**1° EXAMEN - TESTING I (CTD) - NOVIEMBRE 2021**

****

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

**Duración**: 1 hora 45 minutos.

**Nombre y Apellido:** Juan Manuel Arango Tobón

**Parte teórica**

1) Mencionar dos (2) principios de Testing y explicarlos brevemente. (2 renglones)

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

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

4) ¿A qué gran grupo de pruebas pertenece la prueba de partición de equivalencia? Explicá brevemente su uso y aplicación. (2 renglones)

5) ¿Cuál es la diferencia principal entre pruebas estáticas y dinámicas? (2 renglones)

6) Explica cuáles son los ambientes de desarrollo de software. Elegí uno de ellos y menciona qué pruebas se pueden realizar en dicho ambiente.

**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/>

Fuiste seleccionado para probar esta app, desde la visión del usuario **CLIENTE** y contás 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, debés utilizar el template desarrollado en clase.

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

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

SOLUCIÓN:

**1) Mencionar dos (2) principios de Testing y explicarlos brevemente. (2 renglones)**

* **“La prueba muestra la presencia de defectos, no la falta de ellos”:** Esto quiere decir que no se puede probar de manera exacta que no hay defectos en algo, por el contrario, una prueba permite hallar la existencia de defectos y siempre que se mire se puede encontrar un fallo, u error, que a simple vista no se percibe.
* **“La ausencia de errores es una falacia”:** Esto quiere decir, que siempre van a existir errores, algunos descubiertos por las pruebas hechas y muchos mas que faltan por ser descubiertos que salen a la luz al mirar las distintas perspectivas de una situación o caso (A veces imperceptibles).

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

**Caso de prueba:** Se puede entender un caso de prueba como un conjunto de parámetros y lineamientos que se tienen en cuenta al momento de verificar una característica o funcionalidad en conjunto de un sistema o aplicación de software. Un caso de prueba se puede representar como un documento que proporciona la información necesaria sobre que se va a probar y para que se va a hacer.

Los casos de prueba son muy importantes en el mundo del desarrollo de software, ya que son el manual que permite comprobar y garantizar la funcionalidad tanto modular como completa de un sistema o aplicación y es un principio que prima en la calidad de un buen software.

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

* **Defecto:** Se puede entender como defecto el resultado causado en el software a causa de un error sea de una persona, de un calculo, etc. esto puede causar un fallo en el sistema o en el software lo que hace que se tenga un mal funcionamiento, o un funcionamiento defectuoso diferente a lo esperado tanto a nivel de sistema, como de componente.

Hay una confusión entre los términos de **falla** y **error**, para aclarar esto, hay que tener en cuenta que un error es generado por una persona, un calculo y/o una mala decisión, y esto provoca defectos en el software, y por otro lado, cuando estos defectos son ejecutados o incluidos dentro de las pruebas de software generan lo que se conoce una falla.

**4) ¿A qué gran grupo de pruebas pertenece la prueba de partición de equivalencia? Explicá brevemente su uso y aplicación. (2 renglones)**

El tipo de pruebas conocidas como “pruebas de partición de equivalencia”, pertenecen al grupo de **pruebas de caja negra**. Estas son una técnica que clasifica las entradas del software como clases (entiéndase como entrada lo que pueda causar una salida positiva o negativa).

En este tipo de pruebas se dividen en clases, ya que cada clase prueba la misma cosa, y con esto se puede saber si un componente de una clase genera error, se puede deducir que los demás dentro de la misma también. Y se puede saber que la idea principal de este tipo de prueba es identificar los distintos casos de prueba de manera modular, usando un elemento de cada clase que de equivalencia que se ha generado.

**5) ¿Cuál es la diferencia principal entre pruebas estáticas y dinámicas? (2 renglones)**

Este tipo de pruebas, permiten evaluar la calidad del software, permitiendo identificar y tratar defectos de forma temprana para evitar daños a futuro, estas se complementan la una con la otra y pueden ser similares, sin embargo, la prueba **estática**, es una evaluación o análisis que se basa en los productos de trabajo, o en el código en si mismo, antes de ser ejecutado y puesto en funcionamiento, mientras que la prueba **dinámica**, requiere la ejecución del software o de los componentes que se están evaluando o probando, logrando encontrar defectos que en las pruebas estáticas seria imposible hallar.

**6) Explica cuáles son los ambientes de desarrollo de software. Elegí uno de ellos y menciona qué pruebas se pueden realizar en dicho ambiente.**

En el mundo del desarrollo, los ambientes, son el conjunto de procedimientos y herramientas que son usadas por los desarrolladores, sea para programar, probar, integrar o validar programas y componentes, esto puede funcionar como un espacio de trabajo donde los cambios se implementan y complementan desde varios puntos y/o entornos.

* Entorno de integración: Este es el entorno donde se permite la fusion o unión de los diferentes componentes que se llevan a cabo en el software y permite comprobar que los diferentes trabajos no interfieran entre si, en este tipo de entorno se pueden generar varios tipos de pruebas como las siguientes:
  + Pruebas unitarias
  + Pruebas de integracion
  + Pruebas funcionales