DigitalHouse>

Certified Tech Developer
The Ultimate Degree

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:** Sofía Serrano

## Parte teórica

1. ¿Cuál es la diferencia entre una **suite de humo** y una **suite de regresión**?

La diferencia entre ambas es que la **Suite de regresión** realiza pruebas en funciones secundarias de un programa, mientras que las **suites de humo** realiza pruebas en funciones importantes de un programa y que tienen que estar si o si.

2. Explicar la **técnica de prueba de caja blanca**.

Este tipo de técnica se realiza en la estructura interna del programa, el código hace posible que se pueda hacer en cualquier etapa del desarrollo del software y poder solucionar problemas en una etapa temprana del desarrollo.

3. Mencionar 2 diferencias entre **debugging** y **testing**.

*En el Testing no es necesario tener conocimientos de diseño en el proceso de prueba y en el debugging no se puede realizar sin conocimientos de diseño adecuado.

* El debugging no es una etapa del ciclo de vida del desarrollo de software y ocurre como consecuencia de las pruebas mientras que el testing Sí es una etapa del ciclo de vida del desarrollo de Software (SDLC)

4. ¿En qué ambiente aplicarías **pruebas unitarias**?

La pruebas unitarias las aplicaría en el ambiente DEV (Ambiente de desarrollo) Porque tiene como objetivo encontrar defectos en el componente y verificar que los comportamientos funcionales y no funcionales del componente son los diseñados y especificados.

5. Explicar la etapa de **planificación** dentro del **ciclo de vida de las pruebas de software**.

En esta etapa se definen los objetivos y el enfoque de la prueba dentro de las restricciones impuestas, estas restricciones dependen del contexto en el que están.

En esta etapa podemos definir:

- objetivos
- enfoques
- alcance
- Plan de pruebas
- Coordinar las actividades a realizar

También se evalúan los riesgos que se pueden presentar.

# Parte práctica

6. ¿Qué nos permite validar el siguiente **test de Postman**?

```
pm.test("Test", function (){
    let nombre = pm.response.json().nombre;
    pm.expect(nombre).to.equal("Victor");
});

```

Estamos evaluando si un nombre es igual al de Victor.

7. Continuamos trabajando con nuestra app **Comida Ya!** Esta se conecta con un servicio back-end. Si realizo una petición **POST en Postman** a la

siguiente URL: https://ctd-api-resto.herokuapp.com/v1/users/login con los siguientes datos:

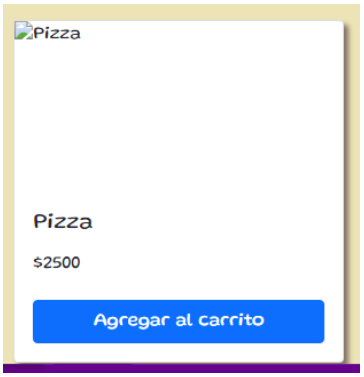
```
{
    "email": "profeDH",
    "password": "123456"
}
```

¿Qué resultado arroja y por qué? ¿Qué significa el código de respuesta devuelto?
Arroja el resultado 404 porque no encuentra al usuario. Este código significa que el host pudo conectarse con el servidor pero no pudo encontrar el recurso que se solicitó.

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

   Validar Registro De Cuenta cliente: Luego de crear una cuenta, Cerramos la cuenta recien creada y apretamos en el texto azul “iniciar sesion” e ingresamos el email “sofiaserrano498@gmail.com” y la contraseña “12345” . El resultado esperado es que nos permite entrar a la página de los productos como cliente .

9. Mencionar **1 defecto** que encuentres en la página de Comida Ya! (**no utilizar el template**).
   La imagen de algunos productos no se pueden visualizar al usuario como es en el caso de la imagen adjunta del producto pizza.




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

Usaría matcher .tobefalsy

```
const miComida={

esManzana:false,

esRojo:true,

};

test(“No es comida”, () =>{

   expect(miComida.esManzana).toBeFalsy();

});
```

---