

LISTA DE COTEJO PARA INVESTIGACIÓN 4 UNIDAD

INSTITUTO TECNOLÓGICO SUPERIOR DE SAN ANDRÉS TUXTLA		ASIGNATURA: PROGRAMACIÓN AVANZADA		
NOMBRE DEL DOCENTE: MTI. ROBERTO ESTEBAN GUERRERO HERNANDEZ		FIRMA DEL DOCENTE		
DATOS GENERALES DEL PROCESO DE EVALUACIÓN				
NOMBRE(S) DEL ALUMNO(S): AHIRAM ALBERTO GOMEZ HERNANDEZ		NOMBRE DEL PROYECTO: REUTILIZACION DE CODIGO	FIRMA DEL ALUMNO(S):	
		FECHA: NOVIEMBRE	PERIODO ESCOLAR: SEP 23 - ENE 24	
INSTRUCCIONES				
Revisar las actividades que se solicitan y marque con una "X" en los apartados "SI" cuando la evidencia se cumple; en caso contrario marque "NO". En la columna "OBSERVACIONES" indicaciones que puedan ayudar al alumno a saber cuáles son las condiciones no cumplidas, si fuese necesario.				
VALOR DEL REACTIVO	CARACTERÍSTICA PARA CUMPLIR (REACTIVO)	CUMPLE		OBSERVACIONES
		SI	NO	
3	Presentación El trabajo cumple con los requisitos de: a. Buena presentación	X		
3	b. No presenta faltas de ortografía	X		
3	c. Cuenta con el Formato (Fuente Arial 12 y títulos en negritas)	X		
3	e. Maneja el lenguaje técnico apropiado	X		
5	Introducción: La introducción da una idea clara del contenido del trabajo, motivando al lector a continuar con su lectura y revisión, no copiar y pegar introducción de otro autor, redactadas por usted mismo.	X		
5	Desarrollo: Sigue una metodología y sustenta todos los pasos que se realizaron al aplicar los conocimientos obtenidos, es analítico y bien ordenado.	X		
3	Citas bibliográficas: menciona las citas bibliográficas donde sustenta su comentario personal.	X		
3	Conclusiones: Las conclusiones son claras y acordes con el tema, no copiar y pegar, redactar sus propias conclusiones.	X		
2	Responsabilidad: Entregó el ensayo en la fecha y hora señalada.	X		
30	CALIFICACIÓN	30		

PROGRAMACIÓN

AVANZADA

ING.ROBERTO ESTEBAN GUERRERO
HERNANDEZ

ING.MECATRONICA



311-A

AHIRAM ALBERTO GOMEZ HERNANDEZ

4.3 REUTILIZACION DE CODIGO

La **reutilización de código** se refiere al comportamiento y a las técnicas que garantizan que una parte o la totalidad de un programa de información existente se pueda emplear en la construcción de otro programa. De esta forma se aprovecha el trabajo anterior, se economiza tiempo, y se reduce la redundancia.

La manera más fácil de reutilizar código es copiarlo total o parcialmente desde el programa antiguo al programa en desarrollo. Pero es trabajoso mantener múltiples copias del mismo código, por lo que en general se elimina la redundancia dejando el código reusable en un único lugar, y llamándolo desde los diferentes programas. Este proceso se conoce como abstracción. La abstracción puede verse claramente en las bibliotecas de software, en las que se agrupan varias operaciones comunes a cierto dominio para facilitar el desarrollo de programas nuevos. Hay bibliotecas para convertir información entre diferentes formatos conocidos, acceder a dispositivos de almacenamiento externos, proporcionar una interfaz con otros programas, manipular información de manera conocida (como números, fechas, o cadenas de texto).

Para que el código existente se pueda reutilizar, debe definir alguna forma de comunicación o interfaz. Esto se puede dar por llamadas a una subrutina, a un objeto, o a una clase.

BENEFICIOS DE LA REUTILIZACION DE CODIGO

- **Ahorro de tiempo.** Reutilizar código significa que no hay que escribir todo el código desde cero. Esto ahorra tiempo, ya que se puede reutilizar el mismo bloque de código en varios proyectos sin tener que volver a escribirlo.
- **Menor riesgo.** Reutilizar el mismo bloque de código significa menor riesgo de errores o fallas en los programas, ya que se ha comprobado muchas veces antes y se conocen sus resultados. Esto significa una mayor fiabilidad para los usuarios finales del software.
- **Mejora la calidad.** Los bloques de códigos reutilizados son revisados por expertos antes de ser publicados, lo que garantiza un alto nivel de calidad en los programas finales.
- **Mayor productividad.** Debido a la facilidad con la que se puede reutilizar el mismo bloque de código, los desarrolladores son capaces de terminar proyectos más rápidamente sin comprometer la calidad del producto final entregado al cliente o usuario final del software.
- **Mejora la economía.** Reutilizando el mismo bloque de códigos, hay una reducción en los costes relacionados con el desarrollador y mantenimiento de software. Esta reducción supone un gran beneficio para las empresas y organizaciones involucradas en su producción.

TECNICAS PARA LA REUTILIZACION DE CODIGO

Estas son algunas de las técnicas más comunes para la reutilización de código:

- **Función.** Una función es un conjunto de líneas de código agrupadas juntas para realizar una tarea específica. Las funciones se pueden llamar desde diferentes partes del programa. Esto hace posible a los programadores escribir el mismo código solo una vez.
- **Bibliotecas.** Una biblioteca es un conjunto predefinido de funciones y clases que se pueden utilizar en un proyecto sin escribir el mismo código por separado. Están diseñadas para facilitar el trabajo del programador ya que le permiten acceder fácilmente a su contenido sin necesidad de escribirlos desde cero.
- **Clase.** Una clase es un fragmento de código que se puede reutilizar en diferentes partes del programa con la ayuda de herencia y polimorfismo. Está compuesta por variables y métodos. Lo que permite crear objetos con los mismos atributos, pero comportamientos ligeramente diferenciados entre sí.
- **frameworks.** Los frameworks son librerías o conjuntos predefinidos de bibliotecas diseñados para facilitar la creación rápida y sencilla de proyectos complejos sin necesidad de empezar desde 0. Se basan en muchas convenciones preestablecidas. En consecuencia, no hay necesidad de volver a escribir el mismo código varias veces para lograr resultados similares entre proyectos distintos.

Ventajas:

- Reduce el tiempo necesario para crear una aplicación. Esto, debido a que se reutiliza código existente.
- Facilita el mantenimiento. Esto, debido a que se simplifican los errores y las actualizaciones se realizan en un solo lugar.
- Mejora la calidad del software al reutilizar componentes confiables y probados.
- Disminuye el coste de desarrollo al usar componentes ya existentes.
- Ofrece flexibilidad al permitir que los componentes estén disponibles para su uso en varios sistemas operativos diferentes.
- Mejora la productividad del equipo al simplificar el proceso de codificación.

Desventajas:

- El coste inicial puede ser bastante elevado debido al tiempo requerido para encontrar y adaptar los componentes adecuados.
- La documentación puede no ser adecuada o incompleta, lo que dificulta la comprensión del código fuente por parte del equipo de desarrolladores.
- Los componentes pueden estar obsoletos o no ser compatibles con versiones más recientes del sistema operativo, lo que requiere una actualización constante o nuevo código personalizado para solucionar problemas críticos.

LISTA DE COTEJO PARA REPORTE DE PRÁCTICA

INSTITUTO TECNOLÓGICO SUPERIOR DE SAN ANDRÉS TUXTLA			ASIGNATURA: PROGRAMACION AVANZADA	
NOMBRE DEL DOCENTE: MTI. ROBERTO E. GUERRERO HERNANDEZ			FIRMA DEL DOCENTE	
DATOS GENERALES DEL PROCESO DE EVALUACIÓN				
NOMBRE(S) DEL ALUMNO(S): AHIRAM ALBERTO GOMEZ HERNANDEZ		MATRICULA: 221U0538		FIRMA DEL ALUMNO(S):
PRODUCTO: REPORTE DE PRACTICA	NOMBRE DE LA PRACTICA: REUTILIZACION DE CODIGO	FECHA: NOVIEMBRE - 2023		PERIODO ESCOLAR: SEPTIEMBRE 23 – ENERO 24
INSTRUCCIONES				
<p>Revisar las actividades que se solicitan y marque con una "X" en los apartados "SI" cuando la evidencia se cumple; en caso contrario marque "NO". En la columna "OBSERVACIONES" indicaciones que puedan ayudar al alumno a saber cuáles son las condiciones no cumplidas, si fuese necesario.</p>				
VALOR DEL REACTIVO	CARACTERÍSTICA PARA CUMPLIR (REACTIVO)	CUMPLE		OBSERVACIONES
		SI	NO	
5%	Presentación El trabajo cumple con los requisitos de: a. Buena presentación	X		
4%	b. No tiene faltas de ortografía	X		
4%	c. Mismo Formato (letra arial 12, títulos con negritas)	X		
4%	d. Maneja el lenguaje técnico apropiado	X		
10%	Desarrollo: Sigue una metodología y sustenta todos los pasos que se realizaron en el análisis y desarrollo en la aplicación de los temporizadores dentro de la programación del PLC y aplicando los conocimientos obtenidos, es analítico y bien ordenado.	X		
3%	Responsabilidad: Entregó la práctica en la fecha y hora señalada.	X		
30%	CALIFICACIÓN	30 %		

C:\> Users > MTI ESTEBAN > Downloads > Practica Unidad 4 PA 1.py

```
1 import math
2
3 # Funcion para calcular el area de un cuadrado
4 def calcular_area_cuadrado(lado):
5     return lado * lado
6
7 # Funcion para calcular el area de un circulo
8 def calcular_area_circulo(radio):
9     return math.pi * radio ** 2
10
11 # Ejemplos de uso
12 lado_cuadrado = 5
13 radio_circulo = 3
14
15 area_cuadrado = calcular_area_cuadrado(lado_cuadrado)
16 area_circulo = calcular_area_circulo(radio_circulo)
17
18 print(f"Area del cuadrado: {area_cuadrado}")
19 print(f"Area del circulo: {area_circulo}")
```

HEẢ Á
ÒÝÒÒSÒP VÒÁ
ÔUỒỒU

EXPLORER

NO FOLDER OPENED

You have not yet opened a folder.

Open Folder

Opening a folder will close all currently open editors. To keep them open, add a folder instead.

OUTLINE

TIMELINE

```

CALCULADORA.py X
C:\Users\MTIEST~1\AppData\Local\Temp\Rar$Dla1952.47133\CALCULADORA.py
1  # EN ESTE SEGUNDO CODIGO SE EJECUTAN LAS OPERACIONES
2  from VALORES import obtener_valores
3
4
5  def realizar_operacion(num1, num2, operacion):
6      resultado = 0
7
8      if operacion == "+":
9          resultado = num1 + num2
10         elif operacion == "-":
11             resultado = num1 - num2
12         elif operacion == "*":
13             resultado = num1 * num2
14         elif operacion == "/":
15             if num2 != 0:
16                 resultado = num1 / num2
17             else:
18                 print("Error: No se puede dividir por cero.")
19                 return None
20
21         return resultado
22
23 def mostrar_resultado(resultado):
24     if resultado is not None:
25         print("El resultado es: {}".format(resultado))
26
27 # Obtener valores
28 num1, num2, operacion = obtener_valores()
29
30 # Realizar operacion
31 resultado = realizar_operacion(num1, num2, operacion)
32
33 # Mostrar resultado

```

ΌΥΘΕΙ ΟΡΑ
ΗΕΑ