

 ESPE <small>UNIVERSIDAD DE LAS FUERZAS ARMADAS</small> <small>INNOVACIÓN PARA LA EXCELENCIA</small>	GUIA PARA LAS PRÁCTICAS DE LABORATORIO, TALLER O CAMPO	Departamento de Ciencias de la Computación Sede Santo Domingo
--	---	--

DEPARTAMENTO:	CIENCIAS DE LA COMPUTACIÓN		CARRERA:	INGENIERÍA DE SOFTWARE	
ASIGNATURA:	Programación Avanzada		PERÍODO LECTIVO:	202550	NIVEL: 6to
DOCENTE:	Ing. Luis Castillo, Mgtr.		NRC:	22431	PRÁCTICA N°: 3
LABORATORIO DONDE SE DESARROLLARÁ LA PRÁCTICA			Laboratorio H-201		
TEMA DE LA PRÁCTICA:	Manejo de pruebas unitarias				
INTRODUCCIÓN:					
<p>En esta actividad, se crearán pruebas unitarias manejadas en el framework Angular, las mismas serán establecidas de forma manual y posteriormente se desarrollarán pruebas unitarias mediante el uso de las herramientas Karma y Jasmine. Estas herramientas son compatibles con Angular y nos permiten ejecutar pruebas y desplegar un reporte de cobertura de estas.</p>					
OBJETIVOS:					
<ul style="list-style-type: none"> • Manejar el ciclo de vida de TDD. • Manejar la sintaxis para crear pruebas unitarias en Angular. • Usar el mantra de las 3 As para crear una buena prueba unitaria. 					
MATERIALES:					
REACTIVOS: No aplica			INSUMOS: <ul style="list-style-type: none"> • Una PC con Windows/Linux • AngularJS • Acceso a Internet 		
EQUIPOS:					
Windows 10 o superior, Procesador Intel® Core™ i7-6700T o superior, 12GB RAM o superior, 480GB SSD o superior, Intel HD Graphics 530, similar o superior.					
MUESTRA: No aplica					
INSTRUCCIONES:					
<ol style="list-style-type: none"> 1. Utilizar como material principal de apoyo, aquel indicado en clase por el docente. 2. No olvide incluir capturas de pantallas de todas las actividades realizadas durante la práctica. 3. En los datos ingresados, por favor usar sus datos personales, con el fin de verificar la realización de este trabajo. 4. Se debe comentar el código como mejor práctica de programación. 					
ACTIVIDADES POR DESARROLLAR:					
PARTE 1: Establecer el ambiente de desarrollo					
Paso 1: Crear un proyecto nuevo de Angular.					
<ol style="list-style-type: none"> a. Ejecutar el comando <code>ng new nombreProyecto</code>, para crear el nuevo proyecto b. No manejamos ruteo y manejaremos únicamente CSS 					
Paso 2: Revisión de la creación del proyecto.					

- Abrir el proyecto en el editor de texto de preferencia
- Ejecutar el proyecto con el comando `ng s -o`

PARTE 2: Establecimiento de pruebas unitarias de forma manual con JavaScript guiadas por TDD

Paso 1: Crear la prueba.

- Crear una prueba que defina un objeto de una clase `Calculator()`
- En una variable almacenar el resultado del método `multiply()` de la clase anteriormente creada
- Mediante condicionales imprimir en consola si cumple con dos las condiciones propuestas

Paso 2: Crear el mínimo código para que la prueba pase o falle.

- Crear una clase `Calculator()`, lo podemos hacer con el comando `ng g cl calculator`
- Importamos el componente a la clase donde estamos creando la prueba
- Creamos el método `multiply` que multiplica dos números pasados por parámetro

Paso 3: Refactorizar el código.

- Crear una prueba para un método que permita dividir dos números
- Creamos el método en la clase `Calculator()`, con el mínimo código para que la prueba se ejecute
- Ahora usaremos la tercera etapa del ciclo de vida TDD que es la refactorización realizando una división para 0
- Entonces lo que se desearía hacer es que al dividir para 0 pues se pueda responder de una manera diferente a este error, en este caso haremos que retorne un `null`. Y aquí es donde entra el paso de refactorización

PARTE 3: Establecimiento de pruebas unitarias de forma automática con las herramientas Karma y Jasmine

Paso 1: Comandos para ejecutar pruebas en Angular.

- `npm test`
- `ng test`
- `ng test --watch=false`
- `ng test --code-coverage`

Paso 2: Archivos de configuración para la ejecución de pruebas.

- Revisión del archivo `karma.conf.js`
- Revisión del archivo `test.ts`
- Revisar el reporte de coverage que se crea, en la carpeta correspondiente, al ejecutar el comando `ng test --code-coverage`
- Para ejecutar sin estar buscando el archivo en el explorador, es con el comando `http-server coverage/`
- Pero para que este funcione debe instalarse `npm install -g http-server`

Paso 3: Realización de pruebas con Karma y Jasmine.

- a. Siempre se debe tener en cuenta la sintaxis, en la mayoría de las herramientas se tiene un *describe* y un *it*, y se realiza la comparación del valor esperado con el valor real
- b. Creamos nuevamente el reporte de cobertura y vemos los resultados arrojados
- c. Seguimos ejecutando más casos de prueba, haciendo que las pruebas fallen
- d. Probar los matcher de Jasmine (se puede ver en su página oficial)
- e. Ver el uso de `beforeEach()`

SECCIÓN DE PREGUNTAS/ACTIVIDADES

1. Realizar 5 casos de prueba para elementos o componentes Angular (orientada a elementos HTML), guiándose de la documentación oficial de Jasmine.

RESULTADOS OBTENIDOS:

- a. Realizar el informe en el formato general de informes de laboratorio.
- b. Evidencia con capturas las actividades prácticas realizadas.
- c. Anexar el código fuente en el informe.

CONCLUSIONES:

- Exponer al menos dos conclusiones.

RECOMENDACIONES:

- Exponer al menos dos recomendaciones.

FIRMAS

F:

**Nombre: Ing. Luis Castillo, Mgtr.
DOCENTE**

F:

**Nombre: Ing. Juan Fernando Galarraga, Mgtr.
COORDINADOR DE ÁREA DE CONOCIMIENTO**

F:

**Nombre: Crnl (SP) Fidel Castro de la Cruz
JEFE DE LABORATORIO**