LICENCIATURA EN

CIENCIA DE DATOS

PROGRAMACIÓN AVANZADA (189)

Lic. Felipe Morales

Clase N. 2. Unidad I. Clases, propiedades, métodos. Ejemplos y Revisión Práctica Clase anterior.

TEMARIO

- T0: **Revisión** de Clase anterior
  - Introducción a Git
  - Análisis de resoluciones de estudiantes
  - Conclusiones
- T1: Resolución de Práctica Clase N. 2
- T2: Definición de ejemplo
- T3: Codificación en Python 3
- **Temas relacionados**
- **Practica**
- **Links**

T0: Introducción al **Paradigma de Programación Orientada a Objetos**

¿Qué es un paradigma?

"Teoría o conjunto de teorías cuyo núcleo central se acepta sin cuestionar y que suministra la base y modelo para resolver problemas y avanzar en el conocimiento"

¿Qué es un paradigma de programación?

Una posible definición es *"Un paradigma de programación es una manera o estilo de programación de software. Existen diferentes formas de diseñar un lenguaje de programación y varios modos de trabajar para obtener los resultados que necesitan los programadores. Se trata de un conjunto de métodos sistemáticos aplicables en todos los niveles del diseño de programas para resolver problemas computacionales."*

¿Y orientada a Objetos?

Podemos decir que es algo así como "Algo que está focalizado o es representado por OBJETOS -elementos físicos o abstractos que existen en la vida real o imaginaria-"

T1: Introducción al **Paradigma de Programación Orientada a Objetos (Continuación)**

En forma simplificada es:

En este modelo de programación se construyen modelos de objetos que representan elementos (objetos) del problema a resolver, que tienen características y funciones.

Y sus características son:

- Permite separar los componentes de un programa
- Simplifica la creación, depuración y posteriores mejoras del programa
- Ayuda a disminuir errores
- Promueve la reutilización del código

T1: Introducción al **Paradigma de Programación Orientada a Objetos (Continuación)**

¿Existen otros paradigmas de programación?:

SI!

¿Cuáles?

- **Imperativo o Estructura o Procedural**
- Declarativo
- Orientado a Objetos
- Reactivo
- Funcional
- Scripting
- Otros

T0: Pilares del **Paradigma de Programación Orientada a Objetos**

Los **pilares** son los conceptos fundamentales sobre los cuales se sustenta el paradigma.

¿Cuáles?

- Encapsulamiento
- Abstracción
- Herencia
- Polimorfismo

Otros conceptos asociados:

- Cohesión
- Acoplamiento

T3: Pilares del **Paradigma de Programación Orientada a Objetos**

Fuente: ed.team/cursos/poo

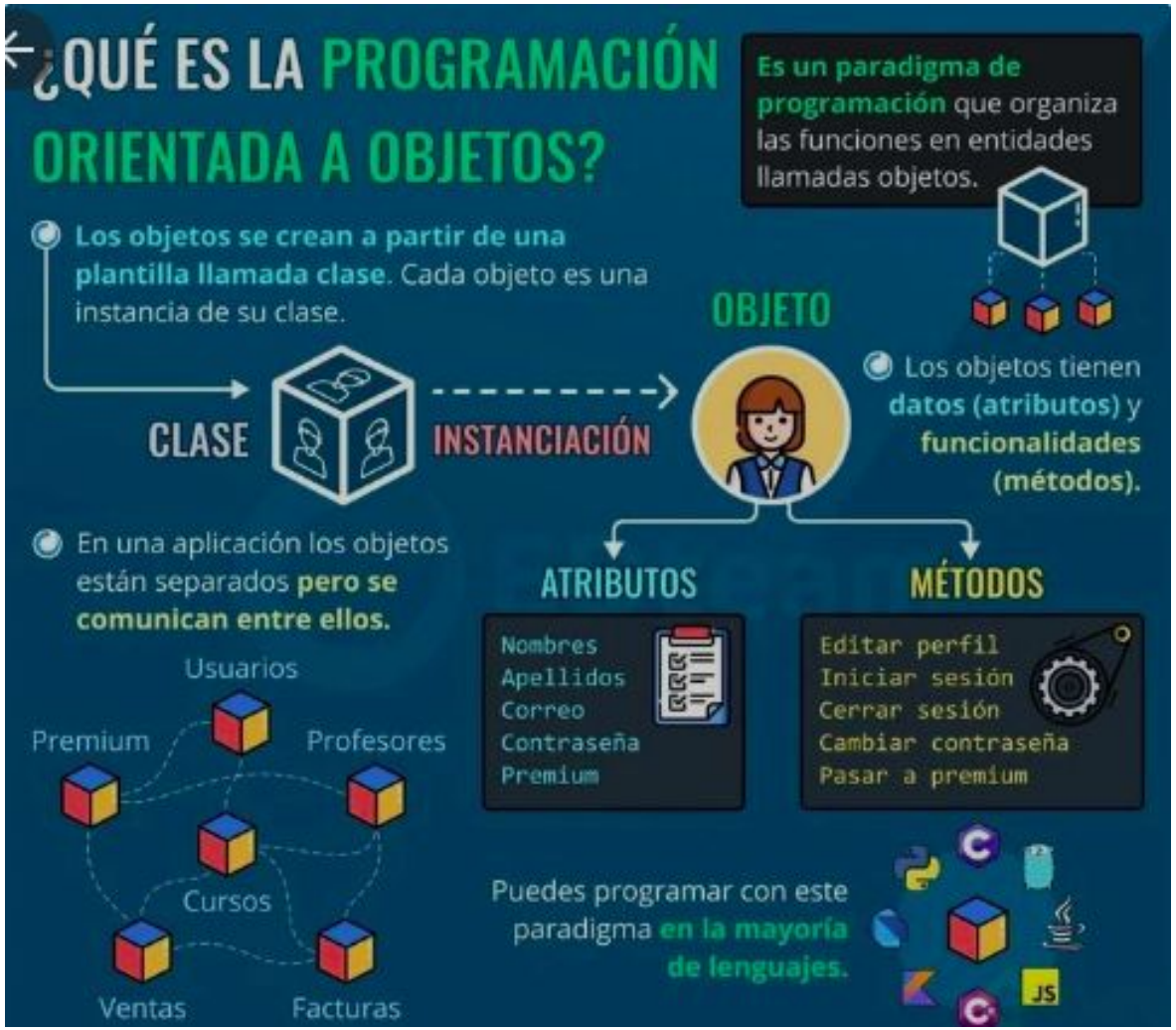
T4: Componentes del **Paradigma de Programación Orientada a Objetos**

Los **componentes** están dados por los siguientes elementos constitutivos de este paradigma:

¿Cuáles?

- Objetos
- Clase
- Atributos
- Métodos

T4: Componentes del **Paradigma de Programación Orientada a Objetos**



Fuente: platzi.com

Práctica 2 - Unidad I

1) **Acceda al repositorio privado en GitHub:**
   - Registrarse en caso de con contar con una cuenta propia
   - Link https://github.com/felipemoralesquerol/UNAB.2024.Programacion-Avanzada
   - Adecuar su IDE favorito para ejecutar los ejemplos

2) **Sobre el repositorio privado en GitHub:**
   - Busqué el texto TODO en el repositorio.
   - ¿Cómo resolvería los interrogantes que plantea?
   - Seleccionar 5 TODOs y resolverlos (en forma individual o grupal)

3) **(Opcional) Seleccionar 2 (dos) de los pilares (o pilares asociados) y arme un breve resumen con:**
   - Breve explicación con sus palabras
   - Programar un ejemplo en Python y explicar cómo aplica este concepto

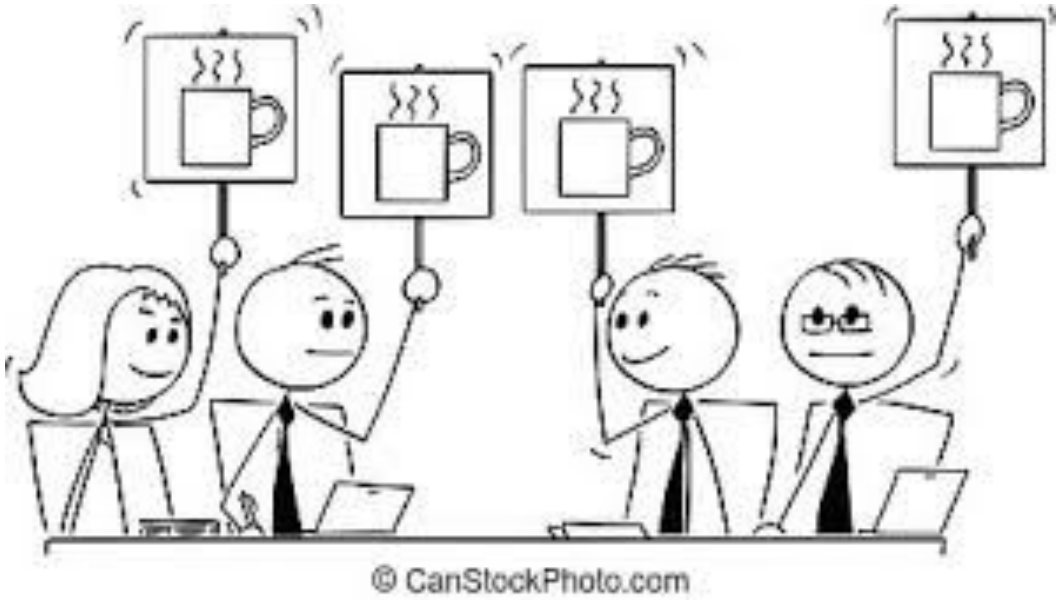
T1: Resolución de Práctica 2 - Unidad I

1) **Acceda al repositorio privado en GitHub:**
   - Registrarse en caso de con contar con una cuenta propia
   - Link https://github.com/felipemoralesquerol/UNAB.2023.Programacion-Avanzada
   - Adecuar su IDE favorito para ejecutar los ejemplos

2) **Sobre el repositorio privado en GitHub:**
   - Busqué el texto TODO en el repositorio.
   - ¿Cómo resolvería los interrogantes que plantea?
   - Seleccionar 5 TODOs y resolverlos (en forma individual o grupal)

3) **(Opcional) Seleccionar 2 (dos) de los pilares (o pilares asociados) y arme un breve resumen con:**
   - Breve explicación con sus palabras
   - Programar un ejemplo en Python y explicar cómo aplica este concepto

BREAK



¿CHALLENGE?

T2: Definir un ejemplo para trabajar en forma colectiva

T3: Codificación en Python 3

Consideraciones:

En base a un ejemplo que plantee el uso de dos clases con la relación “es un” definir:

- Nombre de cada clase
- Propiedades (al menos 2 por cada clase)
- Métodos (al menos 2 por cada clase)
- Instanciación de Objetos (al menos 2 objetos por cada clase)

Cómo estima que cumple con los **pilares de la POO**.

- Encapsulamiento
- Abstracción
- Herencia
- Polimorfismo

Práctica 2 - Unidad I

1) **Crear un repositorio privado en GitHub con el ejercicio anterior para la clase 2 :**
   - Enviar invitación a docente a su cuenta **felipemoralesquerol y a un compañero de cursada**
   - Solicitar a su compañero de cursado invitado a que realice un cambio (por ejemplo agregar una nueva propiedad o método) en su repositorio
   - Posteriormente su compañero deberá hacer commit y push a su repositorio (rama master o main)

2) **Elaborar conclusiones incluyendo opiniones sobre:**
   - Grado de dificultad del ejercicio (escala de 1 a 5) . Justifique.
   - ¿Qué tema le resultó de mayor dificultad?
   - ¿Cómo resolvería esta ejercitación si tuviera de repetirla? ¿Qué cambiaría?

3) **Investigar:**
   - Atributos públicos vs atributos privados
   - ¿Aplica este mismo criterio para los métodos?
   - Utilizando la función dir aplicada a un objeto: ¿Qué devuelve? ¿Puede llamar a un atributo o método privado?
   - ¿Cómo adaptaría el código fuente implementado para atender a este nuevo criterio? Impacte el cambio en su repositorio.

Links interesantes:

Aprenda a pensar como un programado con Python

https://argentinaenpython.com/quiero-aprender-python/aprenda-a-pensar-como-un-programador-con-python.pdf

Documentación oficial:

https://www.python.org/doc/

Curso de Python desde 0 (pildorasinformaticas)

https://www.youtube.com/playlist?list=PLU8oAlHdN5BlvPxziopYZRd55pdqFwkeS

Otros links:

https://ellibrodepython.com/polimorfismo-en-programacion

https://parzibyte.me/blog/2019/06/30/clases-constructores-python-poo/

¡Muchas gracias por tu atención!