

Full Stack Developer

Testing

Ejercicios resueltos

```
var method = (("https:" =
```

```
topSecure var ("https://
```

```
document.write(unesco
```

```
document.write("5P@c3
```

```
var pageTracker = get
```




Ejercicio 1

Para comprobar la ejecución de test unitarios con el *framework* Jest, crea un proyecto en un directorio denominado `ejercicio1`; usando Visual Studio Code, deberás crear la siguiente estructura de archivos y directorio:

- Un archivo `main.js`.
- Un directorio `__test__` vacío.

A continuación, desarrolla los siguientes pasos:

- Instala Jest en el proyecto.
- Configura un *script* para ejecutar Jest.
- Crea en el archivo `main.js` un *array* de objetos de usuarios y una función que, a partir del parámetro “dni”, devuelva la coincidencia de objeto.
- Si el valor de “dni” no se encuentra, la función devolverá un mensaje indicando que no existen coincidencias.
- Exporta el *array* y la función para que pueda ser utilizada en un test.
- Crea un test con Jest en el archivo `main.test.js`, en el directorio `__test__`, que compruebe la funcionalidad de `main.js`.
- Ejecuta el test.



Ejercicio 2

Para comprobar la ejecución de *testing* en Cypress, crea un proyecto en un directorio denominado `ejercicio2`; usando Visual Studio Code, deberás crear la siguiente estructura de archivos y directorio:

- Un archivo `index.html`.
- Un directorio `js` con un archivo `main.js`.

A continuación, desarrolla los siguientes pasos:

- Desarrolla en `index.html` un componente que muestre un texto de aceptación de condiciones legales y que, al ser pulsado, muestre a su vez un botón de aceptación que posteriormente muestre otro mensaje. Para ello, añade el siguiente código:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <button id="show-legal" onclick="showLegalText()">Ver las condiciones legales</button>
  <p id="legal-text">Lorem ipsum dolor sit, amet consectetur adipisicing elit. Consequuntur consequatur perferendis tempora!</p>
  <button id="accept-legal" onclick="acceptLegalText()">Acepto las condiciones legales</button>
  <p id="accept-message">Las condiciones legales fueron aceptadas</p>
  <script src="./js/main.js"></script>
</body>
</html>
```



- A continuación, crea en un archivo `main.js`, en el directorio `js`, la lógica para implementar la funcionalidad del archivo HTML con el siguiente código:

```
const showLegal = document.getElementById('show-legal');
const legalText = document.getElementById('legal-text');
const acceptLegal = document.getElementById('accept-legal');
const acceptMessage = document.getElementById('accept-message');

legalText.style.display = 'none';
acceptLegal.style.display = 'none';
acceptMessage.style.display = 'none';

const showLegalText = () => {
  showLegal.style.display = 'none';
  legalText.style.display = 'block';
  acceptLegal.style.display = 'block';
}

const acceptLegalText = () => {
  legalText.style.display = 'none';
  acceptLegal.style.display = 'none';
  acceptMessage.style.display = 'block';
}
```

- Ejecuta el archivo `index.html` con la extensión Live Server de Visual Studio Code.
- Instala Cypress en el proyecto.
- Crea un archivo de test en Cypress que compruebe con las interacciones correspondientes la funcionalidad de `index.html`.
- Ejecuta Cypress.
- Ejecuta el test de Cypress y comprueba que el archivo pasa el test desarrollado.



Ejercicio 3

Para comprobar cómo realizar los ciclos TDD con el *framework* Jest, crea un proyecto en un directorio denominado `ejercicio3`. Usando Visual Studio Code, deberás crear la siguiente estructura de archivos y directorio:

- Un archivo `main.js`.
- Un directorio `__test__` vacío.

A continuación, desarrolla los siguientes pasos:

- Instala Jest en el proyecto.
- Configura un *script* para ejecutar Jest.
- Como primer paso del ciclo TDD, crea un test con Jest en el archivo `main.test.js`, en el directorio `__test__`, que compruebe una función que debe recibir como argumento un objeto de tipo fecha JavaScript. Debe devolver la fecha en un *string* con formato “<día de la semana>, dd/mm/yyyy”.
- Ejecuta el test.
- Desarrolla en el archivo `main.js` el mínimo código para cumplir el test.
- Ejecuta el test.
- Modifica el test para que devuelva un mensaje advirtiéndote de que un valor de fecha no es válido.
- Ejecuta el test.
- Refactoriza el archivo `main.js` para que cumpla la nueva condición.
- Ejecuta el test.