



**Certified Tech
Developer**
The Ultimate Degree

Testing I

1º Examen

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 tres (3) archivos como máximo por alumno.**

Duración: 1 hora 45 minutos.

Nombre y Apellido: Zoe Jimenez

Parte teórica

1) ¿Cuáles son las actividades principales que se llevan a cabo en el ciclo de vida de las pruebas de software? Mencionar todas y explicar solo una de ellas.

Las actividades principales que se llevan a cabo en el ciclo de vida de las pruebas de software son: planificación, seguimiento y control, análisis, diseño, implementación, ejecución, conclusión y diseño.

Específicamente, el análisis determina "lo que debemos probar" en sí. Se divide en varias subactividades, las cuales son: según el nivel de prueba considera, analizar la base de prueba que corresponda. También, se deben identificar los defectos en las bases de prueba, como lo son las omisiones o las inconsistencias. Se deben identificar los requisitos que elegimos para ser testeados, y definir

cuáles son las condiciones de prueba para uno. Por último, hay que capturar la trazabilidad entre la base de prueba y las condiciones de prueba.

2) ¿Qué es un caso de prueba? ¿Para qué se utiliza?

Un caso de prueba es un documento escrito que nos especifica sobre qué y cómo se debe probar. Cubre al software en profundidad y con mucho detalle.

3) Indique con un ejemplo la diferencia entre los conceptos de caso de prueba positivo y caso de prueba negativo.

La diferencia entre el testing positivo y el negativo es el tipo de flujo que se busca. Utilizando de ejemplo realizar el login a una página, se trataría de un testing positivo si ingresáramos credenciales validas buscando que el sistema nos permita ingresar. Por el otro lado, seria testing negativo ingresando credenciales incorrectas, esperando que no nos permita ingresar.

4) ¿A qué grupo de pruebas pertenece la prueba de tabla de decisión? Explicar brevemente su uso y aplicación.

La tabla de decisiones pertenece al grupo de pruebas de caja negra. Se utiliza principalmente cuando tenemos un sistema que debe implementar pruebas de combinación que tienen reglas complejas. Trabajamos con condiciones y acciones resultantes, las cuales forman las filas de la tabla. Las columnas forman combinaciones de condiciones y la acción resultante de esta combinación. Esta tabla ayuda a verificar las reglas complejas que se debían implementar están correctas.

5) ¿Cuál es la diferencia entre las pruebas que se pueden realizar en el ambiente de QA con respecto a las pruebas que se pueden realizar en el ambiente de UAT? En el ambiente QA el tester lleva a cabo las primeras pruebas de funcionalidad, por lo que disminuye incidencias en el futuro. En cambio, en el ambiente UAT, se llevan a cabo pruebas de aceptación por parte del usuario o el cliente, y evalúa tanto que los cambios realizados hayan sido los solicitados, así como la accesibilidad del software.

6) ¿Cuál es la principal diferencia entre las pruebas de humo y las pruebas regresión?

Las pruebas de humo verifican funcionalidades básicas de una implementación, a diferencia de las pruebas de regresión, que verifican la continuidad en el buen funcionamiento de otras funcionalidades al agregarse una nueva, para poder mantener la integridad del sistema.

Parte práctica

Enunciado

Se tiene un software denominado **Digital Booking!** El cual nos permite buscar las mejores ofertas de alojamientos y realizar la correspondiente reserva. El mismo cuenta con las siguientes funcionalidades: un registro, un login, un buscador de alojamientos y un módulo de reservas. <http://fe.deitech.online/>

Usted ha sido seleccionado para probar esta app, desde la visión del usuario **CLIENTE** y que cuenta con los siguientes requerimientos mínimos:

- El sistema deberá permitir registrar un tipo de usuario cliente. Los datos necesarios son: nombre, apellido, email 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 email 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 habilita un usuario con el link que llega al email 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. El lugar debe ser un listado y debe poder filtrar por palabras claves, mientras que, la fecha no debe permitir

seleccionar una fecha anterior 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 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.

Consignas

7) Redactar un (1) caso de prueba aplicando partición de equivalencia y otro caso de prueba aplicando la técnica de tabla de decisión. **Para redactar el caso de prueba, debés utilizar el template desarrollado en clase.**

8) Reportar un (1) defecto del sistema en cualquiera de los menús disponibles. **Para dicho reporte, debe utilizar el template de defectos visto en clases.**

9) Escribir un (1) caso de prueba positivo y un (1) caso de prueba negativo. **No se requiere escribirlos en formato de template.**

Caso de prueba positivo:

Requerimiento: El sistema deberá permitir loguear a los usuarios. En caso de que el email o la contraseña sean incorrectos, se debe mostrar un mensaje de error.

Realizar el logueo con un mail y una contraseña válidas, y esperar que nos los permita.

Caso de prueba negativo:

Requerimiento: El sistema deberá permitir a los usuarios realizar búsquedas de alojamientos según fecha y lugar. El lugar debe ser un listado y debe poder filtrar

por palabras claves, mientras que, la fecha no debe permitir seleccionar una fecha anterior actual y la estadía mínima permitida debe ser de 1 noche.

Realizar la búsqueda de un alojamiento con una estadía menor a 1 noche, que es lo permitido por el sistema, y esperar que no nos lo permita es un ejemplo de testing negativo.

10) Redactar brevemente un caso de prueba no funcional. **No se requiere escribirlos en formato de template.**
