

**CORPORACION UNIVERSITARIA IBEROAMERICANA**

**Actividad 4 - Scripts de testing**

**Heiner Duvan Acosta Urrego**

**Proyecto de software**

**2022**

## 1. ¿Qué es un Script Test?

Un script test es una prueba que se realiza al código con el fin de verificar que esta función del código fuente si está funcionando, estos script están escritos en distintos lenguajes de programación

### 1.1 ¿Qué son pruebas de TDD?

Estas pruebas son pruebas unitarias que se escriben primero que el código, y luego se busca que al escribir pase esta prueba

Características de una buena prueba unitaria

Se ejecutan rápido

Se programan rápido

No depende de elementos externos

No debe depender de otras pruebas unitarias

Ciclo de TDD

Escribir la prueba antes del código

Programar la funcionalidad

Mejorar el código

## 2. TDD Sobre JUNIORAPP

- Comprobar usuario

[Binding]

```
public class LoginSteps
```

```
{
```

```
private LoginDTO loginDTO = new LoginDTO();
```

```
private bool correcto;
```

```
private static CaducaContext caducaContext;
```

```
[Given(@"El usuario administrador tiene contraseña (.*)")]
```

```
public void GivenElUsuarioTieneLaContraseñaNegro
```

```
(string usuario)
```

```
{
```

```
loginDTO.Usuario = usuario;
```

```
}
```

- Comprobar la contraseña

```
[Given(@"Y tiene el password (.*)")]
```

```
public void GivenYTieneElPasswordDtfhkmTRQmNzgRY(string password)
```

```
{
```

```
loginDTO.Password = password;
```

```
}
```

```
[When(@"Yo ingreso al servicio (.*) con estos datos")]
```

```
public async Task WhenIngresoAlServicioUsuariosLoginConEsosDatosAsync(string nombreServicio)
```

```
{
```

```
Servicios.Inicializa();
```

```
caducaContext = Servicios.caducaContext;

correcto = await Servicios.PostAsync
(nombreServicio, loginDTO);
}
```

```
[Then(@"El resultado deberia ser Ok")]

public void ThenElResultadoDeberiaSerOk()

{

Assert.IsTrue(correcto);

}

}
```

Probamos que el programador ya este enrolado con nosotros

```
@Test

public void testThatprogrammerCanBeEnrolled() {

ProgrammerResponse programmerResponse = PorgrammerController

    . enrollProgrammer ([ImmutableProgrammer]Regularprogramer);

assertNotNull(ProgrammerResponse);

asserEquals(expected "ok", ProgrammerResponse.status());
```

Con estas pruebas ya podemos saber si el programador puede ingresar a la red social y obtener el rol seleccionado a la hora de hacer el registro

### **3. Herramientas para Realizar pruebas unitarias en JavaScript.**

- MochaJS
- Jasmine
- AVA
- Jest
- Karma
- Tape
- Cypress.io
- Puppeteer
- ChaiJS
- Qunit
- Sinon

## 4. ¿Qué es BDD?

“BDD refiere a Behavior Driven Development, o sea, desarrollo dirigido por comportamiento. Como bien lo indica su nombre, no se trata de una técnica de testing, sino que es una estrategia de desarrollo (así como TDD, que es test driven development). Lo que plantea es definir un lenguaje común para el negocio y para los técnicos, y utilizar eso como parte inicial del desarrollo y el testing. Por esto es que es importante que vos como tester entiendas bien qué es BDD.”

(Toledo, 2017)

### 3.1 BDD Sobre JuniorApp

**Feature: Búsqueda del proyecto** , Como programador enrolado quiero buscar en la red social mi proyecto asignado.

**Scenario:** Búsqueda del proyecto en Junior app

**Given:** El proyecto debe haber sido asignado con anterioridad.

**When:** Cuando se de click en proyectos.

**Then:** Se mostraran todos los proyectos asignados a este programador según sus conocimientos.

**Conclusiones:**

- Las pruebas TDD están escritas en código mientras las pruebas BDD están en el lenguaje natural.
- Las pruebas TDD hacen del código un código más automatizado y seguro.
- Utilizar estos dos tipos de pruebas realizan un código más estructurado y con menos inconsistencias.
- Las pruebas TDD son implementadas antes de tener el código fuente.