UNIVERSIDAD TECNOLÓGICA NACIONAL

FACULTAD REGIONAL TUCUMAN

CÁTEDRA DE INGENIERÍA DE SOFTWARE

INGENIERÍA EN SISTEMAS DE INFORMACIÓN

Trabajo Práctico N°3

Tema: Estrategias para casos de prueba, automatización de pruebas de aceptación, pruebas unitarias y pruebas de sistema

Integrantes

- Guitian, Milena de los Angeles - 52449
- Toledo, Alvaro Julian - 52721
- Veliz, Hector Matias - 52417
- Vera López, Roció Macarena del Milagro - 52663

Docentes a cargo:

- Ing. Vicente Francisco José
- Ing. Dufour Alexandra

Año de cursado: 2024

Comisión 4K2

Fecha de presentación: 18/11/2024

1. Pruebas de particiones

a) Una aplicación de entrenamiento físico mide el número de pasos que se caminan cada día y proporciona información para animar al usuario a mantenerse en forma. La retroalimentación para las diferentes cantidades de pasos debe ser:

- Hasta 1000 pasos - ¡Lleva una vida sedentaria!
- Más de 1000 pasos, hasta 2000 - ¡Lleva una vida poco activa!
- Más de 2000 pasos, hasta 4000 - ¡Se acerca al objetivo!
- Más de 4000 pasos, hasta 6000 - ¡No está mal!
- Más de 6000 pasos - ¡Así se hace!

¿Cuál de los siguientes conjuntos de entradas de prueba lograría la cobertura de partición de equivalencia más alta?

Selecciones una y explique:

A:	(0,	1000,	2000,	3000,	4000)
B:	(1000,	2001,	4000,	4001,	6000)
C:	(123,	2345,	3456,	4567,	5678)
D:	(666,	999,	2222,	5555,	6666)

0 1000 2000 4000 6000

A)

0 - 1000	2000	3000 - 4000		

B)

1000		2001- 4000	4001 - 6000	

C)

123		2345- 3456	4567 - 5678	

D)

666 - 999		2222	5555	6666

Respuesta: El conjunto de entradas de prueba de la **partición D** alcanzaría la cobertura de partición más alta.

b) Un registrador de radiación diaria para plantas genera una puntuación de radiación solar basada en una combinación del número de horas a la que una planta

está expuesta al sol (menos de 3 horas, de 3 a 6 horas o más de 6 horas) y la intensidad media de la luz solar (muy baja, baja, media, alta).

Dados los siguientes casos de prueba:

	HORAS	INTENSIDAD	PUNTUACION
CP 1	1,5	myy baja	10
CP 2	7	media	60
CP 3	0,5	muy baja	10

¿Cuál es el número mínimo de casos de prueba adicionales que se necesitan para garantizar la cobertura completa de todas las particiones de equivalencia de ENTRADA válidas? Explique.

a) 1

b) 2

c) 3

d) 4

	X<3	3<X<6	X>6
Muy baja	CP 1 – CP 3		
Baja		**CP 4 (4,5 - baja)**	
Media			CP 2
Alta			**CP 5 (6,5 - alta)**

Respuesta: El número mínimo de casos de pruebas adicionales que se necesitan para garantizar la cobertura completa de todas las particiones de equivalencia de Entrada validas 2: CP4 para baja y CP5 en nuestro caso pusimos alta.

2. Pruebas de caminos

Realiza el grafo para el siguiente código. Calcula la complejidad ciclomática de las tres formas:

CC 1 = $Cantidad\ de\ condiciones$ + 1

CC 2 = $Aristas - Nodos$ + 2

CC 3 = Á$reas\ cerradas$ + 1

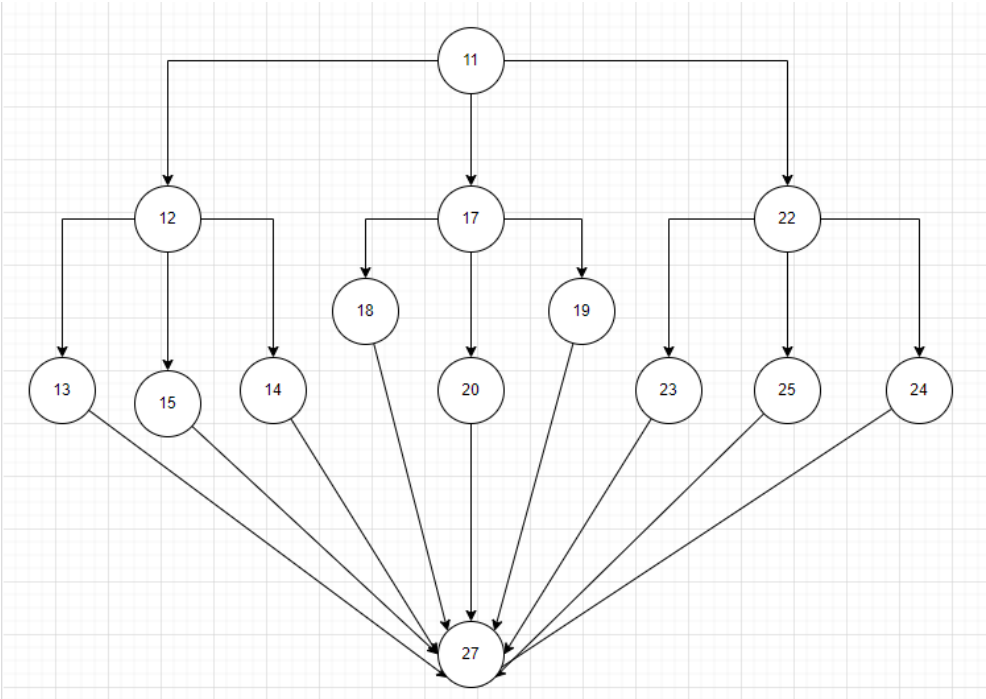
Al mismo tiempo, explica cuáles serían los caminos que se deben recorrer para cubrir todo el programa.

```java
public class PiedraPapelTijera {

    public enum Jugada{ PIEDRA, PAPEL, TIJERA }

        public String jugar(Jugada jugador1, Jugada jugador2){
        final String ganador1 = "Jugador 1";
        final String ganador2 = "Jugador 2";
        final String empate = "Empate";
        final String error = "Error";

            switch (jugador1){
                case PIEDRA -> {
                if (jugador2 == Jugada.PAPEL) return ganador2;
                if (jugador2 == Jugada.TIJERA) return ganador1;
                else return empate;
                }
                case PAPEL -> {
                if (jugador2 == Jugada.TIJERA) return ganador2;
                if (jugador2 == Jugada.PIEDRA) return ganador1;
                else return empate;
                }
                case TIJERA -> {
                if (jugador2 == Jugada.PIEDRA) return ganador2;
                if (jugador2 == Jugada.PAPEL) return ganador1;
                else return empate;
                }
            default -> {return error;}
            }
        }
}
```



CC 1 = $Cantidad\ de\ condiciones$ + 1

CC 1 = 8 + 1 = 9

CC 2 = $Aristas$ − $Nodos$ + 2

CC 2 = 21 − 14 + 2 = 9

CC 3 = Á$reas\ cerradas$ + 1

CC 3 = 8 + 1 = 9

3. **Pruebas de Unidad (unitaias)**

Plantear las pruebas unitarias para la clase **PiedraPapelTijera**. Usar tanto la estrategia de caminos como la estrategia de particiones para cubrir todos los errores.

// **testPiedraContraPapel**: Prueba que cuando jugador 1 elige PIEDRA y jugador 2 elige PAPEL, el resultado es " Ganador Jugador 2".

```
@Test
public void testPiedraContraPapel() {
// Arrange
PiedraPapelTijera juego = new PiedraPapelTijera();
PiedraPapelTijera.Jugada jugador1 = PiedraPapelTijera.Jugada.PIEDRA;
PiedraPapelTijera.Jugada jugador2 = PiedraPapelTijera.Jugada.PAPEL;
// Act
String resultado = juego.jugar(jugador1, jugador2);
// Assert
assertEquals("Jugador 2", resultado); }
```

// testPiedraContraTijera: Prueba que cuando jugador 1 elige PIEDRA y jugador 2 elige TIJERA, el resultado es "Ganador Jugador 1".

```
@Test
public void testPiedraContraTijera() {
// Arrange
PiedraPapelTijera juego = new PiedraPapelTijera();
```

```
PiedraPapelTijera.Jugada jugador1 = PiedraPapelTijera.Jugada.PIEDRA;
PiedraPapelTijera.Jugada jugador2 = PiedraPapelTijera.Jugada.TIJERA;
// Act
String resultado = juego.jugar(jugador1, jugador2);
// Assert
assertEquals("Jugador 1", resultado); }
```

// **testPiedraContraPiedra**: Prueba que cuando ambos jugadores eligen PIEDRA, el resultado es "Empate".

```
@Test
public void testPiedraContraPiedra() {
// Arrange
PiedraPapelTijera juego = new PiedraPapelTijera();
PiedraPapelTijera.Jugada jugador1 = PiedraPapelTijera.Jugada.PIEDRA;
PiedraPapelTijera.Jugada jugador2 = PiedraPapelTijera.Jugada.PIEDRA;
// Act
String resultado = juego.jugar(jugador1, jugador2);
// Assert
assertEquals("Empate", resultado); }
```

4. **Automatización de Pruebas de Aceptación y Pruebas Unitarias**

a) Automatizar, por lo menos, 2 (dos) escenarios en Gherkin realizados para el TP N° 2.

b) Durante el proceso de automatización deberán realizarse, por lo menos, 3 (tres) pruebas unitarias.

5. **Pruebas de Versión (sistema)**

Para el caso de uso Realizar Receta Digital, desarrollar al menos 2 (dos) casos de prueba. Los casos se deben preparar en la plantilla que se adjunta.

Caso de Prueba		
ID: CP1	**Nombre:** Comprobar que el medico puede agregar mínimo 1 o máximo 2 medicamentos a la receta digital	
Descripción: **Como** medico **Quiero** agregar una receta digital **Para** poder asignarle al paciente y tener un registro de la misma		
Prioridad: Alta	**CU / HU:** Realizar Receta Digital	
Modulo / Funcionalidad: Receta Digital		
Diseñado por: Grupo 1		**Fecha: 17/11/2024**
Ejecutado por: Grupo 1		**Fecha: 17/11/2024**

Precondiciones:

- Tener cargada una lista de medicamentos con nombre genérico y comercial
- Contar con una cuenta de medico con sus permisos
- Tener un diagnóstico asociado

Paso	Acción	Resultado Esperado	Paso/ Fallo	Comentarios
1	Buscar medicamento “Optamox Duo” en vista de Receta Digital	Que se muestre una lista con los medicamentos que coindicen con la búsqueda	Paso	Trae opción que no coindicen con las tres primeras letras
2	Seleccionar el medicamento “Optamox Duo”	Que se muestre seleccionada en la vista de Receta Digital	Paso	
3	Buscar medicamento “Riafrianex” en vista de Receta Digital	Que se muestre una lista con los medicamentos que coinciden con la búsqueda	Paso	Trae opción que no coindicen con las tres primeras letras
4	Seleccionar el medicamento “Reafrianex”	Que se muestre seleccionada en la vista de Receta Digital	Paso	
5	Buscar medicamento “Amoxidal” en vista de Receta Digital	Que el buscador se muestre desactivado y que no permita escribir	Paso	

Caso de Prueba		
ID: CP2	**Nombre:** Comprobar que se ingresan todos los datos requeridos nombre, apellido, obra social y numero de afiliado, medicamento y diagnostico	
Descripción: **Como** medico **Quiero** agregar una receta digital **Para** poder enviarla al paciente y tener un registro de las mismas		
Prioridad: Alta	**CU / HU:** Realizar Receta Digital	
Modulo / Funcionalidad: Receta Digital		
Diseñado por: Milena Guitian		**Fecha: 17/11/2024**

Ejecutado por: Rocio Vera Lopez	**Fecha: 17/11/2024**

Precondiciones:
- Tener un diagnóstico asociado
- Tener una cuenta de medico con sus permisos

Paso	**Acción**	**Resultado Esperado**	**Paso/Fallo**	**Comentarios**
1	Ingresar número de DNI		Paso	Obligatorio completar el cuadro de texto
2	Ingresar nombre y apellido		Paso	Obligatorio completar el cuadro de texto
3	Ingresar obra social y numero de afiliado		Paso	Obligatorio completar el cuadro de texto
4	Ingresar medicamento		Paso	Obligatorio completar el cuadro de texto
5	Ingresar diagnostico		Paso	Obligatorio completar el cuadro de texto
5	Confirmar	Mensaje de creación exitosa. Registro almacenado correctamente en la base de datos.		

Caso de Prueba		
ID: CP3	**Nombre:** Comprobar que se pueda enviar la receta digital por WhatsApp	
Descripción: **Como** medico **Quiero** agregar una receta digital **Para** poder enviarla al paciente por WhatsApp y tener un registro de la misma		
Prioridad: Alta	**CU / HU:** Realizar Receta Digital	
Modulo / Funcionalidad: Receta Digital		
Diseñado por: Matías Veliz		**Fecha: 17/11/2024**
Ejecutado por: Alvaro Toledo		**Fecha: 17/11/2024**

Precondiciones:
- Tener cargada una receta digital con los datos requeridos
- Comprobar que el sistema esté conectado con el sistema externo API WhatsApp

Paso	**Acción**	**Resultado Esperado**	**Paso/ Fallo**	**Comentarios**
1	Seleccionar Receta Digital que desee enviar		Paso	

2	Presionar “Enviar por WhatsApp”	Mensaje de envió exitoso. Registro almacenado correctamente en la base de datos.	Paso	