UNIVERSIDAD DE AQUINO BOLIVIA

FICHA DE IDENTIFICACIÓN DE PROYECTO

Título	PROGRAMACION ORIENTADA A OBJETOS	
Autor/es	Nombres y Apellidos	Código de estudiantes
	Joshua Prado Salvatierra	
Fecha	dd/06/2024	

Carrera	Redes y Telecomunicaciones
Asignatura	Programación II
Grupo	A
Docente	Benjamín Vargas Alvarez
Periodo Académico	Semestre III/2024
Subsede	Santa Cruz

RESUMEN:

Debe tener un máximo de 500 palabras y contener la información necesaria para darle al lector una idea de la pertinencia y calidad del proyecto, éste debe contener una síntesis del tema, el alcance del proyecto, algunos detalles de la planificación del proyecto y resultados esperados.

Palabras clave:

ABSTRACT:

Key words:

Tabla De Contenidos

*********EJEMPLOS Y BASES PARA LA ELABORACIÓN DE TRABAJOS******
********Este documento está configurado para seguir las normas APA*********

Lista De Tablas

Aquí debe generar el índice de tablas y cuadros.

Lista De Figuras

Aquí debe generar el índice de los gráficos, diagramas, mapas, dibujos y fotografías.

Introducción

La programación orientada a objetos se basa en un estilo de programación basado en un concepto de
clases y la creación de objetos. Además de ello en este paradigma de programación el programa se
controla de manera autónoma donde el desarrollador ejecuta sus programas ya sean estos exigentes
o acordes con el usuario

La programación orientada a objetos se basa en un estilo de programación basado en un concepto de
clases y la creación de objetos. Además de ello en este paradigma de programación el programa se
controla de manera autónoma donde el desarrollador ejecuta sus programas ya sean estos exigentes
o acordes con el usuario
La POO está compuesta de elementos básicos como las clases que esta a su vez está incluida en paquetes que trae un programa, los objetos, mensajes, es como expresaríamos la programación en la
vida real cotidiana por ejemplo las características de un vehículo poniendo como atributos sus colores, su marca, etc.

¿Qué es la programación orientada a objetos (POO)?

La programación Orientada a Objetos que se basa en cualquier objeto que nos rodea ya sea un aparato televisor, celular, etc. es un paradigma para poder programar siendo así una forma de poder programar ya que cada individuo programa de forma distintas siendo así la forma más casual o común programar de forma estructurada. Sin embargo, para proyectos enormes usamos la programación Orientada a Objetos donde el sistema requiere de una seria de comportamientos características donde estos objetos ya sean vehículos, casas entre otros.

Por otro lado, los objetos y datos poseen datos que vienen a ser sus atributos y funcionalidad que viene a ser la lógica del programa como sus métodos, siendo así al implementar un paradigma POO podemos trabajar con módulos o clases separados haciendo así más sencillos al momento de programar.
La POO también está compuesta por una plantilla que se denomina clase la cual esta compuesta o abarca una serie de métodos u objetos donde estos son instanciados mediante el new, y de esta forma podemos crear miles de objetos o usuarios.

Al aplicar estos objetos en una aplicación real teniendo como ejemplo a un usuario respecto al login de un cajero bancario, este usuario tendría como características sus nombres, apellidos, documento de identidad, dirección y estos vendrían a ser sus atributos y para poder ejecutar sus retiros estos requieren usar métodos en la plantilla Clase, para ello asignaremos método Retiro dinero, Ingreso de Dinero, Consulta de saldos siendo estas los métodos que se ejecutara en la salida o interfaz al ejecutar nuestro proyecto de la vida Real Como paradigma la POO se basa en 4 pilares esenciales como la Abstracción, El Encapsulamiento donde se protegen los datos para prevenir la manipulación de estos datos de forma privada ya que los objetos se comunican entre ellos, Polimorfismo que se basa en poder darle la misma orden a diversos objetos y que estos respondan a su misma manera y la Herencia donde tenemos las clases Padres e Hijos donde existe una Superclase y esta hereda funcionalidades a otras más pequeñas.

¿Qué es un objeto?

Un objeto en POO se define como toda aquella identidad observada en el mundo real ya sean Personas, Plantas Animales donde uno puede modelarlo en el mundo de la computación, donde estos poseen una identidad un comportamiento y un estado.
Sin embargo, para poder acceder o sacar este objeto este está vinculado fuertemente con una clase.
Además de ello estos objetos pueden ser modelados en diagramas UML, para una mejor comprensión de sus atributos sus métodos. Por ejemplo:

```python
class Auto:
    def __init__(self, marca, modelo):
        self.marca = marca
        self.modelo = modelo

    def info(self):
        return f"Auto {self.marca} {self.modelo}"


# Creación de instancias de la clase Auto
mi_auto = Auto("Toyota", "Corolla")
tu_auto = Auto("Ford", "Fiesta")
su_auto = Auto("Honda", "Civic")

# Mostrar información de los autos
print("Mi auto:", mi_auto.info())
print("Tu auto:", tu_auto.info())
print("Su auto:", su_auto.info())
```



Conceptos fundamentales

La POO es una forma de programar más sencilla ya que mediante de ella podemos reutilizar código y
sobre todo, facilita a la resolución de problemas ya que trabaja con modularidad que quiere decir que
trabaja por secciones o clases separadas esto nos beneficiaria que al ejecutar cada clase podemos ver
los errores de cada una de ellas a detalle preciso. Sin embargo, aparte de ofrecer muchas ventajas también este paradigma tiene algunos inconvenientes los cuales podemos mencionar aquí:

- Su ejecución de la POO es mucho más pausada o tardía
- Los desarrolladores pueden presentar algunas dificultades al momento de aprender este nuevo paradigma ya que es un poco complicado al momento de acostumbrarse
- Requieren de mucha documentación para sus procesos
- Requiere de una correcta modelación UML antes de empezar a programarlo

Origen de la programación orientada a objetos

Los orígenes de la programación orientada a objetos ocurren con el lenguaje de programación orientado a objetos Simula y en este caso programado o desarrollado por Kristen Nygaard y Ole Joham alrededor de los años sesenta en el Norwegian Computer Center en mi trabajo de conferencia en IFIO TC 2 Simulation Language Luego con esto En lenguaje simulado es posible definir los diferentes procesos de una organización de donde parten los conceptos de clases con sus modelos y sus objetos de ahí al nombre de lo que comúnmente conocemos hoy como propiedades de instancia comportamientos.

La idea de origen de los POO nació cuando en este centro de instalación, mientras se trabajaba con barcos, se tenían que acoplar a otro tipo de barcos en diferentes clases de objetos, donde se tenían que indicar sus datos y los métodos que tenían que tener puesto. Es por esto que otros lenguajes también soportan estos modelos como el primer lenguaje usado por una mujer mejor conocido como ADA, el lenguaje informático PASCAL.
Sin embargo, con el pasar del tiempo fueron surgiendo nuevos lenguajes de programación que hoy en día están incluyendo estos paradigmas de programación. Ademes de ello antes de que surja el lenguaje Java estuvo por detrás el lenguaje de programación Eiffel que fue reemplazado por el lenguaje Java que está orientado a objetos.

Características de la programación orientada a objetos (POO)

Una de las características de la programación orientada a objetos es que está compuesta por una clase y esta clase a su vez se compone de métodos, características del objeto.
Ademes de ello cuenta con principios fundamentales que son:

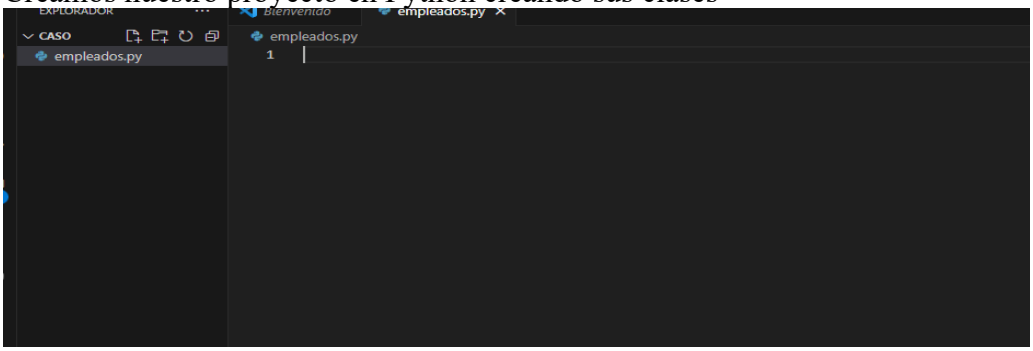
- **La Abstracción**: Es una propiedad que permite los aspectos de un objeto más relevantes como por ejemplo para un niño tendríamos
  Sus datos:
  - Nombre
  - Edad
  - Peso
  - Dirección

  Entre sus métodos tendríamos:
  - Dormir()
  - Jugar()
  - Aprender()
- **El Encapsulamiento:** Esta característica permite incluir todos los elementos que la componen sin que el usuario necesite conocer su estructura
  - Herencia: Útil para compartir elementos y atributos entre clases: Ejemplo Persona-Estudiante-Profesor
  - Polimorfismo: Variable de un objeto puede tomar diversas formas Persona: Estudiante adquiere sus características

Caso de aplicación practico (Python)

Caso: estamos desarrollando un sistema de gestión de empleados para una empresa. Este sistema nos permite manejar la información básica de cada empleado, como su nombre, salario mensual y departamento al que pertenece. Además, proporciona funcionalidades para calcular el salario anual de los empleados y actualizar sus salarios cuando sea necesario.

Creamos nuestro proyecto en Python creando sus clases



Programamos la clase empleado

Para los datos quedaría así

```
def __init__(self, nombre, salario, departamento):
    self.nombre = nombre
    self.salario = salario
    self.departamento = departamento
```

Para calcular el salario

```
def calcular_salario_anual(self):
    return self.salario * 12
```

El código completo quedaría así

```
class Empleado:
    def __init__(self, nombre, salario, departamento):
        self.nombre = nombre
        self.salario = salario
        self.departamento = departamento

    def calcular_salario_anual(self):
        return self.salario * 12

    def actualizar_salario(self, aumento):
        self.salario += aumento

    def __str__(self):
        return f"Empleado: {self.nombre}, Salario: {self.salario}, Departamento: {self.departamento}"


if __name__ == "__main__":
    empleado1 = Empleado("Juan Pérez", 2500, "Ventas")
    empleado2 = Empleado("María López", 3000, "Marketing")

    # Mostrar información de los empleados
    print(empleado1)
    print(empleado2)

    # Calcular salario anual
    print(f"Salario anual de {empleado1.nombre}: {empleado1.calcular_salario_anual()}")
    print(f"Salario anual de {empleado2.nombre}: {empleado2.calcular_salario_anual()}")

    # Actualizar salario de empleado1
    empleado1.actualizar_salario(500)

    # Mostrar información actualizada
    print(f"Nuevo salario de {empleado1.nombre}: {empleado1.salario}")
```

Conclusiones

- El paradigma de la POO más usado por desarrolladores para desarrollar programas exigentes.
- Podemos analizar lo dicho anteriormente que los lenguajes de programación orientados a objetos tratan los programas como una colección de objetos que se ayudan mutuamente a realizar acciones, entienden las entidades que contienen datos como objetos y permiten que los programas sean más fáciles de escribir, mantener y reutilizar.
- Otra conclusión a mencionar es que con el gran avance de este paradigma en la actualidad desarrolla la productividad del desarrollador y permite que los programas sean robustos basándonos en un estándar de calidad como el ISO 25000 y además de ofrecer una robustez, el programa tiene que ser amigable.

Bibliografía

- Morero F. (2000). Introducción a la OOP.Grupo EIDOS. Recuperado de https://kataix.umag.cl/~ruribe/Utilidades/Introduccion%20a%20la%20Programacion%20Orientada%20a%20Objetos.pdf
- Barraza O. et. al (2006). Introducción a la programación orientada a objetos. Recuperado el 9 de junio de 2022, de https://travezurasdeltraviezo.wordpress.com/wp-content/uploads/2014/02/introduccion-a-la-programacion-orientada-a-objetos.pdf
- (S/f-e). Upm.es. Recuperado el 9 de junio de 2022, de https://www.etsisi.upm.es/sites/default/files/curso_2013_14/MASTER/MIW.JEE.POOJ.pdf
- Pedro G. et. al (2016). Recuperado el 9 de junio de 2022, de http://www.cua.uam.mx/pdfs/revistas_electronicas/libros-electronicos/2016/2intro-poo/programacion_web.pdf
- Programación, L. (s/f). Tema 11: Programación orientada a objetos Índice. Rua.ua.es. Recuperado el 9 de junio de 2022, de https://rua.ua.es/dspace/bitstream/10045/4042/1/tema11.pdf
- Wikipedia contributors. (s/f). Programación orientada a objetos. Wikipedia, The Free Encyclopedia. https://es.wikipedia.org/w/index.php?title=Programaci%C3%B3n_orientada_a_objetos&oldid=143668761
- Perfil, V. T. mi. (s/f). HISTORIA DE LOS LENGUAJES DE PROGRAMACION ORIENTADA A OBJETOS. Blogspot.com. Recuperado el 9 de junio de 2022, de http://sis324loo.blogspot.com/2008/09/historia-de-los-lenguajes-de_29.html
- Gavarró Rodríguez, A. (s/f). Introducción a la programación orientada a objetos. Gvsig.org. Recuperado el 9 de junio de 2022, de http://downloads.gvsig.org/download/documents/learning/collaborations/ce_1104_01/Programacion_personalizacion_SIG_1.pdf
- Ventajas y desventajas de la programación orientada a objetos – Acervo Lima. (s/f). Acervolima.com. Recuperado el 9 de junio de 2022, de https://barcelonageeks.com/ventajas-y-desventajas-de-la-programacion-orientada-a-objetos/