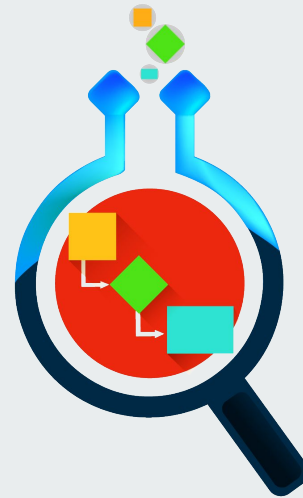


# Coding UI Automated Tests

Diego Bosin & Juan Berdugo



Mendoza  
TESTING  
DAY 2<sup>DA</sup> EDICIÓN

Quiénes somos

GlobalLogic





# Agenda

- Conceptos básicos
- Diseño del framework
- BDD y Cucumber
- Arquitectura del framework

# Conceptos básicos

---

# DOM



## Bienvenido!

Ingrese su usuario:

Ingrese su password:

# DOM

## Bienvenido!

Ingrese su usuario:

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>
```

# DOM

## Bienvenido!

Ingrese su usuario:

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>
```

```
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

Coding UI Automated Tests

File Edit View Insert Format Slide Arrange Tools Add-ons Help All changes saved in Drive

Format options...

DevTools - docs.google.com/presentation/d/1oFPbpw3H2EkCJixjCAR9aTmHoNvj8zjxH9kJ-tTo/edit

Elements Console Sources Network Performance Memory Application Security Audits

```
<div id="docs-editor" class="companion-enabled" style="width: 1384px;">
  <script nonce="DOC5_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="speakersnotes-dragger" class="speakersnotes-dragger-active" style="width:
              1164px; top: 554px;"></div>
            <div id="speakersnotes"></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#speakersnotes-dragger.speakersnotes-dragger-active
```

Styles Computed

Filter :hov .cls +

element.style {

- width: 1164px;
- top: 554px;

465752523-editor\_css\_... #speakersnotes-dragger {

- border-width: 1px;
- border-color: #5e5e5e;
- background-color: #fff;

465752523-editor\_css\_... #speakersnotes-dragger {

- border-style: solid none;
- height: 8px;
- width: 100%;
- z-index: 2;

Console

Filter Default levels 4 hidden

⚠ [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

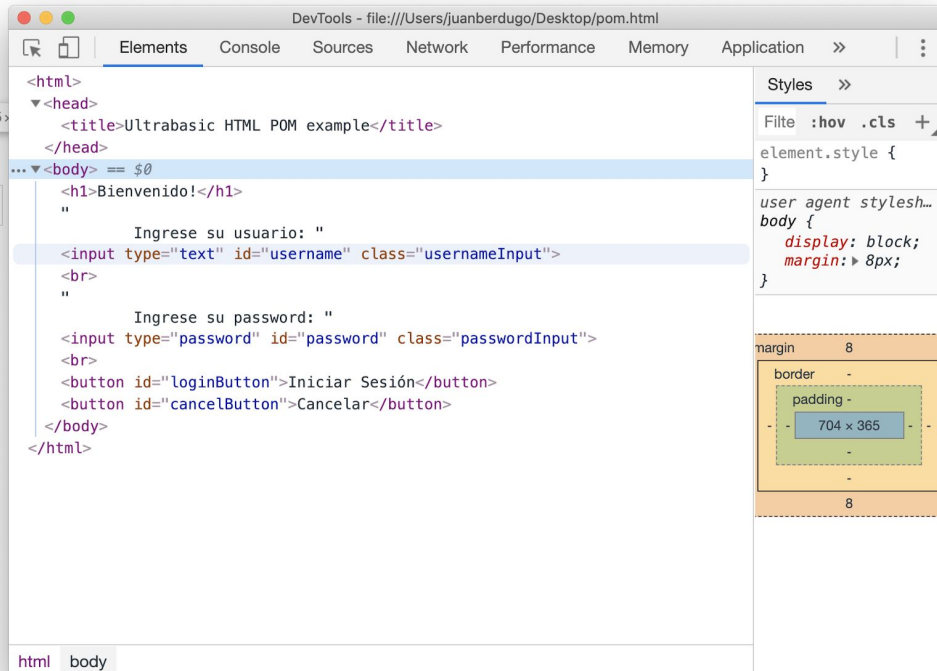
## Bienvenido!

Ingrese su usuario:

Ingrese su password:

Iniciar Sesión

Cancelar



# Diseño del framework

---



## Que es un framework?

Un framework es una abstracción que provee funcionalidades genéricas para un uso determinado,

Provee un entorno universal y reusable que resuelve alguna funcionalidad, y facilita el desarrollo de sistemas y soluciones más grandes.

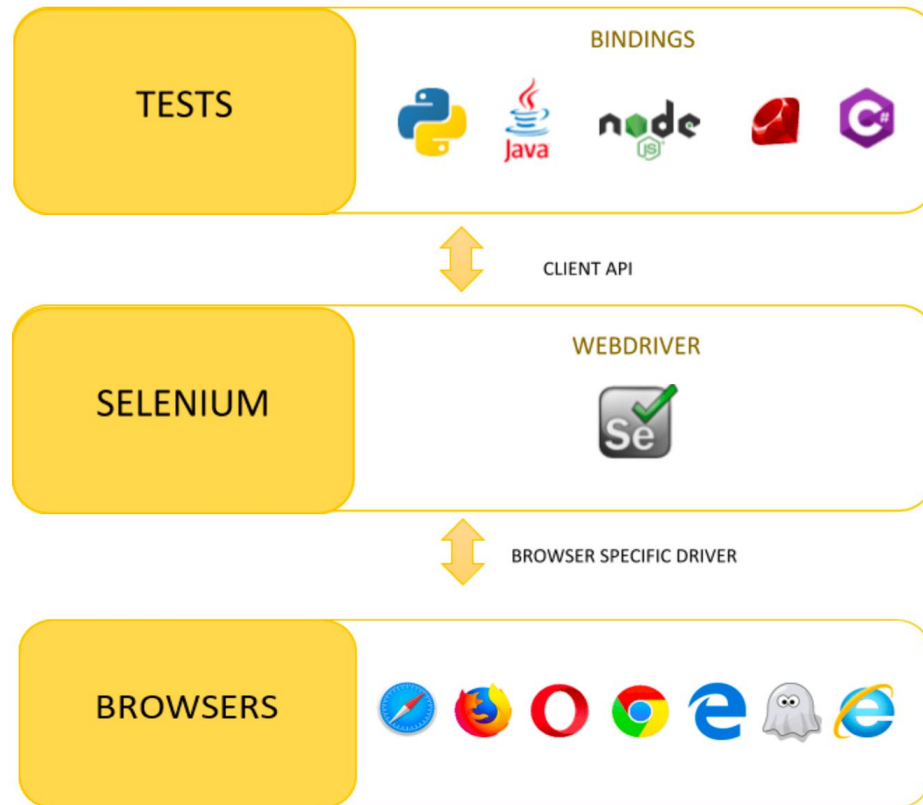
## División de responsabilidades

- ¿Qué probar?
  - Tests (scripts)
- ¿Cómo probarlo?
  - Page Objects
  - Webdriver
  - Librerías
- ¿Cómo ver los resultados?
  - Logs
  - Reportes
- ¿Cómo ejecutar los tests de forma automática?
  - Integración continua





# WebDriver





## Testing with WebDriver

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

    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();
}
```

# Testing with WebDriver

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

    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 = "InvalidPass!";

    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:

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

## Tests

```
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 = "valldPass!";
        Assert.assertTrue(loginPage
            .login(user, pass)
            .isOnMainPage());
    }

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

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

## Page Object

```
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;

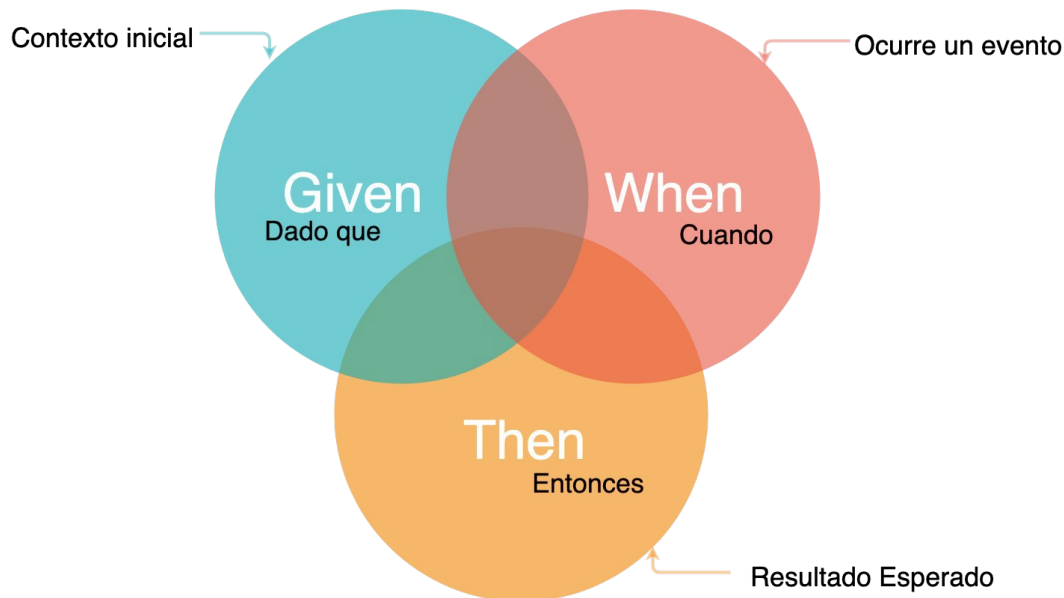
    public MainPage login(String username, String password){
        if (validateLogin(username, password)) {
            return new MainPage(driver);
        }
        return null;
    }

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

# BDD & Cucumber

---

# Behaviour Driven Development



# Cucumber

## Tests

**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

## Steps Definition

```
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();
    }

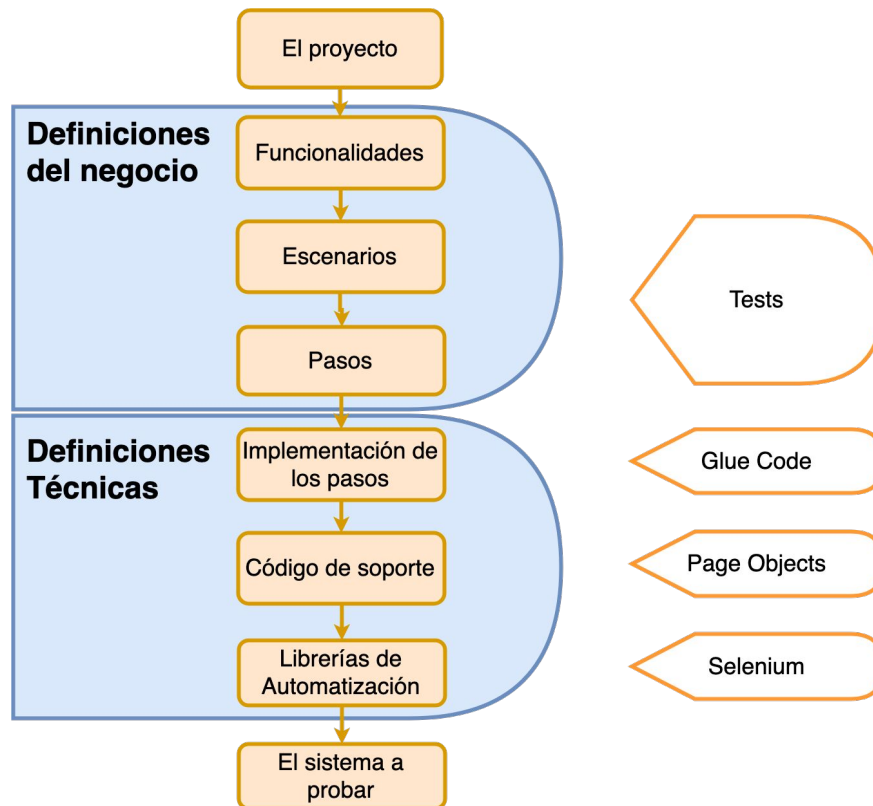
    @Given("^User is on Home Page$")
    public void user_is_on_Home_Page() {
        logger.info("user_is_on_Home_Jenkins_Page");
    }

    @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());
    }
}
```

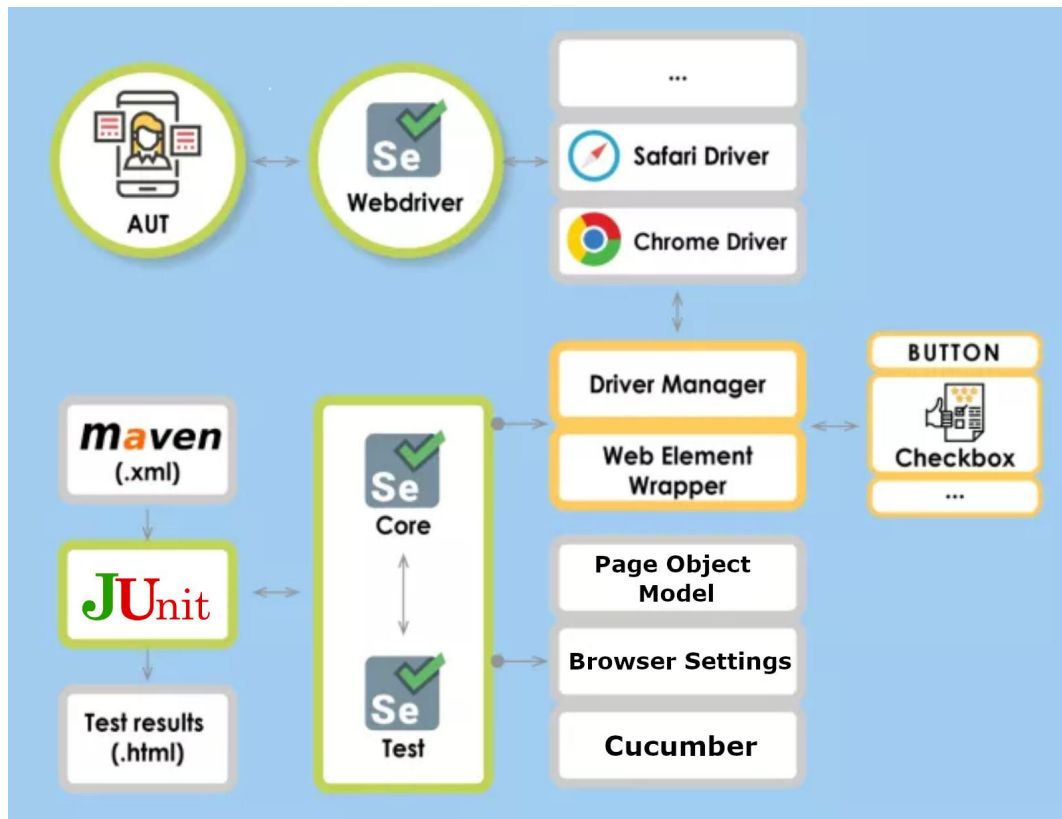
# Cucumber



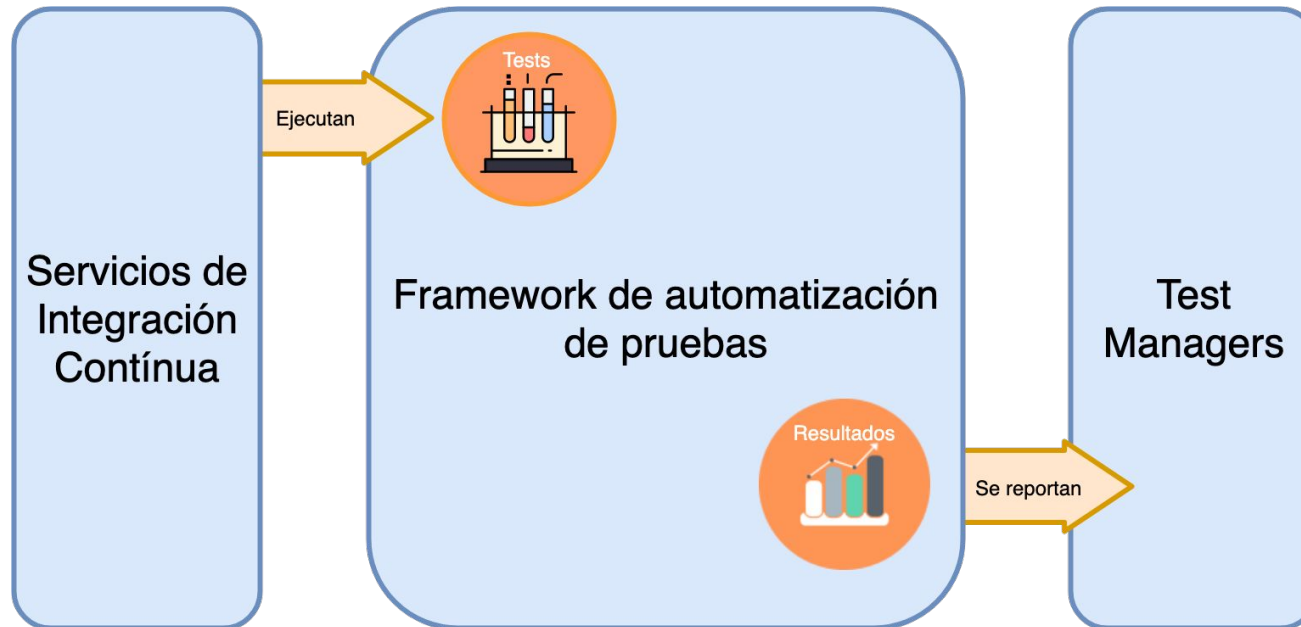
# Arquitectura del framework

---

# Arquitectura del framework



## Ecosistema de CI / Test Management





**Veamos el código...**

---

---

Dudas?





# Gracias!



<http://bit.do/MTDAutomation>

---



[diego.bosin@globallogic.com](mailto:diego.bosin@globallogic.com)



[/diegobosin](#)



[juan.berdugo@globallogic.com](mailto:juan.berdugo@globallogic.com)



[/juanberdugo](#)

# Appendix - CI / Test Management ecosystem detail

