

---

13 DE DICIEMBRE DE 2021 - [COM 09]

---

## Testing I

# Examen integrador

Les pedimos que lean atentamente las siguientes consignas y respondan a las preguntas de acuerdo a lo solicitado.

**No se aceptarán links de Drive, solo documentos adjuntos. Caso contrario, el examen no será considerado para su corrección.**

Nota aclaratoria: al enviar el formulario con el adjunto se debe esperar la confirmación del profesor **antes de salir de la sala de Zoom** para garantizar que se recibió correctamente para posterior corrección. Caso contrario, no se recibirá la evaluación y el alumno deberá recuperar esta instancia de evaluación. **Solo se recibirá 1 (un) documento por alumno.**

**Duración:** 1 hora 30 minutos.

**Nombre y Apellido:** ..Laura Ruiz

## Parte teórica

1. Mencionar **2 características** que debe tener **un buen caso de prueba**.  
(Sugerencia: utilizar 2 a 5 renglones/líneas)  
Entres Las condiciones que debe cumplir un caso de prueba, sin falta, se pueden mencionar:

- 1.- Precondiciones.
- 2.- Secuencia de pasos, claros y que permitan reproducir el caso de prueba y obtener los mismo resultados.

2. ¿Qué **ventajas** presentan las **pruebas automatizadas**?  
(Sugerencia: utilizar 2 a 5 renglones/líneas)  
Entres muchas de las ventajas de la automatización, se pueden mencionar las siguientes:

- 1.- Mejorar la eficiencia de la prueba,



- 2.- Reducir el costo total de las pruebas.
- 3.- Aportar una cobertura de funciones mas amplias.
- 4.- Acortar el periodo de ejecución de las pruebas.
- 5.- Realizar pruebas que no se pueden hacer manualmente.

3. Explicar la diferencia entre una **prueba funcional** y una **prueba no funcional**.

(Sugerencia: utilizar 2 a 5 renglones/líneas)

Las pruebas funcionales se orientan hacia cumplir los requerimientos del usuario. Mientras que las pruebas no funcionales se orientan más a nivel código o programación.

4. Explicar la **técnica de prueba de caja negra**.

(Sugerencia: utilizar 2 a 5 renglones/líneas)

Se basan en la información extraída de documentos como requisitos formales, casos de uso, historias de usuarios, entre otros. Tiene como finalidad evaluar el comportamiento del sistema viendo sólo las entradas y salidas del mismo, sin considerar la estructura interna.

5. Explicar el **nivel de prueba de componente**.

(Sugerencia: utilizar 2 a 5 renglones/líneas)

Tiene entre sus objetivos verificar que los comportamientos funcionales y no funcionales de componente sea el esperado. Así como, prevenir la propagación de defectos a niveles de prueba superiores. Utiliza como base de prueba el código y el modelo de datos. Sus objetos de prueba son los componentes, unidades y módulos. códigos o estructura de datos, clases ó módulos de bases de datos.

## Parte práctica

6. ¿Este **test de Postman** es correcto para validar si el contenido devuelto es igual al esperado? Justificar tu respuesta.

```
1  pm.test("Nombre es Victor", function (){  
2      let nombre = pm.response.json().nombre;  
3      pm.response.to.have.status(200);  
4  });  
5
```

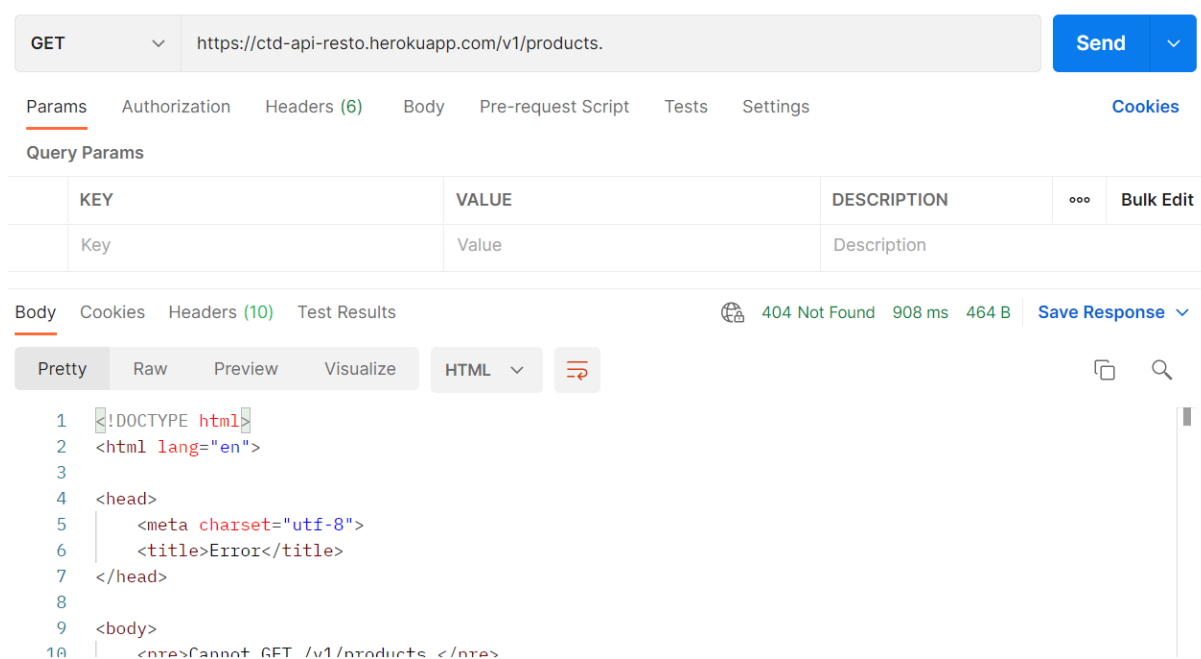


No. Con esta prueba estamos validando que el código de respuesta de la API sea 200. Si esto es correcto el test devolverá PASS: eso significa que el servicio está respondiendo según lo esperado.

Para validar si el contenido devuelto es igual al esperado. Buscamos, en la subsección de fragmentos, haremos clic en "Cuerpo de respuesta: Verificación del valor del JSON". La secuencia de comandos se completará automáticamente. También debemos actualizar el cuerpo de la función reemplazando `jsonData.value` con `jsonData[i].propiedad`; Donde `i` es la posición en la lista del objeto literal JSON que tenemos y `propiedad` es el atributo o propiedad que al cual queremos tener acceso en el JSON. Al cual luego compararemos utilizando `.toEqual()`.

- Continuamos trabajando con nuestra app **Comida Ya!** Esta se conecta con un servicio back-end. Si realizo una petición **GET en Postman** a la siguiente URL: <https://ctd-api-resto.herokuapp.com/v1/products>. ¿Qué resultado arroja y por qué?

Al realizar la solicitud GET a la URL indicada haciendo uso de Postman, se obtiene un error 404. Lo cual significa que el host ha sido capaz de comunicarse con el servidor, pero no existe el recurso que ha sido pedido. Tal como se indica en la siguiente imagen.



- Detallar **1 caso de prueba** que aplicarías en la página de [Comida Ya!](#), solo explicando su descripción (**no utilizar el template**).



Una prueba funcional consistiría en crear un usuario nuevo utilizando email y contraseña válidos. Otra, podría ser crear un producto utilizando una cuenta de administrador (Ambos casos de pruebas positivos).

9. Mencionar **1 defecto** que encuentres en el sitio de [Comida.Ya!](#) (**no utilizar el template**).

Al ingresar en el campo email en el login del usuario, sin cumplir con la estructura de un email (nombre, @, dominio). Se muestra el mensaje "error:usuario no existe". Cuando en realidad, se debe mostrar el mensaje "ingresar email válido", así como exigir y explicar al usuario el error que está cometiendo, el cual consiste en que debe colocar un email válido.

10. Si estoy trabajando con Jest y quiero validar que el resultado devuelto sea **true**. ¿Qué matcher puedo utilizar? Dar un ejemplo de un posible test.

```
test('Resultado True', () => { expect(isTrue).toBeTruthy(); });
```

---