**EXAMEN INTEGRADOR - TESTING I (CTD) - 05 DE NOVIEMBRE DE 2021 [0921TDTE1M1C1LAED0621FT]**

****

Leer atentamente las siguientes consignas, y responder a las preguntas de acuerdo a lo solicitado.

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

*Nota aclaratoria:* al enviar el adjunto en el [formulario](https://forms.gle/XuJqXFQu7ss9kePv9), 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. **Sólo se recibirá 1 (un) documento por alumno**.

**Duración:** 1 hora 45 minutos.

**Parte teórica**

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

(2 renglones)

las suite de humo se realizan para probar las funciones más importantes de un programa(lo que NO tiene que faltar)mientras que las pruebas de regresion se centran en las funcionalidades secundarias

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

(2 renglones)

Las pruebas de caja blanca se realizan en la estructura interna del programa es decir su código esto le da la ventaja de poder realizarse en cualquier etapa del desarrollo del software

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

(2 renglones)

-El debugging lo realizan los desarrolladores mientras que el testing lo hace el equipo de QA.

-En el debugging se encuentran los errores y se los arregla mientras que en el testing se los encuentra y se los reporta para que lo solucionen en el equipo de desarrollo.

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

(2 renglones)

Las pruebas unitarias se aplican en el ambiente de desarrollo (DEV)

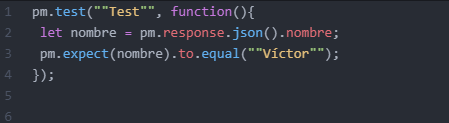
1. Explicar la etapa de **planificación** dentro del **ciclo de vida de las pruebas de software**.

(2 renglones)

En la planificacion definimos los objetivos, los enfoques, el alcance,los calendarios de pruebas ,las tecnicas necesarias es decir generamos un plan de pruebas .También se evalúan los riesgos que se puedan presentar

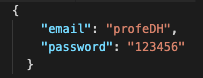
**Parte práctica**

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



en este caso estamos evaluando Victor

1. 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:



¿Qué resultado arroja y por qué? ¿Qué significa el código de respuesta devuelto?

Arroja “el usuario no existe” con el status 404 not found(el servidor web no encuentra el recurso solicitado )

1. Detallar **1 caso de prueba** que aplicarías en la página de [Comida Ya!](https://ctd-app-resto.herokuapp.com/admin/dashboard/), solo explicando su descripción (**no utilizar el template**).

Agregar al carrito:después de crear nuestra cuenta nos dirigimos al botón de productos y seleccionamos los productos que deseamos apretando el boton agregar al carrito, estos se agregan al carrito sumando su importe

1. Mencionar **1 defecto** que encuentres en la página de [Comida Ya!](https://ctd-app-resto.herokuapp.com/admin/dashboard/) (**no utilizar el template**).

A la hora de crear un usuario en la parte del formulario que nos pide nuestro E-mail no es necesario colocar un E-mail con un formato adecuado

1. 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.

Se podría usar el matcher .tobefalsy

const miVehiculo={

esVerde:true,

esmoto:false,

};

test(“no es moto”, () =>{

expect(miVehiculo.esmoto).toBeFalsy();

});