

**PRUEBA PARCIAL N°1**  
**CSY4111**  
**CALIDAD DE SOFTWARE**

<b>NOMBRE:</b>	
<b>SECCIÓN:</b>	<b>FECHA:</b>

<b>DETALLE EVALUACIÓN</b>	<b>UNIDAD DE APRENDIZAJE:</b> Principios Calidad de Software	
	<b>UNIDAD DE COMPETENCIA:</b> Planifica las actividades de ejecución de pruebas a fin de determinar los recursos e iteraciones de acuerdo al requerimiento del cliente y las buenas prácticas de la industria.	
	<b>APRENDIZAJES EVALUADOS:</b>  Implementar el proceso de gestión para el aseguramiento de calidad de software a través de métodos que garanticen el cumplimiento de los requerimientos informáticos que permitan validar y verificar los productos de software a nivel funcional y no funcional.  Diseña casos de prueba de acuerdo a las definiciones recibidas de los requerimientos propios del cliente y tomando en consideración los estándares definidos por la industria.	
<b>Puntaje Total:</b>	<b>20,0</b>	<b>Nota: 7.0</b>
<b>Puntaje:</b>	<b>12,0</b>	<b>Nota: 4.0</b>
<b>Puntaje obtenido:</b>		<b>NOTA:</b>

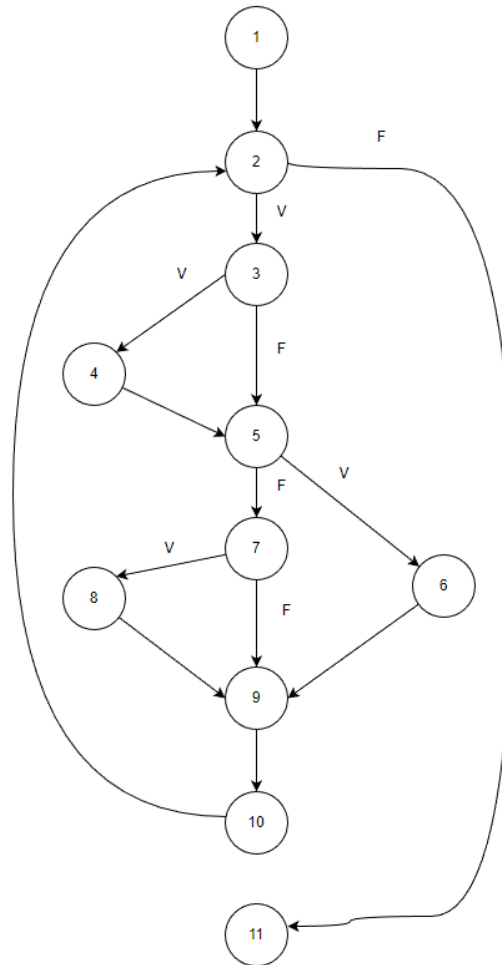
Desarrolle los problemas que se presentan a continuación:

1. Para el grafo que se presenta a continuación:

Calcule su nivel de complejidad ciclomatica, utilizando la fórmula provista

Comente el nivel de complejidad que tiene

Identifique los caminos independientes que posee



---

**2.** Dado el siguiente código:

Presente el grafo que describe a esta pieza de código en un archivo .jpeg o pdf.  
Calcule su complejidad ciclomatica.

```
def actualizar_calidad(self):
    for item in self.items:
        if item.nombre == "Sidra":
            continue
        if item.nombre == "Vino Dulce":
            if item.calidad < 50:
                item.calidad = item.calidad + 1
            if item.vender_en < 1 and item.calidad < 50:
                item.calidad = item.calidad + 1
        elif item.nombre == "whiskey":
            if item.calidad < 50:
                item.calidad = item.calidad + 1
            if item.vender_en < 11 and item.calidad < 50:
                item.calidad = item.calidad + 1
            if item.vender_en < 6 and item.calidad < 50:
                item.calidad = item.calidad + 1
            if item.vender_en < 1:
                item.calidad = item.calidad - item.calidad
        else:
            if item.calidad > 0:
                item.calidad = item.calidad - 1
            if item.vender_en < 1 and item.calidad > 0:
                item.calidad = item.calidad - 1
    item.vender_en = item.vender_en - 1
```

**3.** Para el caso que se describe a continuación, desarrolle y presente la tabla de decisión correspondiente y los casos de prueba que se deducen de ella.

Las siguientes instrucciones fueron tomadas de un formulario de solicitud de examen universitario. Examine y cree una tabla de decisión para representar las reglas de la universidad y, por lo tanto, decidir los casos de prueba que se diseñarán.

Un candidato que se presente al examen debe tener un mínimo de 80% de asistencia.

Ha asistido a 3 pruebas internas con un promedio de 10 o más calificaciones o

Ha asistido a 2 pruebas internas con un promedio de 15 o más calificaciones y

Ha realizado el examen externo y obtuvo más de 35 puntos.

Si se cumplen las condiciones anteriores, el estudiante puede ser considerado como aprobado en esa materia.

4. Dada la especificación y pieza de código que se presenta a continuación, efectúe la inspección mediante usar la lista de verificación que se proporciona en archivo Word (Tabla Inspeccion Código Python) que acompaña a esta prueba.

**Especificación:**

Diseñe un programa que use una iteración para construir una lista de números válidos que contenga solo los números entre 0 y 100 a partir de la lista que se proporciona mas abajo. El programa debe construir esta nueva lista e imprimir por pantalla la suma total de estos números (de la nueva lista) y el valor promedio de los números presentes en esta lista generada.

El programa debe contener comentarios que expliquen lo que hace cada conjunto de instrucciones.

En el programa todas las variables que use deben comenzar con los caracteres var\_ seguido del nombre de la variable.

Las variables de tipo contador deben comenzar con cont\_ seguido del nombre de la variable

Las variables de tipo acumuladoras deben comenzar con sum\_ seguido del nombre de la variable

**Código:**

```
def main():
    numeros = [74, 19, 105, 20, -2, 67, 77, 142, -45, 38]
    total = 0

    numeros_validos = []

    for num in numeros:
        if num > 0 and num < 100:
            numeros_validos.append(num)
            total += total

    promedio = total / len(numeros_validos)

    print('Total de numeros validos: ', total)
    print('Promedio de numeros validos: ', format(promedio, '.4f'))
```

5. Cálculo y Visualización de Métricas. Para la planilla Excel (P1\_Metricas.xlsx) que acompaña a esta prueba, desarrolle los cálculos y gráficos que se solicitan.

## **Envío**

Envíe todo su desarrollo al Blackboard (vía mensaje, en formato RAR) y COMO RESPALDO al correo: f.madridp@profesor.duoc.cl, con asunto CSY\_TE1\_numero de sección. Su envío debe estar dentro de la hora asignada al desarrollo del esfuerzo.

### **Nombre el Archivo a enviar:**

CSY\_TE1\_NroSección\_PrimerLetraNombreIntegrante1Apellido\_PrimerLetraNombreIntegrante2Apellido\_  
PrimerLetraNombreIntegrante3Apellido

Ejemplo: Si el grupo es de la sección 006D y está compuesto por Carlos Cárdenas, Julio Saldivia y Heriberto Valenzuela, el nombre del archivo debe ser:

CSY\_TE1\_006D\_CCardenas\_JSaldivia\_HValenzuela

**ATENCIÓN: NO SE ACEPTARÁ LA ENTREGA DE INFORMES POR MEDIO DE HACER REFERENCIAS A LINKS O DRIVE COMPARTIDO.**