



Leer atentamente las siguientes consignas, y realizar los ejercicios de acuerdo a lo solicitado para el siguiente sistema. Enviar la resolución del parcial a este [formulario](#) con la siguiente nomenclatura: [Apellido, Nombre - DD/MM/YYYY]

No se aceptarán links de Drive, sólo documentos adjuntos (máximo 5). Caso contrario, el examen no será considerado para su corrección.

Nota aclaratoria: al enviar el parcial, 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. **Cada pregunta vale 1 (un) punto, de un total de 10 (diez).**

Duración: 1 hora 45 minutos.

Parte teórica

1) Mencionar tres (3) principios de Testing y explicarlos brevemente.

- a) *La prueba muestra la presencia de defectos, no su ausencia. El objetivo del tester es buscar los defectos, demostrarlos, NO es demostrar la ausencia de ellos o comprobar que el software no tiene defectos, por que eso es una falacia.*
- b) *Cuidado con la prueba del pesticida: Si recurrimos siempre a los mismos metodos o tecnicas para testear, los defectos del software se vuelven resistentes a estas tecnicas, con lo cual debemos ir cambiando las pruebas para una deteccion mas eficiente.*
- c) *La ausencia de errores es una falacia, siempre hay errores, el desconocimiento de estos y su ausencia son dos cosas diferentes, siempre debemos asumir que hay errores y el trabajo del tester es encontrarlos.*

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

Caso de prueba es el documento que indica como se va a testear, diseñado idealmente por un test manager o una persona idonea y ejecutado por testers generalmente, excepto aquellos que requieran un conocimiento especializado y tecnico como por ejemplo relacionados con el codigo. Es decir de caja blanca.

3) Definir brevemente un defecto y quién se responsabiliza por su reporte. ¿Qué diferencia existe con los conceptos de error y falla?

Un defecto puede ser que al registrarnos, al introducir un mail nos permita hacerlo con una serie de caracteres invalidos, por ejemplo que no tenga un arroba ni un .com .org .net, etc. Es decir que se introduzcan parametros que afectarian al funcionamiento o flujo correcto.

Un defecto viene del codigo por ejemplo, generado por un Error humano debido a una mala interpretacion de por ejemplo un requerimiento. Mientras que un fallo se genera a causa de uno o mas defectos, causando justamente un fallo en el sistema. Un fallo es a nivel general de todo el sistema.

4) ¿A qué gran grupo de pruebas pertenece la prueba de partición de equivalencia? Explique brevemente su uso y aplicación.

Particion de equivalencia pertenece al grupo de pruebas de Caja negra por que No podemos observar cómo fue construida, no vemos el código, no sabemos su arquitectura, no tenemos nociones más que la interfaz que estamos interactuando.

En la apticion de equivalencia se dividen los datos en particiones conocidas como clases de equivalencia donde cada miembro de estas clases o particiones es procesado de la misma manera.

Las validas nos mostrara las particiones que el software acepte, las invalidas las que no acepte.

Ejemplo: RGB son los colores que acepta, cualquier otro es invalido.

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

Las pruebas estaticas son aquellas que evaluamos de forma manual, en un tiempo especifico y determinado, por eso se llaman estaticas.

Revisamos documentacion, una linea de codigo, cosas puntuales que no necesitan la ejecucion del software, pueden ser llevadas a cabo por gente que entienda del codigo, es decir con conocimiento tecnico por lo general.

Las pruebas dinamicas estan relacionadas con la ejecucion del software, sistema o componente. Puede ser basadas en experiencia, se complementan con las estaticas debido a que encuentran diferentes tipos de defectos. Las pruebas dinamicas pueden ser llevadas a cabo por personas sin conocimiento tecnico especializado como las pruebas estaticas.

Las proebas dinamicas necesitan que el software este corriendo.

6) Explica brevemente qué pruebas puede realizar en cada ambiente de desarrollo de software.

En el ambiente de DEV, se puede realizar pruebas que necesiten conocimiento tecnico sobre ciertos temas como Arquitectura del software, codigo, etc. Debido a que este es el ambiente de los desarrolladores del software. Ejemplos:

Prueba unitarias o de componente

Prueba de integracion.

En el ambiente QA entran en accion los testers, donde pueden trabajar en un ambiente con el software corriendo y no se necesita del conocimiento requerido en el ambiente DEV. Ejemplos:

Pruebas funcionales.

Pruebas de Humo.

El ambiente UAT Permite a los usuarios del cliente poder verificar que los cambios realizados son los que realmente se solicitaron, evaluando a su vez accesibilidad y usabilidad. Ejemplos:

Prueba de aceptacion.

Prueba de usabilidad.

En el entorno de STG se requiere una configuracion tecnica identica a la que nos encontraremos en entorno de produccion, busca emular este entorno con el fin de probar las actualizaciones y asegurar que estas no corromperan la aplicación en los servidores en produccion cuando sean desplegadas. Minimizamos caidas del sistema, cortes de los servicios de produccion, etc. Ejemplo:

Pruebas de mantenimiento.

Pruebas de seguridad.

El ambiente de produccion no cuenta con pruebas asignadas.

Parte práctica

Enunciado (Relevamiento)

Se tiene un software denominado **Comida Ya!** El cual nos permite realizar la compra y venta de productos. El mismo cuenta con las siguientes funcionalidades: un registro (tanto para un usuario administrador y un usuario cliente), un Login, un ABM de productos y pedidos (Alta-Baja-Modificación) y Listar productos y pedidos.

<https://ctd-app-resto.herokuapp.com/admin/dashboard/>

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

- El sistema debe 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.
- El sistema debe permitir loguear a los usuarios. En caso de que el email o la contraseña sean incorrectos, se debe mostrar un mensaje de error.
- El usuario cliente podrá ver todos los productos disponibles y agregarlo al carrito de compras. Al hacer clic en un producto se debe mostrar un mensaje indicando que el producto fue añadido al carrito.
- El usuario cliente podrá ver su carrito de compras, elegir su forma de pago y confirmar la compra. Si el carrito no posee productos el botón "confirmar compra" debe aparecer deshabilitado.
- 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 300.000 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, debes utilizar el template desarrollado en clase.**

8) Reportar dos (2) defectos del sistema en cualquiera de los menús 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.**

Caso positivo: deberíamos poder registrarnos rellorando los datos correspondientes en la pagina de registro, completando los campos con formatos validos, es decir

nombre y apellido con texto requerido, email con un formato que incluya un arroba y un dominio como .com .net o .org, y una contraseña con una longitud minima de 8 caracteres y maximo de 10 caracteres.

Ingresando los datos correctamente deberia permitir el registro.

Caso negativo: Deberiamos impedirnos el registro y avisar con un cartel que campos no se completaron correctamente, al intentar registrarnos rellenando los datos del registro con un formato invalido, como una contraseña que exceda el limite maximo de caracteres, un mail sin arroba o sin completar y dejando en blanco el nombre y apellido.

10) Redactar brevemente una (1) prueba funcional y una (1) prueba no funcional.
No se requiere escribirlos en formato de template.

Prueba funcional: Deberiamos poder ingresar con un usuario ya registrado anteriormente, completando los campos de usuario y contraseña de forma correcta y darnos acceso al portal de usuarios despues de pulsar el boton de Iniciar sesion.

Prueba no funcional: Si intentamos ingresar con una contraseña incorrecta a un usuario, al cabo del quinto intento consecutivo en un lapso de 10 minutos desde el primero al ultimo intento, deberia bloquearse el acceso al usuario desde esa terminal y un mail deberia ser enviado al asociado con ese usuario para advertir de intentos repetidos de ingreso al portal, con informacion de la hora de los intentos.
