**FINAL TESTING – CAMADA 5 – SANCHEZ PULGAR**

PARTE TEORICA:

1. Explique con sus palabras el concepto de Prueba Funcional.

Una prueba funcional es aquella que se basa en probar el comportamiento del sistema. Son aquellas funciones que debe realizar el sistema para su correcto funcionamiento y también describe que hace el software. Y estas pruebas se pueden realizar en todos los niveles de pruebas.

1. ¿Qué técnica de prueba considera correcta para aplicar en una aplicación con muchos menús y workflows? Justifique su respuesta.

La técnica que utilizaría sería la de Transición de estados ya que, iría cambiando de área de trabajo y dicho documento va cambiando de estado en estado dependiendo de dónde este.

1. Mencionar las principales diferencias entre debugging y testing.

Las principales diferencias entre debuggin y testing, es que debuggear se trata del proceso de encontrar, remover y analizar las causas de los fallos en el sistema y consiste en realizar una ejecución de cada parte del programa analizando todas las variables, mientras que por otro lado el testing tiene como fin la búsqueda de errores y es cuando se comprueba que el sistema funcione de acuerdo a lo esperado.

Y una de las principales diferencias es que el testing se realiza antes o después de tener el código, mientras que el debug comienza con la ejecución del código debido a un caso de prueba fallido.

1. Mencionar que pruebas se realizan en el ambiente/entorno de Stage-STG.

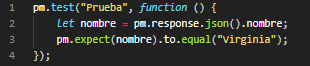
Las pruebas que se realizan en el ambiente de STG son:

* Mantenimiento – seguridad – migración de datos – Interoperabilidad – compatibilidad- gestión de memoria- rendimiento – carga, estrés y escalabilidad – Infraestructura.

1. Indicar cuales son los objetivos específicos de las pruebas de aceptación (Niveles)

Los objetivos de las pruebas de aceptación se basan en; prevenir la propagación de defectos en producción, encontrar defectos, considerar el sistema como un todo, ver si el sistema está completo y por ultimo verificar que los comportamientos funcionales y no funcionales del sistema sean los adecuados.

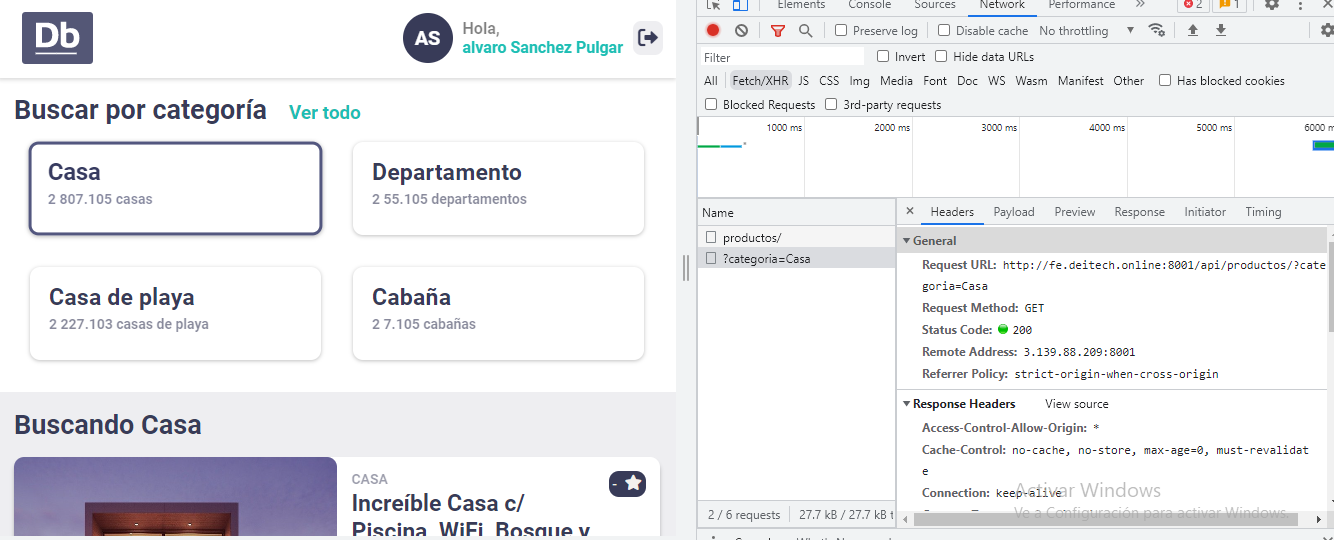
1. ¿Qué nos permite validar el siguiente test de Postman?



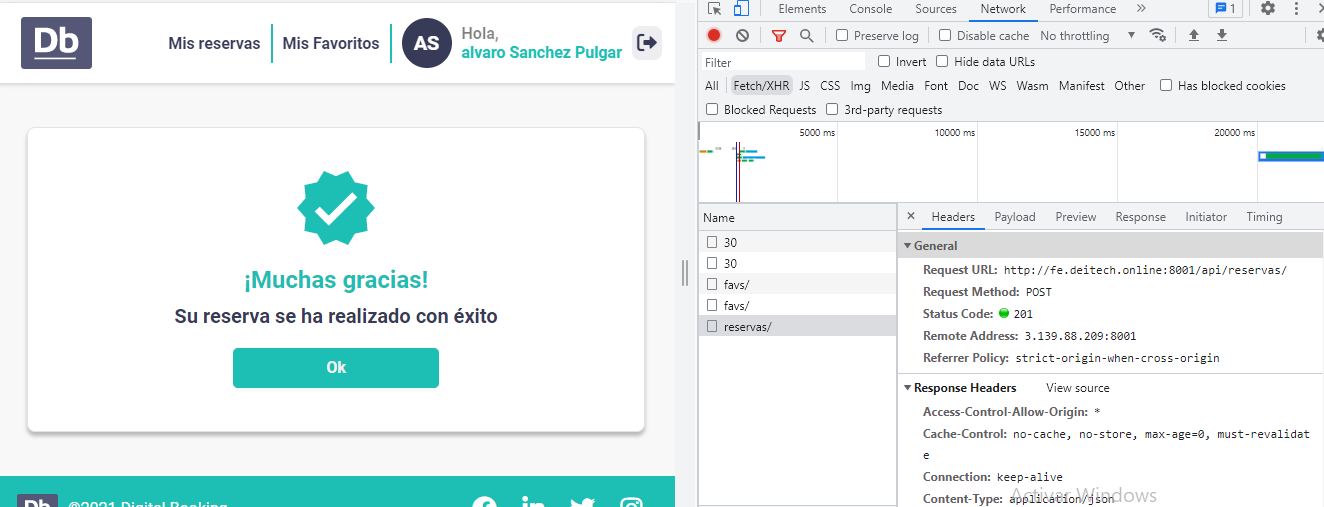
Este test de postman nos indica con el matcher “toEqual” (usado para comprar objetos y todas sus propiedades), se espera que nombre sea igual al String “Virginia”.

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.

GET: Cuando buscamos por categoría en la página Booking y apretamos en casa, hacemos una petición GET para que me muestre todas las casas que hay disponibles en la pagina.



POST: Cuando realizamos una reserva hacemos una petición POST.



La petición POST se da cuando necesitamos agregar datos a nuestra aplicación y a traves de esta solicitud enviamos datos y el API nos devuelve una respuesta que valida que la creación sea exitosa. Los datos para una petición POST no se los pasa por la URL porque no viajan seguros: se pasan por el BODY.

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).

Un caso de prueba negativos de la página Digital Booking seria, cuando nos logueamos con nuestro usuario, entramos a la página y queremos realizar una reserva sin poner las fechas correspondientes a check-in o check-out, en donde luego nos saltara un error que diga <<complete los campos faltantes>>, esta sería una prueba NO FUNCIONAL, son aquellos casos de prueba que validan flujos no contemplados dentro de los requisitos de un sistema bajo prueba.

1. Mencionar 1 defecto que encuentres en la página de [Digital Booking!](http://fe.deitech.online/) (No se requiere escribirlos en formato de template).

Ingresamos con nuestro usuario y contraseña en la página Digital Booking, buscamos cualquier casa que nos guste y antes de querer reservar, clickeamos sobre <<Mostrar en el MAPA>> y en todas las casas disponibles/departamentos/cabañas aparecen en el misma ubicación del mapa.

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 para cualquier sistema bajo prueba.

El matcher que utilizaría para validar un resultado falso seria el de “toBeFalsy”.

Por ejemplo:

t("getObjectDescription('DRAFT') to be false", () => {

const result = getObjectDescription('DRAFT');

expect(result).toBeFalsy(); // toBeFalsy comprueba si el valor es nulo/undefined/false

});