Por favor indique dentro del documento y/o en el nombre de los archivos que adjunte, su nombre y apellido.

1. Detectar, marcar con un círculo, y explicar brevemente los defectos, si es que hubiere, en el siguiente ejemplo de una consulta de movimientos de una cuenta (tener en cuenta la unificación de criterio y que sólo se detallan registros de movimientos, sin importar el saldo de la cuenta):



FECHA DESCRIPCION IMPORTE PESOS IMPORTE DOLARES



21/11/2011 Extracción -1000.00



12/21/2011 Dep&oacute;sito 1000,00



01/10/2011 Débito por transferensia 500.00



1. Falta de ortografía en DESCRIPCION (DESCRIPCIÓN)
2. Falta de ortografía en DOLARES(DÓLARES)
3. Formato de fecha no unificado en la segunda fila
4. Error de formato en la segunda línea al escribir Depósito
5. Error de ortografía en transferensia (transferencia)
6. Error de lógica en el importe de la tercera fila (Dice débito y acredita el importe)



1. La fecha de los movimientos está desordenada, no sigue un orden ascendente ni descendente
2. Detallar los campos que se deberían utilizar al informar un defecto y realice el reporte de todos los errores encontrados en el primer enunciado:

1)Título: Falta de ortografía

Descripción: En la columna DESCRIPCIÓN, hay un error de ortografía, figura DESCRIPCION

Pasos para reproducir: Ingresar con usuario y contraseña, luego realizar una consulta de movimientos.  
Criterio de aceptación: El campo debería decir DESCRIPCIÓN.

Entorno: Test

Prioridad: 3  
Severidad: 1

2)Título: Falta de ortografía

Descripción: En la columna IMPORTE DÓLARES, figura IMPORTE DOLARES.

Pasos para reproducir: Ingresar con usuario y contraseña que tenga cuenta en dólares, luego realizar una consulta de movimientos.  
Criterio de aceptación: El campo debería decir IMPORTE DÓLARES

Entorno: Test  
Prioridad: 3  
Severidad: 1

3)Título: Formato de fecha no unificado  
Descripción: Al realizar una consulta de movimientos, no está unificado el formato de fechas, algunas figuran en formato DD/MM/YYYY y otras en MM/DD/YYYY  
Pasos para reproducir: Ingresar con usuario y contraseña, luego realizar una consulta de movimientos.  
Criterio de aceptación: Al realizar una consulta de movimientos, la fecha de los mismos debe seguir el formato DD/MM/YYYY

Entorno: Test  
Prioridad: 2  
Severidad: 3

4)Título: Error de formato  
Descripción: Al realizar una consulta de movimientos, hay un error de formato en los movimientos en dólares.  
Pasos para reproducir: Ingresar con usuario y contraseña que tenga cuenta en dólares, luego realizar una consulta de movimientos.  
Criterio de aceptación: No debería haber errores de formato al utilizar palabras con tilde, debería usar el formato UTF-8

Entorno: Test  
Prioridad: 1  
Severidad: 4

5)Título: Error de ortografía  
Descripción: En la descripción de los débitos por transferencias, figura: Débito por transferensia.  
Pasos para reproducir: Ingresar con usuario y contraseña que tenga débitos por transferencias, luego realizar una consulta de movimientos.  
Criterio de aceptación: La descripción de los débitos por transferencia debe decir: Débito por transferencia.

Entorno: Test  
Prioridad: 3  
Severidad: 3

6)Título: Error de lógica en el importe  
Descripción: Al realizar una consulta de movimientos de una cuenta con débitos por transferencia, figura como una acreditación en vez de un débito.  
Pasos para reproducir: Ingresar con usuario y contraseña que tenga débitos por transferencias, luego realizar una consulta de movimientos.  
Criterio de aceptación: El importe en pesos de un débito por transferencia debe ser negativo.

Entorno: Test  
Prioridad: 1  
Severidad: 5

7)Título: Fecha de movimientos desordenada  
Descripción: Al realizar una consulta de movimientos, la lista está desordenada.  
Pasos para reproducir: Ingresar con usuario y contraseña, luego realizar una consulta de movimientos.  
Criterio de aceptación: Al realizar una consulta de movimientos, la lista debe estar ordenada por fecha de manera descendente.

Entorno: Test  
Prioridad: 1   
Severidad: 5

1. Defina lo que son las siguientes técnicas de test y diseñe un caso de prueba para cada una (puede armar un escenario de ejemplo para utilizarlo para explicar cada técnica):

Pruebas Funcionales: Son aquellas pruebas encargadas de comprobar que el sistema cumple su función.

Voy a usar como ejemplo un sistema de distribución porque es en el que más trabajé.

Caso de prueba: Verificación de filtros para usuario con rol de admin.

Descripción: Este caso de prueba tiene el fin de garantizar que los filtros funcionen correctamente y no tengan fallas entre si.  
Configuración inicial: Ingresar al sistema con un usuario cuyo rol sea administrador.

Detalle:

- Presionar buscar sin ingresar ningún filtro  
 Verificar que el sistema muestre un cartel que solicite ingresar un filtro válido y el sistema no se rompa.  
- Buscar las guías desde el 01/01/2025 hasta el 03/03/2025.  
 Verificar que las guías mostradas estén dentro del rango de fechas.

- Ingresar un rango inválido de fechas, EJ: desde el 03/03/2025 hasta el 01/01/2025, letras, símbolos, etc.  
 Verificar que se muestra un mensaje de error solicitando una fecha válida y el sistema no se rompa.

Pruebas No Funcionales: Son aquellas encargadas de demostrar que tan bien cumple su función el sistema, se enfocan más bien en el cómo, y las funcionales, en cambio, en el qué hace. Por ejemplo, cuestiones de usabilidad del usuario o performance de la aplicación.

Caso de prueba: Facilidad para completar el formulario de solicitud de recolección.

Descripción: Este caso de prueba tiene el fin de mejorar la experiencia del usuario al completar el formulario.

Configuración inicial: Ingresar al sistema con un usuario cuyo rol sea cliente.  
Detalle:  
- Observar los campos del formulario.  
 Verificar que estén alineados y que no sobre espacio innecesario entre los mismos.  
- Completar los campos del formulario.  
 Verificar que las ayudas al ingresar un dato sean útiles y no confusas.

Pruebas de Carga y Estrés: Son aquellas pruebas no funcionales, que se encargan de verificar cuanto tarda en cargar una página, o cual es la carga de consultas que soporta. Por ejemplo, al realizar una transacción, verifica cuánto tarda en realizarse la misma, y si está alineada o no a lo que necesita el negocio.

Caso de prueba: Resistencia del sistema frente a cargas inusuales de usuarios.

Descripción: Este caso de prueba tiene el fin de cuantificar cuantos usuarios soporta como máximo el sistema hasta romperse, y desde ahí, evaluar si es suficiente o necesita expandirse.

Configuración inicial: Ejecutar el login y acciones de usuarios en el sitio a través de Jmeter.

Detalle:  
- Probar el login y acciones con 100 usuarios.

Verificar que las acciones se ejecuten correctamente.

- Aumentar el volumen de usuarios hasta que el sistema falle.  
 Analizar si la cantidad de usuarios que soporta es la indicada.

1. Se necesita realizar un cálculo de bonos para los empleados. No puede ser negativo, pero puede ser cero. El bono está basado en la duración del empleo. Una persona puede ser empleado por menos o igual a 2 años, más que 2 años pero menos de 5 años, 5 a 10 años, o más de 10 años. Dependiendo del período del empleo, un empleado podrá obtener: ningún bono o un bono del 10%, 25% o 35%. ¿Cuántas particiones equivalentes se necesitan para testear el cálculo de bonos? Justifique su respuesta.

* 3
* 5
* 8
* 4

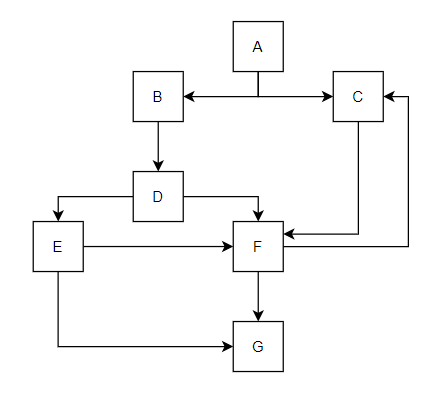
Se necesitarían 4 particiones equivalentes, la primera de 0 a 2 años, con ningún bono, la segunda de 2 a 5 años, con un bono del 10%, la tercera de 5 a 10 años, con un bono del 25% y, por último, la cuarta de más de 10 años con un bono del 35%

1. Uno de los objetivos del test para el proyecto, es tener una cobertura del 100% de los flujos de decisión. Los siguientes tres tests han sido ejecutados para el siguiente diagrama de flujo detallado debajo.

El test A cubre los caminos: A, B, D, E, G.

El test B cubre los caminos: A, B, D, E, F, G.

El test C cubre los caminos: A, C, F, C, F, C, F, G.



¿Cuál de las siguientes afirmaciones es correcta, en relación a la cobertura de los flujos de decisión? Justifique su respuesta.

* La decisión D no ha sido testeada completamente.
* El 100% de la cobertura ha sido alcanzada.
* La decisión E no ha sido testeada completamente.
* La decisión F no ha sido testeada completamente.

La decisión F no ha sido testeada completamente, debido a que le faltó atravesar el flujo A,B,D,F,G sin pasar por E primero

6- ¿En qué te basarías para estimar el esfuerzo que te llevará crear y ejecutar los casos de prueba de un requerimiento?

Primero que nada, en la complejidad del requerimiento, dependiendo de cuantas o en que medida sean complejas las reglas de negocio asociadas.

También en la criticidad del requerimiento, ya que si forma parte de un módulo crítico hay que prestarle especial atención para que no genere fallas importantes en producción, aparte de ser más propenso.

En el tiempo disponible para que salga a producción, ya que hay que ver cuantos casos de prueba pueden automatizarse desde un inicio, cuantos posponer por un límite de tiempo, en el cual quizás conviene testearlo manualmente para reducir el tiempo de entrega.

7- Ejercicios API Testing: Resuelva tratando de abarcar la mayoría de casos de prueba posible y explique qué es lo que busca validar en cada caso. Puede realizar casos de prueba manual (para esto utilice el formato: Título; Prioridad Precondiciones; Pasos; Resultados esperados) o puede escribir los casos automatizados (postman).

Esta solicitud (GET) {{LOCATION}}/bonds/{{AL30}} obtiene la siguiente respuesta:

{

"shortName": "AL30",

"name": "Bono (canje 2020) 2030 u$s Step-up (Ley Argentina) (AL30)",

"debtType": "Título Público",

"links": {

"coupons": "{{LOCATION}}/bonds/ecfe885d-8163-4222-acb1-f7020cf05a4a/coupons",

"identifiers": "{{LOCATION}}/bonds/ecfe885d-8163-4222-acb1-f7020cf05a4a/identifiers",

"self": "{{LOCATION}}/bonds/ecfe885d-8163-4222-acb1-f7020cf05a4a",

"issueConditions": "{{LOCATION}}/bonds/ecfe885d-8163-4222-acb1-f7020cf05a4a/issue-conditions",

"paymentSummary": "{{LOCATION}}/bonds/ecfe885d-8163-4222-acb1-f7020cf05a4a/payment-summary"

}

pm.test("El response devuelve codigo 200", *function* () {

  pm.response.to.have.status(200);

});

*const* response = pm.response.json();

const self = response[“self”];

pm.test("Es el bono y tipo correcto", () *=>* {

    pm.expect(response["name"]).to.contain("Bono (canje 2020)");

pm.expect(response["debtType"]).to.contain("Título Público");

});

pm.test("Contiene los archivos correspondientes", () *=>* {

    pm.expect(response["coupons "]).to.contain(self + "/coupons");

pm.expect(response["identifiers "]).to.contain(self + "/identifiers ");

pm.expect(response["issueConditions "]).to.contain(self + "/issue-conditions ");

pm.expect(response["paymentSummary "]).to.contain(self + "/payment-summary ");

});

8- **Ejercicio Automatización:** Mediante el archivo html (ejercicioQA-login.txt) y el elemento proporcionado se solicita diseñar casos automatizados, puede utilizar cypress (preferido), playwright o selenium, explique el paso a paso de como fue pensando y generando el test. Se deberá subir a un repositorio público para poder realizar la corrección desde ahí.

