# 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:** ..............................

Parte teórica

1. Explique con sus palabras el concepto de Prueba Funcional.
2. ¿Qué técnica de prueba considera correcta para aplicar en una aplicación con muchos menús y workflows?

Justifique su respuesta.

1. Mencionar las principales diferencias entre **debugging** y **testing**.
2. Mencionar que pruebas se realizan en el ambiente/entorno de **Stage-STG**
3. Indicar cuales son los objetivos específicos de las pruebas de aceptación (Niveles)

# **Requerimientos Digital Booking**

**Digital Booking** es nuestro sitio de reservas: [**Ingresá desde acá**](http://fe.deitech.online/)

**Requerimientos:**

* El sistema deberá permitir el registro de un tipo de usuario “cliente”.
* Los datos necesarios serán: nombre, apellido, e-mail y contraseña.
* Se debe validar que todos los campos estén completos y tengan el formato correcto:
  + Nombre (de type=”text”)
  + Apellido (de type=”text”)
  + Email (de type=”email”)
  + Password (de type=”password”- min:6 caracteres)
  + Repetir password (de type=”password”- min:6 caracteres)
* El sistema deberá permitir loguear a los usuarios. En caso de que el e-mail o la contraseña sean incorrectos, se debe mostrar un mensaje de error.
* El sistema deberá permitir a los usuarios realizar una reserva. Los usuarios registrados pero no habilitados no pueden realizar reservas.
* Se habilitará un usuario con el link que llega al e-mail de registro.
* Al momento de realizar una reserva, el sistema deberá enviar un mail al usuario logueado con toda la información detallada de su reserva.
* El sistema deberá permitir a los usuarios realizar búsquedas de alojamientos según fecha y lugar.
* Los lugares se mostrarán mediante un listado y debe poder filtrar por palabras claves, mientras que la fecha no debe permitir seleccionar una fecha anterior a la actual y la estadía mínima permitida debe ser de 1 noche.
* En el caso de que no existan alojamientos disponibles para las fechas deseadas, el sistema deberá indicarle al usuario con un mensaje.
* El sistema permitirá la habilitación del usuario mediante el envío de un SMS.
* El sistema deberá permitir agrupar los alojamientos por tipo y también mostrar los más recomendados por los clientes.
* El sistema deberá permitir al usuario agregar alojamientos a favoritos, como así también, quitarlos.
* El sistema deberá permitir al usuario ver las reservas que realizó con los detalles de la misma.
* La aplicación debe ser responsive, es decir, debe adaptarse a las diferentes resoluciones del navegador hasta llegar a la versión móvil.
* La aplicación debe ser capaz de operar adecuadamente con hasta 100.000 usuarios con sesiones concurrentes.

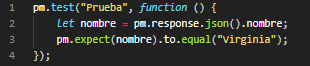
Parte práctica

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

Acceder al contenido de la respuesta de las invocaciones tenemos el objeto pm.response y su método .json() que nos permitirán acceder a los elementos de la respuesta en JSON.

Otro método importante es el que nos permite realizar una comprobación de contenido, este es pm.expect.

Y con el To.equal podemos comparar la igualdad de objetos. En este caso que el nombre sea “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.
2. 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**).
3. Mencionar **1 defecto** que encuentres en la página de [**Digital Booking!**](http://fe.deitech.online/) (**No se requiere escribirlos en formato de template**).
4. 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. toBefalsy- toBeTruthy