# 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.**

**Duración**: 1 hora 30 minutos.

**Nombre y Apellido:** Jorge Sierra

Parte teórica/práctica

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.

*Se puede utilizar el matcher .toBeFalsy()*

*const {funcion} = require(“<direccion relativa del archivo que contiene los export>”);*

*test(“Test de prueba resultado falso”, () => {*

*const variableAProbar = funcion(<parametros>);*

*expect(variableAProbar).toBeFalsy;*

*});*

1. Mencionar las principales diferencias entre **debugging** y **testing**.

*Debugging:*

*Es realizado por el programador.*

*Se centra en averiguar donde esta el error.*

*Arregla los fallos en el codigo.*

*Se realiza de forma manual.*

*Testing:*

*Es realizado por el tester.*

*Identifica las fallas del codigo.*

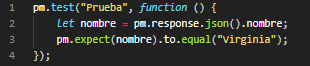
*En este se comprueba que el sistema funcione de acuerdo a lo que se espera.*

*Puede ser manual o automatizado.*

1. Seleccionar las pruebas que considere que se realizan en el ambiente de **Stage-STG:**

* Pruebas de sistema
* Pruebas de regresión
* Pruebas de confirmación
* Pruebas unitarias o de componente
* Pruebas de infraestructura
* Pruebas de gestión de la memoria, compatibilidad e interoperabilidad
* Pruebas de migración de datos
* Pruebas de confirmación
* Pruebas de integración

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

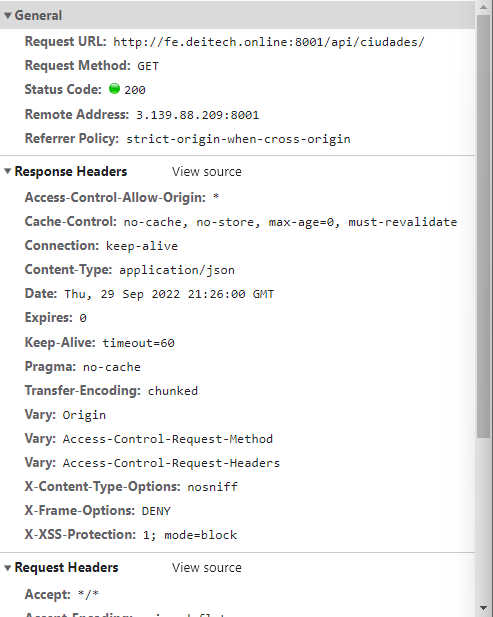


*Permite validar que el nombre obtenido en la respuesta sea igual a Virginia.*

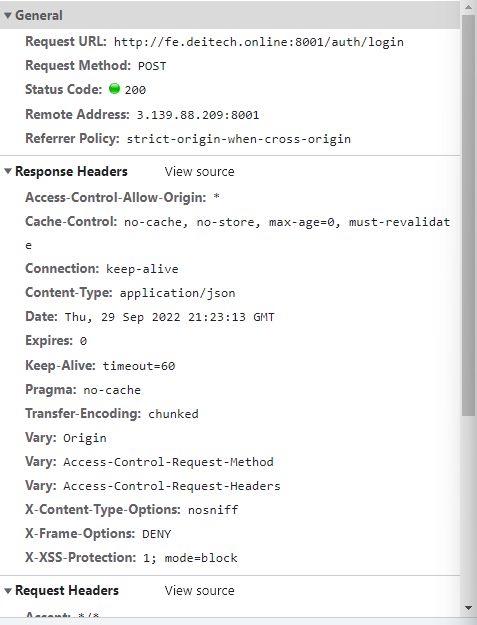
1. Identificar una petición GET y una POST de nuestra app [**Digital Booking!**](http://fe.deitech.online/), (Recomendación: Utilizar la herramienta de desarrollo > DevTools) Explicar brevemente de qué se trata una petición POST.

*GET:*

*Cuando se carga la pagina que obtiene el listado de ciudades*



*POST:*

*Cuando se loguea en la pagina y al enviar los datos de email y contraseña.*  
 

*POST es normalmente usado para enviar información al servidor y que este la añada o tome acciones correspondientes con la información recibida. Es más seguro que el GET.*