**1° EXAMEN - TESTING I (CTD) - MARZO DE 2022 [0222TDTE1M1C1LAED1021FT]**

****

# Testing I

# **1° 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:** Ivan Rodriguez Russo

**Parte teórica**

1) Explique con un ejemplo la diferencia entre error, defecto y falla.

**1**.La Diferencia que se halla entre estos 3 conceptos es que error es una acción humana que produce un resultado incorrecto, el defecto es la causa de este error y el fallo es lo que produce el defecto al ejecutarse. Por ejemplo:

-Error : Un requerimiento mal especificado

-Defecto: Una función de login tiene las variables declaradas incorrectamente.

-Fallo : Visualización de un mensaje de alerta que el desarrollador no definió/programó.

2) Defina el concepto de “Suite”. ¿En qué escenario podrías aplicar una Suite de Regresión?

**2.** Suite puede ser definida como un grupo de casos de pruebas que se utilizan para ejecutar estas mismas. Aplicaría una Suite de Regresión luego de modificar agregar nuevas funciones, verificando que todo esto suceda sin fallas y funcione a la perfección.

3) Explique la etapa de “Corregido” dentro del ciclo de vida de los defectos.

**3.** En la etapa de Corregido se realizan los cambios de código para solucionar los defectos.

4) Explique el concepto de Técnica de los 3 amigos / Mesa de 3 patas, y porqué es importante utilizarlo.

**4.** El concepto de mesa de 3 patas se basa básicamente en 3 actores en el cual cada uno tiene un rol distinto, en este encontramos al QA, Software Developer y al Business Analyst.

Es importante utilizarlo ya que cada uno al ocupar un rol distinto y tener distintas responsabilidades, estas luego se ponen en común a través de una reunión en la cual se da el punto de vista del software en desarrollo. A la mesa no puede faltarle ninguna pata.

5) ¿Cuál es la diferencia principal entre pruebas estáticas y dinámicas?

**5.** La principal diferencia entre estas dos es que las Pruebas ESTÁTICAS no requieren que el software esté en ejecución en cambio las Pruebas DINÁMICAS si requieren dicha ejecución.

6) Explica cuáles son los niveles de pruebas. Elegí uno y menciona un ejemplo de una prueba en ese nivel.

**6.** Los niveles de prueba son:

* **Prueba Unitaria o de componente:** Se basa en la lectura y verificación de que el código se encuentre correctamente, generalmente el desarrollador que lo escribió realiza esta prueba.
* **Prueba de integración:** Se basa en las interacciones dadas entre componentes o sistemas.
* **Prueba de sistema:** Se basa como su nombre indica en el comportamiento del sistema tanto funcional como no funcional.
* **Prueba de aceptación:** Se basa en validar que el sistema se encuentre completo y de la forma en que se esperaba

**Ejemplo de Prueba de Aceptación:**

Que un requerimiento para loguearse a una web sea ingresar con credenciales válidas y que al intentar ingresar con una clave que no es la mia esta web indique que la clave es incorrecta demostrando que funciona correctamente.

**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 un (1) caso de prueba aplicando la técnica de tabla de decisión. **Para redactar el caso de prueba, debes utilizar el template desarrollado en clase**.

8) Reportar un (1) defecto del sistema en cualquiera de los menúes disponibles. **Para dicho reporte, debes 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**.

**9.** **Caso de Prueba positivo**: Loguear con datos válidos, vamos a ingresar a la web y a loguearnos con nuestras credenciales correctas, el resultado esperado es visualizar la página principal de Digital Booking.

**Caso de Prueba Negativo**: Loguear con clave Invalida, vamos a ingresar a la web y en el login escribir la credencial del mail correcta pero no así la de la clave, el resultado esperado es que aparezca un mensaje de error el cual indique que la clave es incorrecta.

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

**10.** Un ejemplo de prueba no funcional sería: sobrecargar el sistema con la capacidad de 100.000 usuarios con sesiones concurrentes para controlar el rendimiento de la web con esta cantidad de usuarios la cual en los requerimientos está confirmada.