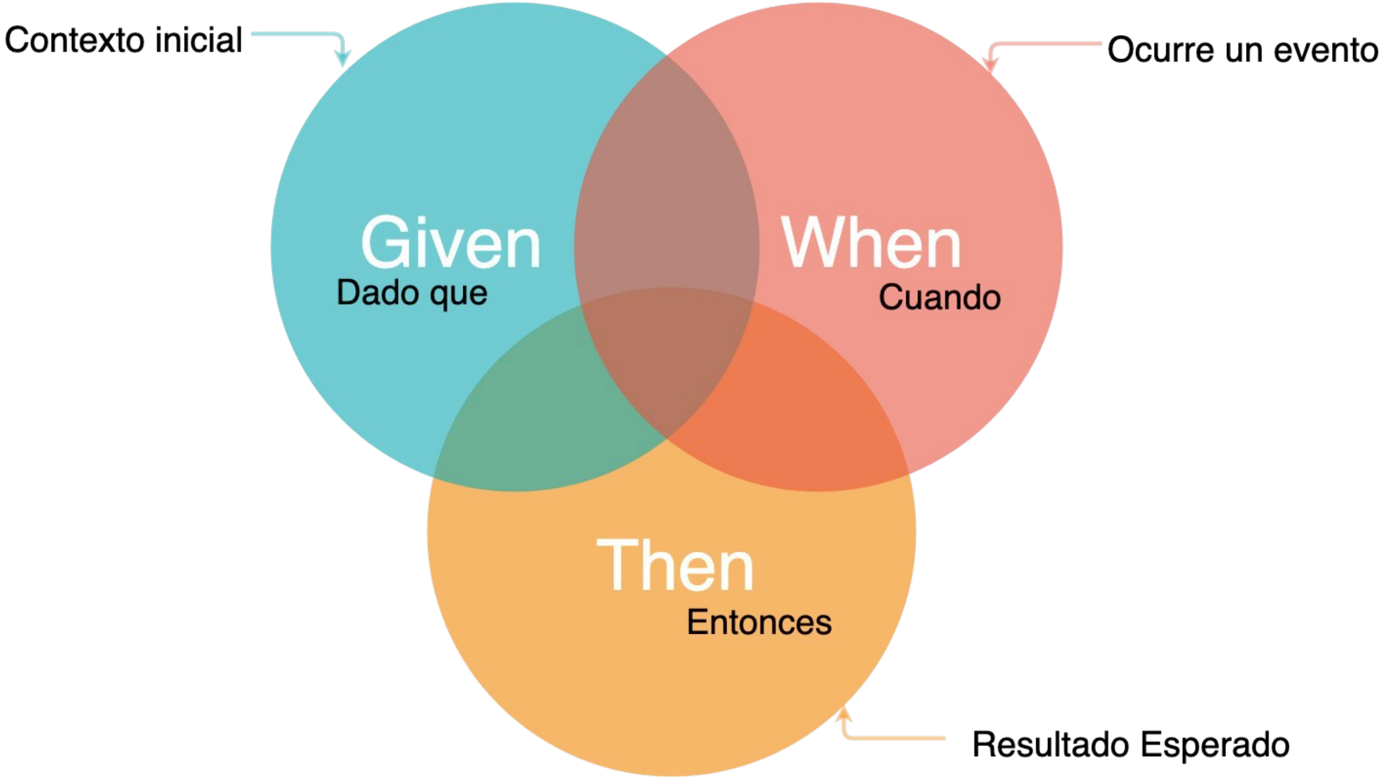
```
public class LoginTest2 {
    private WebDriver driver;
    private pageObjects.UltrabasicPOMExample.LoginPage loginPage;

    @BeforeClass
    public void envSetup() {
        driver = new ChromeDriver();
        loginPage = new LoginPage(driver);
    }

    @Test
    public void validLogin() {
        String user = "someuser";
        String pass = "val1dPass!";
        Assert.assertTrue(loginPage
                        .logIn(user, pass)
                        .isOnMainPage());
    }

    @Test
    public void invalidLogin() {
        String user = "someuser";
        String pass = "val1dPass!";
        Assert.assertFalse(loginPage
                        .validateLogIn(user, pass));
    }

    @AfterClass
    public void quit() {
        driver.quit();
    }
}
```



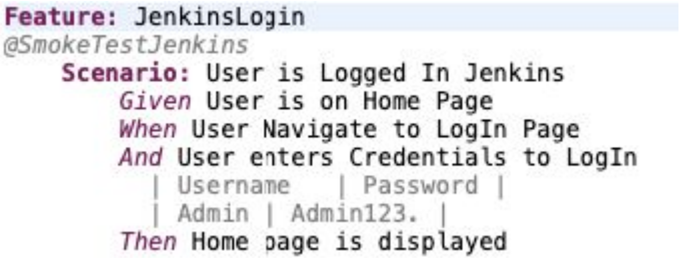
```
public class LoginPage {

    private WebDriver driver;

    public LoginPage(WebDriver driver){
        this.driver = driver;
        PageFactory.initElements(driver,this);
    }

    @FindBy(how = How.ID, using = "username")
    private WebElement usernameField;

    @FindBy(how = How.ID, using = "password")
    private WebElement passwordField;

    @FindBy(how = How.ID, using = "loginButton")
    private WebElement loginButton;

    @FindBy(how = How.ID, using = "cancelButton")
    private WebElement cancelButton;

    @FindBy(how = How.ID, using = "errorMessage")
    private WebElement errorMessage;

    public MainPage logIn(String username, String password){
        if (validateLogIn(username, password)) {
            return new MainPage(driver);
        }
        return null;
    }

    public boolean validateLogIn(String user, String pass) {
        usernameField.sendKeys(user);
        passwordField.sendKeys(pass);
        loginButton.click();
        return !errorMessage.isDisplayed();
    }
}
```

BDD y Cucumber

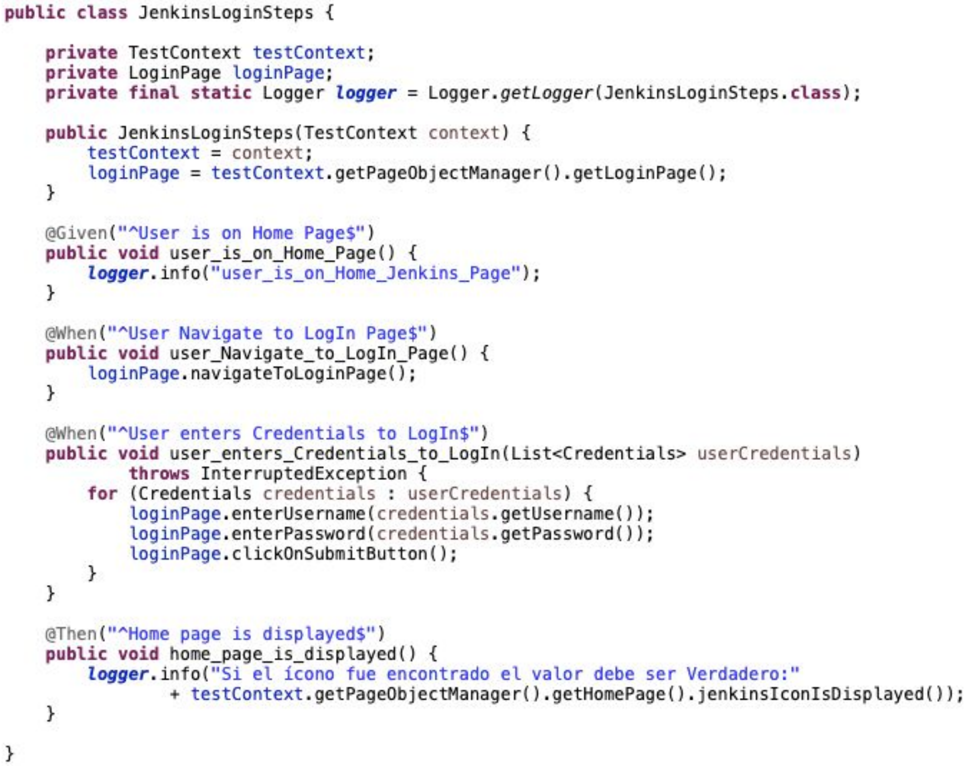
Behaviour Driven Development



Cucumber

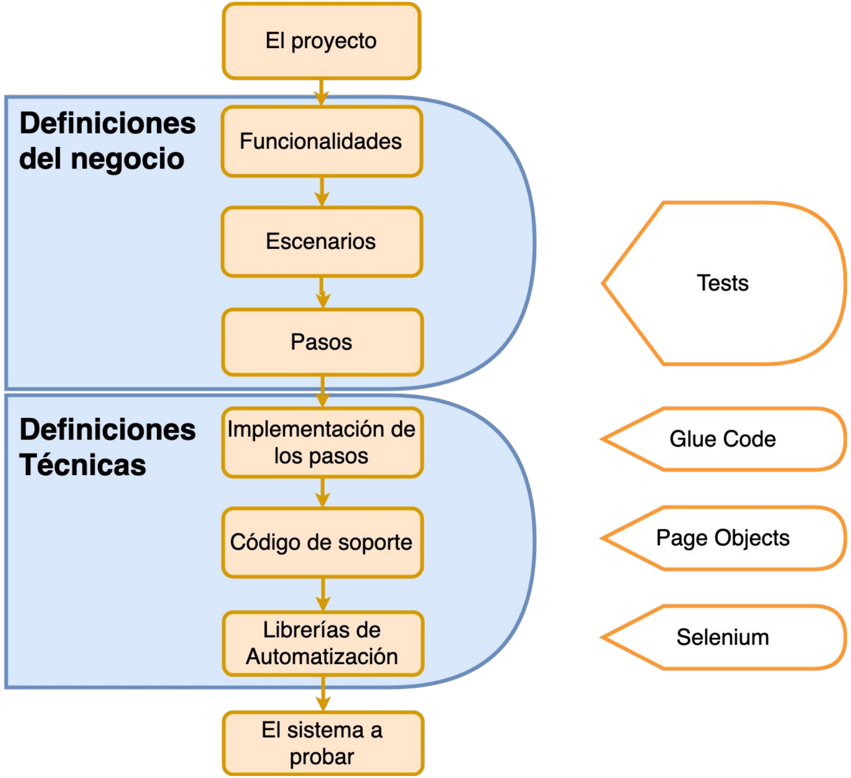


```
Feature: JenkinsLogin
@SmokeTestJenkins
    Scenario: User is Logged In Jenkins
        Given User is on Home Page
        When User Navigate to LogIn Page
        And User enters Credentials to LogIn
          | Username    | Password |
          | Admin | Admin123.  |
        Then Home page is displayed
```



```java
public class JenkinsLoginSteps {

    private TestContext testContext;
    private LoginPage loginPage;
    private final static Logger logger = Logger.getLogger(JenkinsLoginSteps.class);

    public JenkinsLoginSteps(TestContext context) {
        testContext = context;
        loginPage = testContext.getPageObjectManager().getLoginPage();
    }

    @Given("^User is on Home Page$")
    public void user_is_on_Home_Page() {
        logger.info("user_is_on_Home_Jenkins_Page");
    }

    @When("^User Navigate to LogIn Page$")
    public void user_Navigate_to_LogIn_Page() {
        loginPage.navigateToLoginPage();
    }

    @When("^User enters Credentials to LogIn$")
    public void user_enters_Credentials_to_LogIn(List<Credentials> userCredentials)
            throws InterruptedException {
        for (Credentials credentials : userCredentials) {
            loginPage.enterUsername(credentials.getUsername());
            loginPage.enterPassword(credentials.getPassword());
            loginPage.clickOnSubmitButton();
        }
    }

    @Then("^Home page is displayed$")
    public void home_page_is_displayed() {
        logger.info("Si el ícono fue encontrado el valor debe ser Verdadero:"
                + testContext.getPageObjectManager().getHomePage().jenkinsIconIsDisplayed());
    }

}
```

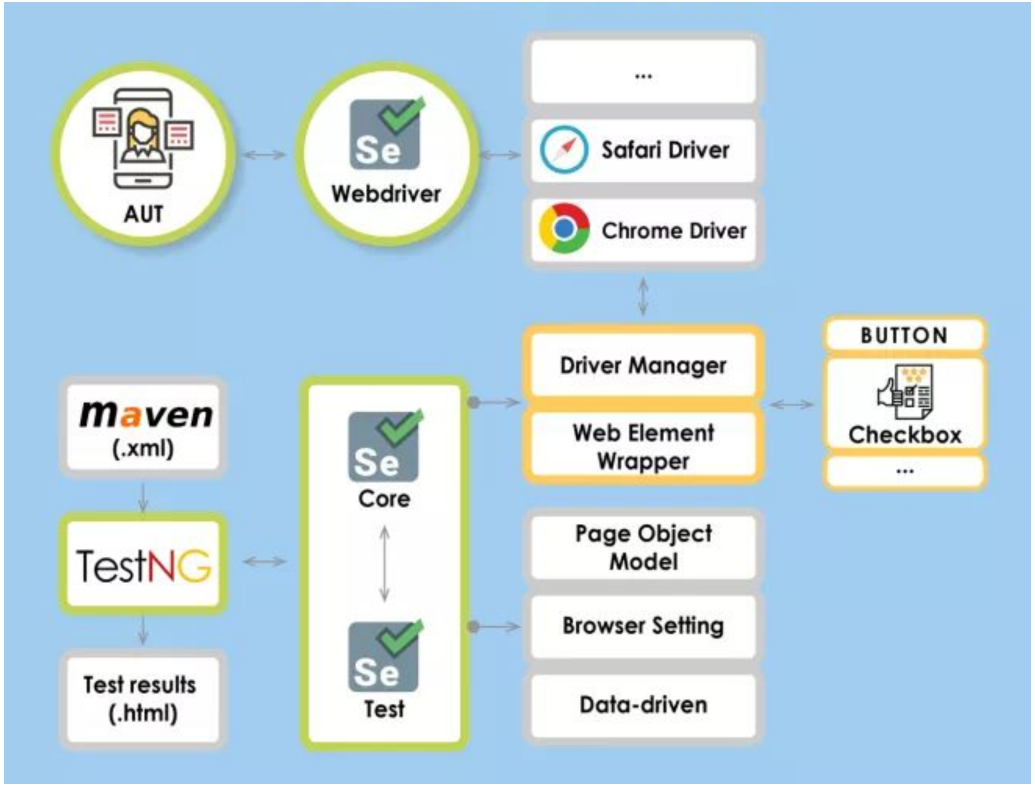
Cucumber



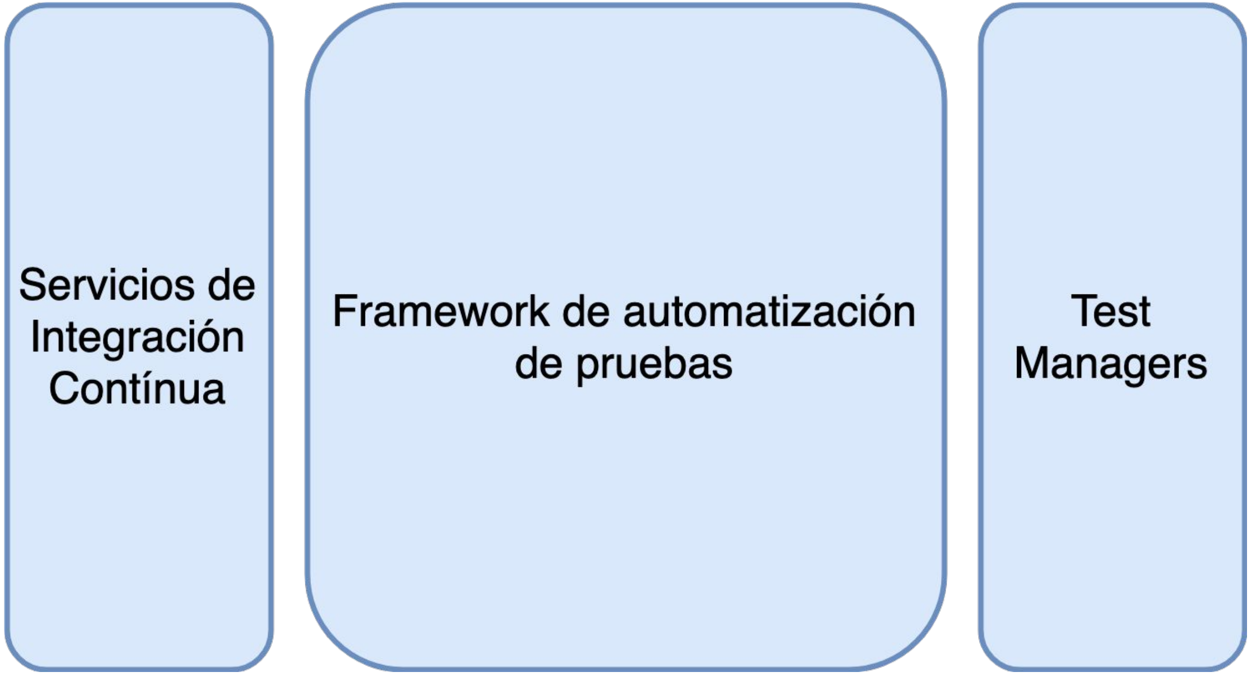
El proyecto
Definiciones del negocio
Funcionalidades
Escenarios
Pasos
Tests
Definiciones Técnicas
Implementación de los pasos
Glue Code
Código de soporte
Page Objects
Librerías de Automatización
Selenium
El sistema a probar

Arquitectura del framework

Framework Architecture



CI / Test Management ecosystem



Servicios de Integración Contínua

Framework de automatización de pruebas

Test Managers

Veamos el codigo...

Dudas?

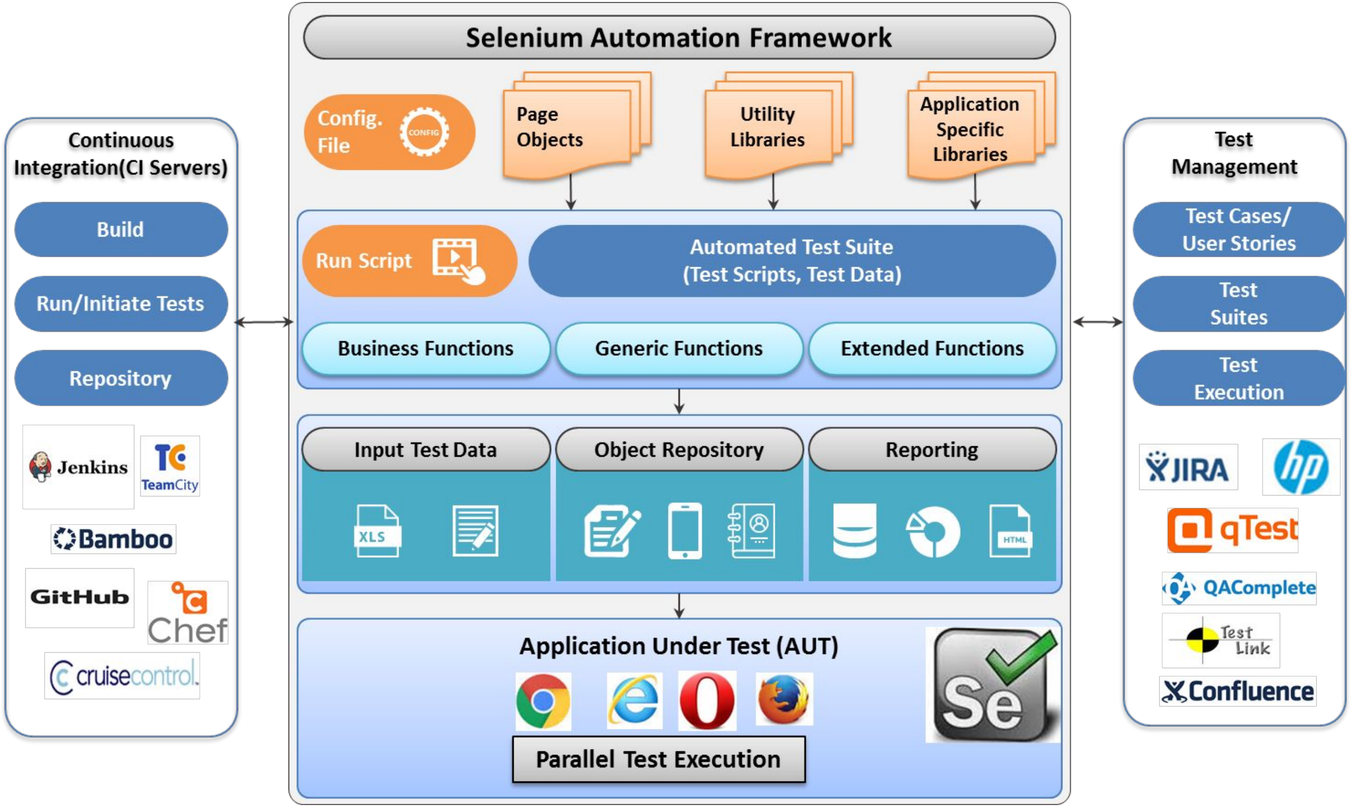
Gracias!

https://github.com/dbosin/MTDAutomationFrameworkTalk

diego.bosin@globallogic.com
in /diegobosin

juan.berdugo@globallogic.com
in /juanberdugo

Appendix - CI / Test Management ecosystem detail


Selenium Automation Framework
Config. File
CONFIG
Page Objects
Utility Libraries
Application Specific Libraries
Run Script
Automated Test Suite (Test Scripts, Test Data)
Business Functions
Generic Functions
Extended Functions
Input Test Data
XLS
Object Repository
Reporting
HTML
Application Under Test (AUT)
Parallel Test Execution
Se
Continuous Integration(CI Servers)
Build
Run/Initiate Tests
Repository
Jenkins
TC TeamCity
Bamboo
GitHub
Chef
cruisecontrol
Test Management
Test Cases/ User Stories
Test Suites
Test Execution
JIRA
hp
qTest
QAComplete
Test Link
Confluence