

XX DE DICIEMBRE DE 2021 - [COM XX]

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:Erika Martinez.....

Parte teórica

1. Mencionar **2 características** que debe tener **un buen caso de prueba**.
(Sugerencia: utilizar 2 a 5 renglones/líneas)
 - No asumir funcionalidad o características, verificar siempre la documentación.
 - Debe tener autonomía para que genere siempre los mismos resultados en el caso de prueba.
2. ¿Qué **ventajas** presentan las **pruebas automatizadas**?
(Sugerencia: utilizar 2 a 5 renglones/líneas)
 - Mejora la eficiencia en la prueba aportando una cobertura amplia, reduciendo de esta manera el tiempo y el costo de las pruebas, las pruebas automatizadas son aptas para pruebas complejas y ayudan al tester a encontrar menos errores.



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

(Sugerencia: utilizar 2 a 5 renglones/líneas)

- Las pruebas funcionales son todos los requisitos del sistema desde el punto de vista del usuario y las pruebas no funcionales son las condiciones de funcionamiento del sistema.

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

(Sugerencia: utilizar 2 a 5 renglones/líneas)

- Para la realización de la técnica de prueba de caja negra se dividen los datos en particiones de equivalencia donde cada parte se procesa de la misma manera.

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

(Sugerencia: utilizar 2 a 5 renglones/líneas)

- Su objetivo es buscar defectos en el componente (clases, módulos, código o unidades) y prevenir la propagación de errores a niveles superiores, el desarrollador ejecuta esta prueba.

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
```

- Para validar que el test es correcto es necesario validar el nombre dentro del índice de la lista que debe ser igual al nombre proporcionado, la prueba es exitosa cuando el status 200 indica que la solicitud fue enviada correctamente.



7. 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?
 - El mensaje que arroja es: "Requiere Autorización" en el body y en el status arroja 401 lo que quiere decir que ese endpoint necesita de una autorización para poderlo usar.
8. Detallar **1 caso de prueba** que aplicarías en la página de [Comida Ya!](#), solo explicando su descripción (**no utilizar el template**).
 - ID: DH- 01
 - NOMBRE: Login página web Comida Ya!
 - Descripción: Se ejecuta la prueba para verificar el funcionamiento y el ingreso correcto de usuarios a la página web.
9. Mencionar **1 defecto** que encuentres en el sitio de [Comida Ya!](#) (**no utilizar el template**).
 - En el login de la página Comida Ya! no aparece la opción de olvide contraseña en caso de que se olvide no sé podría restablecer y por lo tanto ingresar el usuario al sistema.
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.
 - Al inicializar mi proyecto con el comando " npm init " se crea mi package. json, se corre el debbuging de nuestro proyecto y es así como se crea el launch.json (entramos a este archivo le modificamos la (url/file) con la ruta de nuestro proyecto, luego se le agrega al final el "///index.html" y en el resto se colocan las barras dobles (\\). Luego en el package.json en la sección que dice scrips se modifica la información que figura como test a "jest" porque así lo requiere el framework y con el comando "npm install -save-dev jest" en la terminal desde la raíz del proyecto se puede visualizar su correcta instalación para luego crear en sus dependencias la carpeta node_modules.