21 DE ABRIL DE 2022 [0222TDTE1M1C1LAED1021FT]

# 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**.

Recordá que cada ejercicio vale 1 punto.

**Duración**: 1 hora 30 minutos.

**Nombre y Apellido:** Dayana Otagrí

Parte teórica

1. ¿Cuál es la diferencia entre las pruebas de **humo** y las pruebas **regresión**?

En la pruebas de HUMO se prueban las funciones que tienen mayor relevancia en el programa, además son aplicadas antes de haberlo lanzado, mientras que, en las pruebas de regresión se aplican funcionalidades primarias y secundarias del programa incluso si este ya ha sido lanzado.

1. Explique con un ejemplo un tipo de **técnica de prueba de caja blanca**.  
     
   Esta técnica se basa en el código, la arquitectura, el diseño detallado y estructura.   
   Por ejemplo cuando se tiene el código en funcionamiento y vamos a verificar que el código sea correcto respecto a lo que se espera que haga.
2. Mencionar las principales diferencias entre **debugging** y **testing**.

El tester registra los defectos y fallos del sistema   
El Debugging corrige los casos de prueba reportados por testing

El testing no necesita ver el código del programa, mientras que para el debugging es lo que más necesita para poder corregir.

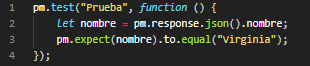
1. ¿En qué ambiente aplicarías **pruebas unitarias**? Justifique su respuesta.

Las pruebas unitarias aíslan una sección de código y prueban que su comportamiento sea el esperado y funcione a la perfección, normalmente es llevado a cabo por los desarrolladores

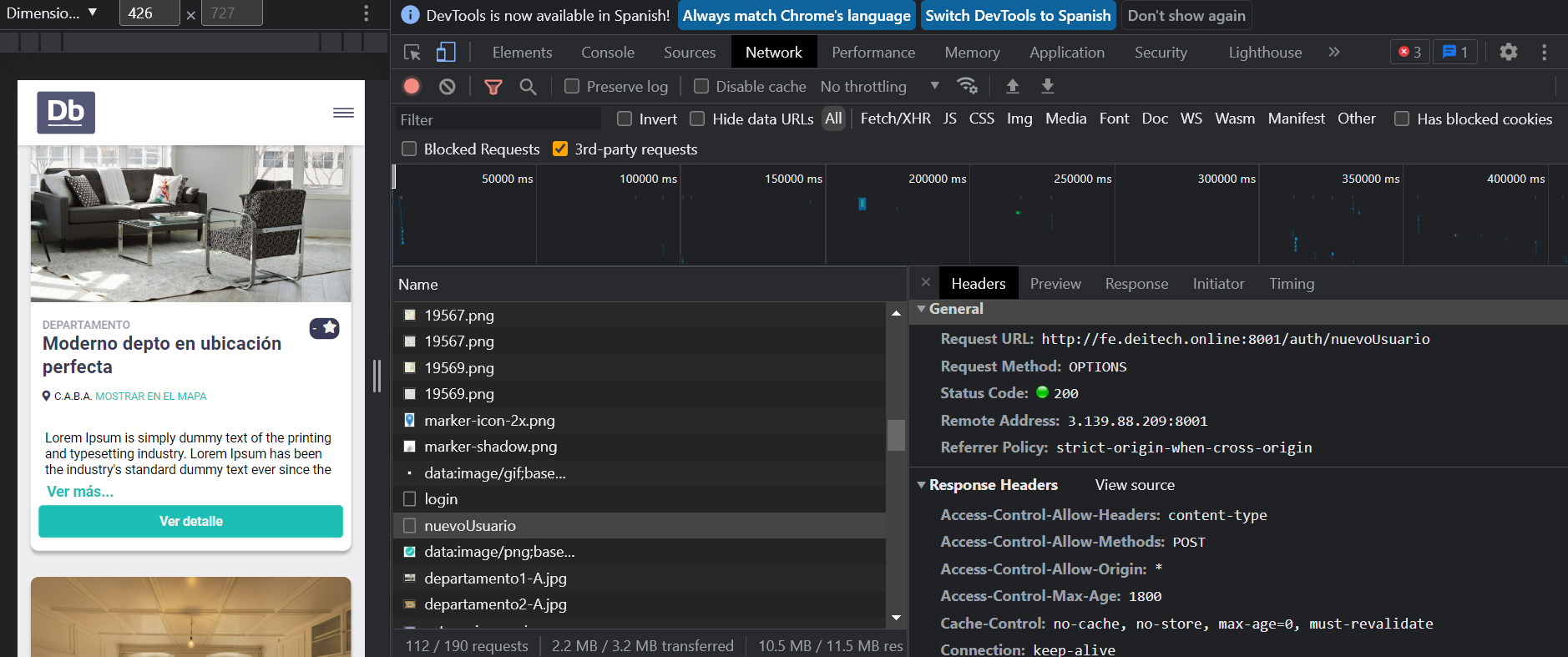
1. ¿Cuál es el objetivo de una prueba de **confirmación**?  
     
   Confirmar el funcionamiento ideal de un componente, la corrección de defectos que haya tenido.

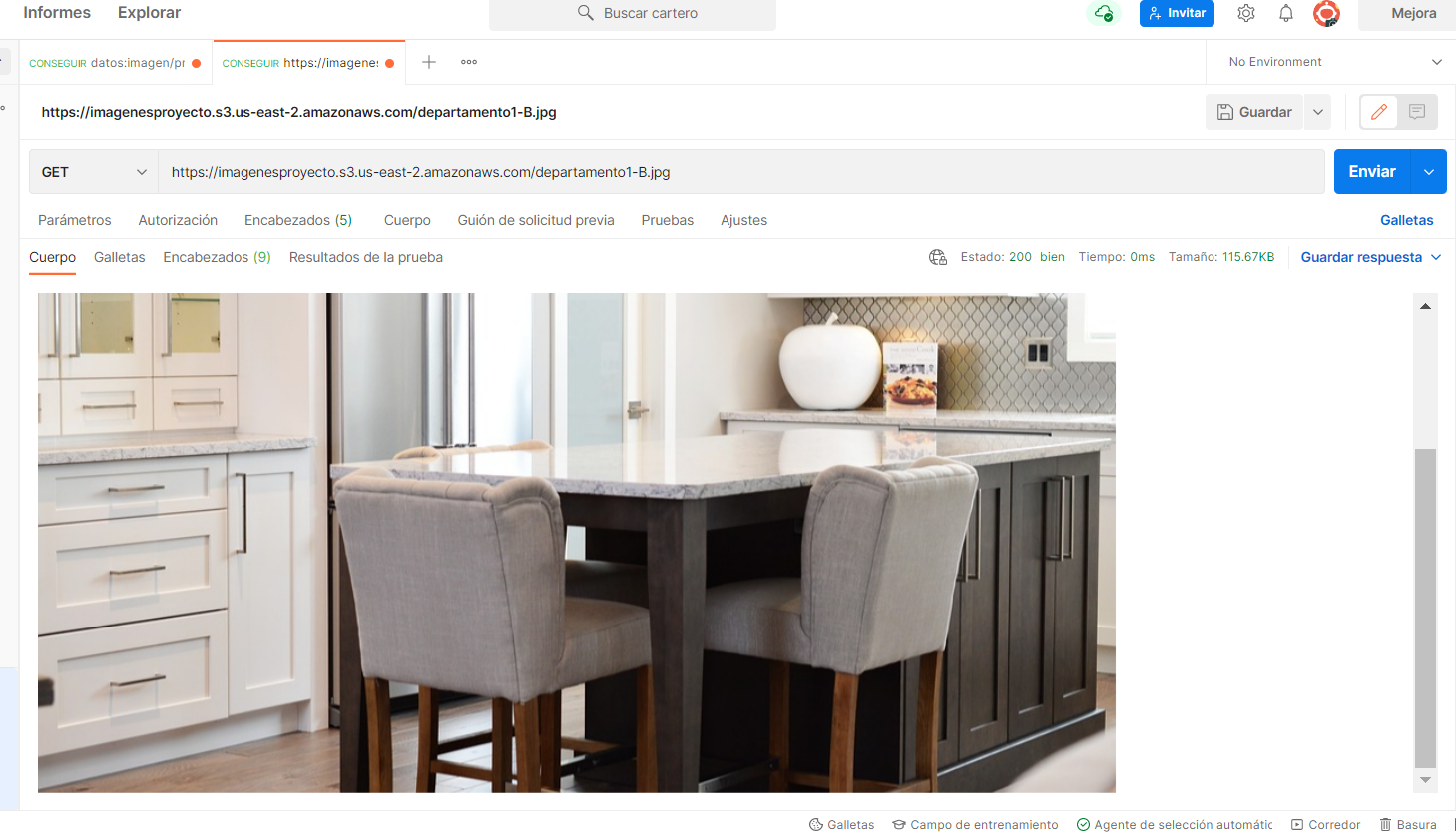
Parte práctica

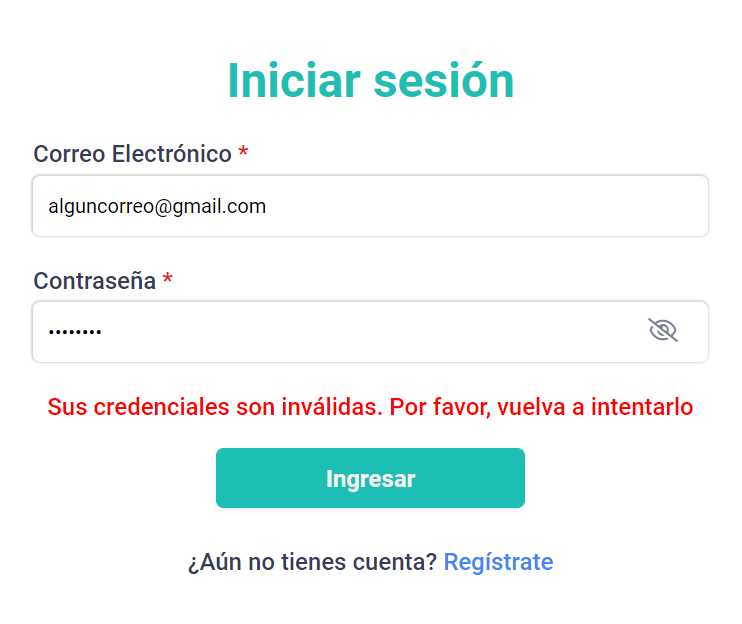
1. ¿Qué nos permite validar el siguiente **test de Postman**?



Está evaluando que en la variable nombre se encuentre o se pueda pasar el String “Virginia”.

1. Identificar una petición GET y una POST de nuestra app [**Digital Booking!**](http://fe.deitech.online/), (Recomendación: Utilizar la herramienta de desarrollo > DevTools) Explicar brevemente de qué se trata una petición POST.  
     
     
     
   Una petición POST es utilizada para envíar un recurso o información a un servidor, es más utilizada de lado del usuario.

Enviando datos para el registro de usuario, **METHOD: POST**Dar la opción añadir a favoritos, **METHOD POST  
  
**Se realiza una petición de tipo GET para pedir toda la información relacionada con un apartamento, entre todos los URL que se generaron seleccione el de una imagen del apartamento donde se muestra que el estado es 200 OK, es decir, que está correcto.

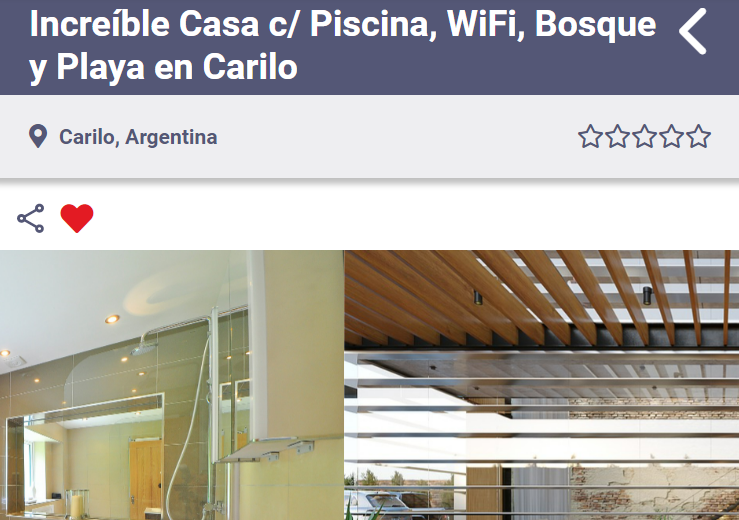
1. Redactar brevemente **1 caso de prueba** **negativo** que aplicarías en la página de [**Digital Booking!**](http://fe.deitech.online/) (**No se requiere escribirlos en formato de template**).  
      
   Título: Credenciales inválidas

Descripción: Se ingresaran credenciales que no están registradas en el sistema   
Precondición: Que el nombre de usuario y la contraseña no estén en el sistema de registro, es decir, no sea un usuario registrado

Pasos:   
a. Ingresar a la página e ir a el menú iniciar sesión

b. En el campo correo ingresar un email que no se haya registrado antes

c. en el campo contraseña ingresar una contraseña que no se haya registrado antes.  
Resultado esperado: Se espera que las credenciales no sean válidas y por lo tanto no pueda iniciar sesión

1. Mencionar **1 defecto** que encuentres en la página de [**Digital Booking!**](http://fe.deitech.online/) (**No se requiere escribirlos en formato de template**).  
     
   No permite asignar una puntuación en estrellas al inmueble   
     
     
     
   Lo que en la página de inicio se refleja como un recurso sin funcionalidad alguna.   
   
2. 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 para cualquier sistema bajo prueba

const persona = {

esAlta: true,

esBaja: false

}

test(“No es baja”, () => {

expect(persona.esBaja).toBeFalsy();

})

Se puede validar utuluzando el matcher Falsy, así si la condiciones es falsa el resultado devuelto con toBeFalsy será true, afirmando que lo que se está evaluando es falso.