

## 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. Indicar cuales son los objetivos específicos de las pruebas de componentes (Niveles)
2. Explique con sus palabras el ciclo de vida de un defecto.
3. Explicar con un ejemplo la diferencia entre una **prueba funcional** y una **prueba no funcional**.
4. Leer el siguiente caso de prueba e indicar en base a qué técnica de prueba fue diseñado.

Título /	Descripción	Precondici	Pasos	Estado de	Creado
----------	-------------	------------	-------	-----------	--------



Nombre	n	ón	#	Acción	Resultado esperado	diseño	por
Cambio de estado de pedido	Cambio de estado del pedido llamado "1xPOLLO" del estado "Nuevo" a "Confirmada"	Tener cuenta de usuario admin en ComidaYA	1	Ir a la URL: <a href="https://ctd-app-resto.herokuapp.com/">https://ctd-app-resto.herokuapp.com/</a>	Visualizar el home de ComidaYA	Finalizado	JG
			2	Ingresar mail (soyadmin@yopmail.com) y contraseña (testqa)	Loguearse en ComidaYA con rol de admin.		
			3	Visualizar que en el módulo "Pedidos", el pedido "1xPOLLO" tiene el estado "Nuevo"	Ubicar el registro con el pedido en cuestión		
			4	Cambiar el estado del pedido a "Confirmada"	Visualizar que se realiza el cambio del estado del pedido "1xPOLLO" en el listado de pedidos		

5. Mencionar que pruebas se realizan en el ambiente/entorno de **QA**

## Requerimientos Digital Booking

**Digital Booking** es nuestro sitio de reservas: [Ingresa desde acá](#)

### 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

6. ¿Este **test de Postman** es correcto para validar si el contenido devuelto es igual al esperado? Justificar tu respuesta.

```
1  pm.test("Nombre Virginia", function () {  
2      let nombre = pm.response.json().nombre;  
3      pm.response.to.have.status("200");  
4  });
```

7. Identificar una petición GET y una POST de nuestra app [Digital Booking!](#). (Recomendación: Utilizar la herramienta de desarrollo > DevTools), Explicar brevemente de qué se trata una petición GET.
8. Redactar brevemente **1 caso de prueba positivo** que aplicarías en la página de [Digital Booking!](#). (**No se requiere escribirlos en formato de template**).
9. Mencionar **1 defecto** que encuentres en el sitio de [Digital Booking!](#) (**No se requiere escribirlos en formato de template**).



10. Si estoy trabajando con **Jest** y quiero validar que el resultado devuelto sea **true**. ¿Qué **matcher** puedo utilizar? Dar un ejemplo de un posible test para cualquier sistema bajo prueba.
-