

QA Engineer - Technical Test

Tarea 1: Pruebas de API

A continuación voy a utilizar JavaScript para obtener un usuario aleatorio de la API de <https://jsonplaceholder.typicode.com/> e imprimir su dirección de correo electrónico:

1. Debo crear el archivo de prueba con la siguiente línea de código

```
Users > andreamorales > JS Tarea1.js > ...
1  fetch('https://jsonplaceholder.typicode.com/users')
2    .then(response => response.json())
3    .then(users => {
4      const user = users[Math.floor(Math.random() * users.length)];
5      console.log(`Correo electrónico del usuario ${user.name}: ${user.email}`);
6    })
7    .catch(error => console.error(error));
8
```

2. Se debe instalar la dependencia node-fetch y ejecutar la prueba

```
MacBook-Pro-de-Andrea:~ andreamorales$ npm install node-fetch
up to date, audited 7 packages in 2s
3 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
```

Resultado:

```
MacBook-Pro-de-Andrea:~ andreamorales$ node Tarea1.js
Correo electrónico del usuario Clementina DuBuque: Rey.Padberg@karina.biz
```

3. Para obtener las publicaciones asociadas al usuario Clementina DuBuque y verificar que contienen ID de publicación válido (un entero entre 1 y 100). Ejecutó la siguiente línea de código:

```
Users > andreamorales > JS Ejercicio2.js > ...
1  fetch('https://jsonplaceholder.typicode.com/users')
2    .then(response => response.json())
3    .then(users => {
4
5      const user = users.find(user => user.name === 'Clementina DuBuque');
6      return fetch(`https://jsonplaceholder.typicode.com/posts?userId=${user.id}`);
7    })
8    .then(response => response.json())
9    .then(posts => {
10
11      const validPosts = posts.filter(post => post.id >= 1 && post.id <= 100);
12      console.log(validPosts);
13    })
14    .catch(error => console.error(error));
15
```

Resultado:

```
MacBook-Pro-de-Andrea:~ andreamorales$ node Ejercicio2.js
[
  {
    userId: 10,
    id: 91,
    title: 'aut amet sed',
    body: 'libero voluptate eveniet aperiam sed\n' +
      'sunt placeat suscipit molestias\n' +
      'similique fugit nam natus\n' +
      'expedita consequatur consequatur dolores quia eos et placeat'
  },
  {
    userId: 10,
    id: 92,
    title: 'ratione ex tenetur perferendis',
    body: 'aut et excepturi dicta laudantium sint rerum nihil\n' +
      'laudantium et at\n' +
      'a neque minima officia et similique libero et\n' +
      'commodi voluptate qui'
  },
  {
    userId: 10,
    id: 93,
    title: 'beatae soluta recusandae',
    body: 'dolorem quibusdam ducimus consequuntur dicta aut quo laboriosam\n' +
      'voluptatem quis enim recusandae ut sed sunt\n' +
      'nostrum est odit totam\n' +
      'sit error sed sunt eveniet provident qui nulla'
  },
]
```

```
{
  userId: 10,
  id: 94,
  title: 'qui qui voluptates illo iste minima',
  body: 'aspernatur expedita soluta quo ab ut similique\n' +
    'expedita dolores amet\n' +
    'sed temporibus distinctio magnam saepe deleniti\n' +
    'omnis facilis nam ipsum natus sint similique omnis'
},
{
  userId: 10,
  id: 95,
  title: 'id minus libero illum nam ad officiis',
  body: 'earum voluptatem facere provident blanditiis velit laboriosam\n' +
    'pariatur accusamus odio saepe\n' +
    'cumque dolor qui a dicta ab doloribus consequatur omnis\n' +
    'corporis cupiditate eaque assumenda ad nesciunt'
},
{
  userId: 10,
  id: 96,
  title: 'quaerat velit veniam amet cupiditate aut numquam ut sequi',
  body: 'in non odio excepturi sint eum\n' +
    'labore voluptates vitae quia qui et\n' +
    'inventore itaque rerum\n' +
    'veniam non exercitationem delectus aut'
},
{
  userId: 10,
  id: 97,
  title: 'quas fugiat ut perspiciatis vero provident',
  body: 'eum non blanditiis soluta porro quibusdam voluptas\n' +
    'vel voluptatem qui placeat dolores qui velit aut\n' +
    'vel inventore aut cumque culpa explicabo aliquid at\n' +
    'perspiciatis est et voluptatem dignissimos dolor itaque sit nam'
},
}
```

```

{
  userId: 10,
  id: 98,
  title: 'laboriosam dolor voluptates',
  body: 'doloremque ex facilis sit sint culpa\n' +
    'soluta assumenda eligendi non ut eius\n' +
    'sequi ducimus vel quasi\n' +
    'veritatis est dolores'
},
{
  userId: 10,
  id: 99,
  title: 'temporibus sit alias delectus eligendi possimus magni',
  body: 'quo deleniti praesentium dicta non quod\n' +
    'aut est molestias\n' +
    'molestias et officia quis nihil\n' +
    'itaque dolorem quia'
},
{
  userId: 10,
  id: 100,
  title: 'at nam consequatur ea labore ea harum',
  body: 'cupiditate quo est a modi nesciunt soluta\n' +
    'ipsa voluptas error itaque dicta in\n' +
    'autem qui minus magnam et distinctio eum\n' +
    'accusamus ratione error aut'
}
}

```

4. Para hacer una publicación usando el usuario Clementina DuBuque con un título y cuerpo no vacíos, se debe verificar que se devuelve la respuesta correcta, para esto debo ejecutar la siguiente línea de código:

```

Users > andreamorales > JS Ejercicio3.js > then() callback
1  fetch('https://jsonplaceholder.typicode.com/users')
2    .then(response => response.json())
3    .then(users => {
4
5      const user = users.find(user => user.name === 'Clementina DuBuque');
6      const newPost = {
7        title: 'Nuevo post',
8        body: 'Cuerpo del nuevo post',
9        userId: user.id,
10     };
11
12     return fetch('https://jsonplaceholder.typicode.com/posts', {
13       method: 'POST',
14       body: JSON.stringify(newPost),
15       headers: {
16         'Content-Type': 'application/json',
17       },
18     });
19   })
20   .then(response => {
21     if (response.ok) {
22       return response.json();
23     } else {
24       throw new Error(`Error al crear la publicación. Código de estado HTTP: ${response.status}`);
25     }
26   })
27   .then(data => console.log(data))
28   .catch(error => console.error(error));
29

```

Resultado:

```
MacBook-Pro-de-Andrea:~ andreamorales$ node Ejercicio3.js
{
  title: 'Nuevo post',
  body: 'Cuerpo del nuevo post',
  userId: 10,
  id: 101
}
```

Tarea 2: Pruebas Automatizadas de UI

1. Se debe instalar la dependencia selenium-webdriver

```
For more details, please visit https://support.apple.com/kb/HT208050.
MacBook-Pro-de-Andrea:Tarea2 andreamorales$ npm install selenium-webdriver

up to date, audited 34 packages in 1s

5 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
MacBook-Pro-de-Andrea:Tarea2 andreamorales$
MacBook-Pro-de-Andrea:Tarea2 andreamorales$
```

2. Importar la librería

```
Users > andreamorales > Downloads > JS Tarea2.js > example
1  const {Builder, By, Key, until} = require('selenium-webdriver');
```

3. Inicializar el controlador de Selenium y abre la página de Google:

```
await driver.get('https://www.google.com');
await driver.findElement(By.name('q')).sendKeys('Aid for Aids', Key.RETURN);
```

4. Luego se ejecutan las validaciones de la prueba

```
await driver.wait(until.elementLocated(By.css('#search')));

let results = await driver.findElements(By.css('.g'));
let count = 0;

for (let result of results) {
  let link = await result.findElement(By.css('a')).getAttribute('href');
  if (link.includes('aidforaids.org')) {
    count++;

    await driver.navigate().to(link);
    let pageTitle = await driver.getTitle();
    console.log(`Página ${count}: Título: ${pageTitle}`);

    let menuItems = await driver.findElements(By.css('.nav-primary li a'));
    if (menuItems.length > 0) {
      let randomIndex = Math.floor(Math.random() * menuItems.length);
      await menuItems[randomIndex].click();
      await driver.wait(until.urlContains(menuItems[randomIndex].getAttribute('href')), 5000);
      console.log(`Página ${count}: Menú: ${await driver.getTitle()}`);
    }
  }
}
```

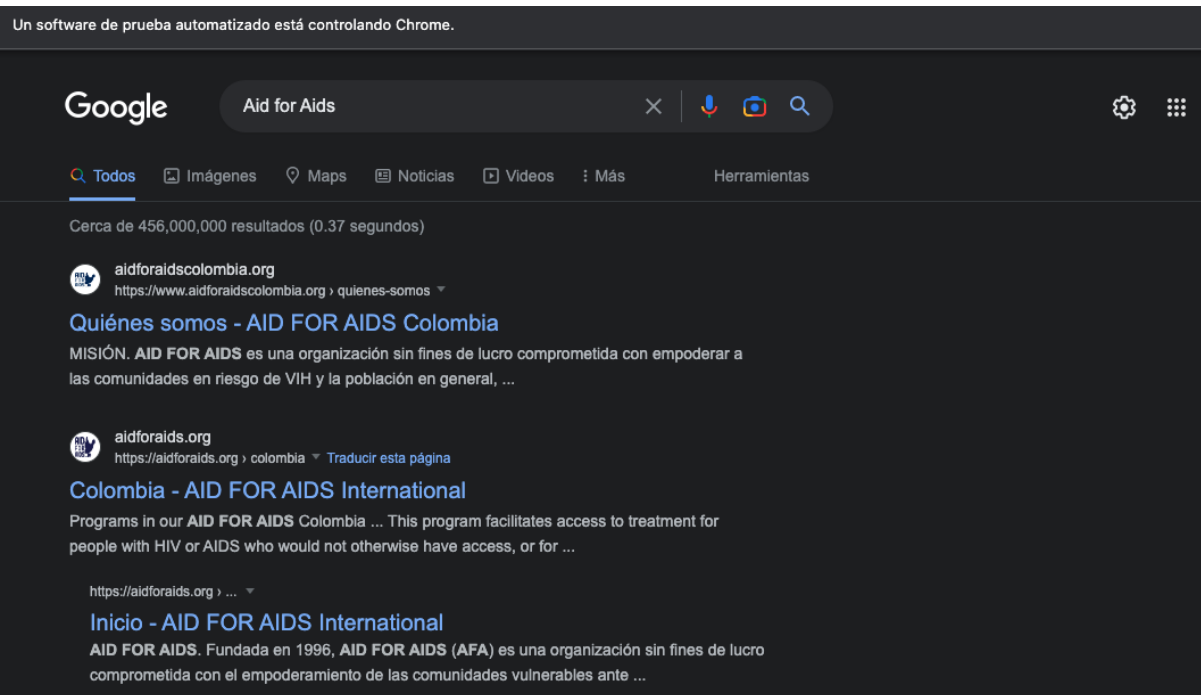
```

}

if (count > 0) {
  console.log(`Se encontraron ${count} resultados que pertenecen al dominio aidforaids.org`);
} else {
  console.log('No se encontraron resultados para el dominio aidforaids.org');
}

} finally {
  await driver.quit();
}

})();
```



Resultado:

```
MacBook-Pro-de-Andrea:Tarea2 andreamorales$ node Ejercicio.js
Página 1: Título: Colombia - AID FOR AIDS International
/Users/andreamorales/node_modules/selenium-webdriver/lib/error.js:524
    let err = new ctor(data.message)
                ^
StaleElementReferenceError: stale element reference: element is not attached to the page document
(Session info: chrome=111.0.5563.110)
    at Object.throwDecodedError (/Users/andreamorales/node_modules/selenium-webdriver/lib/error.js:524:15)
    at parseHttpResponse (/Users/andreamorales/node_modules/selenium-webdriver/lib/http.js:587:13)
    at Executor.execute (/Users/andreamorales/node_modules/selenium-webdriver/lib/http.js:515:28)
    at process.processTicksAndRejections (node:internal/process/task_queues:95:5)
    at async Driver.execute (/Users/andreamorales/node_modules/selenium-webdriver/lib/webdriver.js:745:17)
    at async toWireValue (/Users/andreamorales/node_modules/selenium-webdriver/lib/webdriver.js:145:15)
    at async /Users/andreamorales/node_modules/selenium-webdriver/lib/webdriver.js:195:16
    at async forEachKey (/Users/andreamorales/node_modules/selenium-webdriver/lib/webdriver.js:189:9)
    at async convertKeys (/Users/andreamorales/node_modules/selenium-webdriver/lib/webdriver.js:194:3)
    at async Driver.execute (/Users/andreamorales/node_modules/selenium-webdriver/lib/webdriver.js:743:22) {
  remoteStackTrace: '0   chromedriver                               0x00000000106539428 chromedriver + 4899880\n' +
    '1   chromedriver                               0x000000001064b6a23 chromedriver + 4364835\n' +
    '2   chromedriver                               0x00000000106100bf6 chromedriver + 474102\n' +
    '3   chromedriver                               0x00000000106104c94 chromedriver + 490644\n' +
    '4   chromedriver                               0x00000000106104876 chromedriver + 489590\n' +
    '5   chromedriver                               0x00000000106104d1c chromedriver + 490780\n' +
    '6   chromedriver                               0x0000000010614410c chromedriver + 749836\n' +
    '7   chromedriver                               0x00000000106144751 chromedriver + 751441\n' +
    '8   chromedriver                               0x00000000106136486 chromedriver + 693382\n' +
    '9   chromedriver                               0x0000000010616a58d chromedriver + 906637\n' +
    '10  chromedriver                              0x00000000106136361 chromedriver + 693089\n' +
    '11  chromedriver                              0x0000000010616a71e chromedriver + 907038\n' +
    '12  chromedriver                              0x00000000106185b5b chromedriver + 1018715\n' +
    '13  chromedriver                              0x0000000010616a333 chromedriver + 906035\n' +
    '14  chromedriver                              0x0000000010613455f chromedriver + 685407\n' +
    '15  chromedriver                              0x00000000106135a7e chromedriver + 690814\n' +
    '16  chromedriver                              0x0000000010650679e chromedriver + 4691870\n' +
    '17  chromedriver                              0x0000000010650b961 chromedriver + 4712801\n' +
    '18  chromedriver                              0x000000001065122ff chromedriver + 4739839\n' +
    '19  chromedriver                              0x0000000010650c85a chromedriver + 4716634\n' +
    '20  chromedriver                              0x000000001064defce chromedriver + 4530126\n' +
    '21  chromedriver                              0x0000000010652c5c8 chromedriver + 4847048\n' +
    '22  chromedriver                              0x0000000010652c747 chromedriver + 4847431\n' +
    '23  chromedriver                              0x0000000010654187f chromedriver + 4933759\n' +
    '24  libsystem_pthread.dylib                   0x00007fff6e1cb109 _pthread_start + 148\n' +
    '25  libsystem_pthread.dylib                   0x00007fff6e1c6b8b thread_start + 15\n'
}
```

El resultado de la prueba indica que se ha producido un error de referencia de un elemento caducado, pues se intentó interactuar con una página que ya no está disponible. Además, se muestra una serie de detalles sobre el error, incluyendo la versión de Chrome utilizada en la prueba y una pila de llamadas de funciones que llevaron al error.

Tarea 3: Casos de Prueba

1. Descripción detallada de las características de la aplicación que se probarán.

Objetivo:

- Validar el correcto funcionamiento de la aplicación móvil de restaurante que permite realizar reservas en franjas horarias específicas, calcular el monto de la propina y el total de la factura.

Pruebas de funcionalidad:

- Verificar que la aplicación permite realizar reservas en franjas horarias específicas.
- Verificar que la aplicación permite ingresar el costo de la comida y el porcentaje de propina deseado.
- Verificar que la aplicación calcula correctamente el monto de la propina y el total de la factura.

Pruebas de usabilidad:

- Verificar que la interfaz de usuario es clara y fácil de usar.
- Verificar que la navegación en la aplicación es intuitiva y sin problemas.
- Verificar que los mensajes de error y las alertas son claros y útiles.

Pruebas de rendimiento:

- Verificar que la aplicación carga y responde rápidamente.
- Verificar que la aplicación funciona correctamente en diferentes dispositivos móviles y sistemas operativos.
- Verificar que la aplicación funciona correctamente bajo diferentes condiciones de red, como Wi-Fi

Pruebas de seguridad:

- Verificar que la aplicación no permite el acceso no autorizado a la información del usuario.

- Verificar que la aplicación no permite la manipulación de datos de otras reservas.
- Verificar que la aplicación utiliza un cifrado seguro para proteger la información del usuario.

Pruebas de Integración:

- Verificar que la aplicación se integra correctamente con el sistema de reservas y que las reservas se registran correctamente en la base de datos.
- Verificar que las reservas se pueden hacer correctamente en diferentes franjas horarias.
- Verificar que la aplicación sea capaz de calcular el monto de la propina basado en el porcentaje ingresado por el usuario.
- Verificar que la aplicación realiza correctamente los cálculos y muestra el monto de la propina correctamente.

2. Casos de prueba escritos para cada una de las características que se probarán.

Caso de prueba para reserva exitosa:

- Entrada: el usuario selecciona un día y una franja horaria disponibles para hacer una reserva.
- Resultado esperado: la aplicación confirma la reserva y muestra los detalles de la misma.

Caso de prueba para reserva fallida:

- Entrada: el usuario selecciona una franja horaria que ya está reservada.
- Resultado esperado: la aplicación muestra un mensaje de error indicando que la franja horaria seleccionada ya está ocupada.

Caso de prueba para calcular el costo de la comida y la propina:

- Entrada: el usuario ingresa el costo de su comida y el porcentaje de propina que desea dejar.
- Resultado esperado: la aplicación muestra el monto de la propina y el total de la factura, calculados correctamente.

Caso de prueba para propina máxima:

- Entrada: el usuario ingresa el costo de su comida y selecciona un 100% de propina.

- Resultado esperado: la aplicación muestra el monto de la propina igual al costo de la comida y el total de la factura igual al doble del costo de la comida.

Caso de prueba para campos vacíos:

- Entrada: el usuario no ingresa el costo de la comida o el porcentaje de propina.
- Resultado esperado: la aplicación muestra un mensaje de error indicando que los campos no pueden estar vacíos.

3. Descripción detallada del entorno de prueba que se podría utilizar para realizar las pruebas, incluyendo cualquier herramienta de prueba que se utilizará.

Para el entorno de prueba se debe incluir un dispositivo móvil o un emulador, un sistema operativo compatible, un IDE y las herramientas de Selenium necesarias para realizar las pruebas de la aplicación móvil en el ambiente de pruebas. Además se debe contar con un conjunto de datos para los diferentes usos de aplicación, como ingresar el costo de su comida y el porcentaje de propina que desean entre otros. Por último se debe realizar un informe sobre las pruebas realizadas para registrar los problemas que se presenten en la aplicación

4. Explicación de cómo se registran los resultados de las pruebas y cómo se informará sobre los problemas encontrados.

Para registrar los resultados de las pruebas en Selenium, es necesario:

- Registrar los resultados de las pruebas, obtenidos y los errores encontrados.
- Se deben ejecutar las pruebas en emulador móvil o en un dispositivo real.
- Capturar los resultados si las pruebas han pasado o han fallado
- Informar sobre los problemas encontrados, debe incluir errores en la aplicación, problemas de rendimiento o problemas de compatibilidad con los dispositivos.

Tarea 4: Revisión de UI

Revisar las dos imágenes y realizar reporte de bugs sobre las diferencias encontradas.

A continuación se encontraron diferencias entre el Mockup y la imagen en el entorno de QA, que pueden producir errores en la funcionalidad o la experiencia del usuario final.

- En el menú de la imagen Mockup se encuentran los botones “Inicio”, “Newsletter” y “Catálogo”.
- En el menu de la imagen de Entorno QA se encuentran los botones “Inicio”, “Boletin”, “Catálogo” y “Blogs”.
- El botón “Espacio Asesora” se encuentra en diferentes espacios respecto a cada imagen.
- La imagen Mockup tiene las opciones Mi cuenta y Crear cuenta.
- El header tiene estructura diferente en cada imagen de las opciones Mujeres y Club - Club-BIP
- En la primera sección del body, la imagen de Entorno QA tiene 5 slides de imágenes sobre los servicios que ofrecen mientras la imagen Mockup tiene 3 slides.
- En la primera sección, el cuadro de texto “Tierna y sensual a la vez” tiene una estructura diferente en ambas imágenes.
- En la segunda sección la imagen de Entorno QA tiene el tipo de fuente en color rosa más grande respecto a la imagen Mockup
- En la tercera sección el cuadro "Ofertas" y “Si eres Asesora” tiene una estructura diferente en ambas imágenes.
- En el footer de la imagen de Entorno QA, existe un icono para “Suscribete”
- En el footer de la imagen de Entorno QA debajo de “Suscribete” tiene la leyenda “a nuestro boletín”
- En el footer de la imagen de Mockup debajo de “Suscribete” tiene la leyenda “a nuestro Newsletter”
- En el footer de la imagen de Entorno QA existen los campos “Nombre” y “Correo”
- En el footer de la imagen de Mockup existe el campo que permite ingresar un correo electrónico.

Al reportar estas diferencias, se puede solucionar el problema de diseño o funcionalidad para asegurar que la página cumpla con los requisitos y expectativas iniciales.