



Programador (Orientado a Objetos)

Propósito del Taller (En parejas)

Identificar las características básicas del paradigma orientado a objetos y emplear la sintaxis de Java para crear clases siguiendo las convenciones y buenas prácticas.

Taller:

- I. Relaciona los conceptos con su definición, escribiendo la letra correspondiente en el paréntesis.

a. Identidad	(d) Es una metodología que se basa en la interacción de entidades para organizar y estructurar la información.
b. Comportamiento	(f) Es una representación informática de una entidad real o imaginaria.
c. Herencia, Encapsulamiento y Abstracción.	(e) Almacenan las características del objeto en un determinado momento.
d. Programación Orientada a Objetos	(a) Esta dimensión permite que los objetos sean irrepetibles a pesar de que existan dos o más con las mismas características.
e. Atributos	(b) Está definido por los métodos del objeto y permite su interacción con el programa principal y otros objetos.
f. Objeto	(c) Son características de los objetos que extienden sus propiedades.



II. Observa el código de Java y contesta las preguntas.

```
class Mamifero {
    // Atributos
    private String tipo;
    private boolean cola;
    private boolean garras;

    // Constructor
    public Mamifero(String tipo, boolean cola, boolean
garras) {
        this.tipo = tipo;
        this.colas = cola;
        this.garras = garras;
        nacer(); // Llamamos al método nacer en el
constructor
    }

    // Método nacer
    public void nacer() {
        System.out.println(tipo + ": ha nacido");
    }

    // Método comer
    public void comer() {
        System.out.println(tipo + ": ha comido");
    }

    // Método rugir
    public void rugir() {
        System.out.println(tipo + ": ha rugido");
    }

    // Main para probar la clase Mamifero
    public static void main(String[] args) {
        Mamifero perro = new Mamifero("perro", true, true);
        perro.comer();

        Mamifero ballena = new Mamifero("ballena", true,
false);
        ballena.comer();
    }
}
```

a) ¿Qué nombre tiene la clase?

Mamifero

B) ¿Qué métodos están definidos?

Mamifero() // lo agrego porque es un método llamado constructor
nacer()
comer()
rugir()
main() // anexo el main ya que está dentro de la clase

c) ¿Qué variables de instancia están definidas?

perro
ballena

d) ¿Cuántos objetos se han instanciado?

2: perro y ballena

e) ¿Qué método(s) se ha(n) llamado incluyendo el constructor?

Mamifero() // para los dos objetos
comer() // se usa para los dos objetos instanciados

f) ¿Cuántos atributos están declarados?

3: tipo, cola, garras

g) ¿Cuántos objetos se han instanciado?

2: perro, ballena

III. Crear el paso a paso de cómo crear un repositorio en GitHub y subir el código de los ejercicios.

Se infiere que ya se tiene una cuenta de Github para realizar este proceso.

Creación de repositorio en Github:

1._ Iniciar sesión.

2._ Estando en el Dashboard busca la opción 'create new (esta tiene un + como logo)' presiona click izquierdo allí.

3._ Presiona click izquierdo en create new repository.

- 4._ Le das un nombre al repositorio usando la convención de nombres kebab-case.
- 5._ Seguidamente le das una descripción opcional del repositorio.
- 6._ Eliges si quieres que tu repositorio sea publico o privado.
- 7._ Eliges si quieres que tu repositorio tenga un README file.
- 8._ Eliges si quieres una licencia.
- 9._ Al finalizar le das a create repository, es un botón en color verde.
- 10._ ya tienes un repositorio creado satisfactoriamente.

Se infiere que ya se tiene instalado y configurado git en tu equipo local, es decir ya se tiene sincronizada la llave (ssh) privada con la publica en la cuenta de github, se tiene configurado el nombre de usuario y correo en git config, etc.

Subir archivos .java a Github.com (repositorio remoto):

- 1._ Mediante consola te ubicas en el directorio donde se encuentran los archivos .java
- 2._ Estando allí en la consola y directorio, das la instrucción **"git init ."**. Esta instrucción le dirá a git que inicialice el repositorio local y le da seguimiento a todos los archivos en la ubicación que estás.
- 3._ Después, ingresas la instrucción **"git add ."**. Esta instrucción lo que hace es que agregas todos los archivos a un area muy conocida como staging esto para preparar los archivos para el commit.
- 4._ Luego ingresas la instrucción **"git commit -m "mensaje aqui" "**. Esta instrucción lo que hace es confirmar los archivos que estan en el staging y los guarda en el historial de git (es como si guardara una version de tus archivos confirmados).
- 5._ Ahora es tiempo de sincronizar el repositorio remoto creado en Github.com con el repositorio local en el cual hemos estado trabajando con nuestros archivos, para sincronizar lo dicho anteriormente hacemos en consola **"git remote add origin url-ssh"**. url-ssh es la url que extraemos de nuestro repositorio remoto, tambien hay forma de hacerlo con url-https pero esto ya no se usa el dia de hoy ya que fue removido en 2021.
- 6._ Estando en este punto lo unico que nos hace falta es enviar los archivos del repositorio local al repositorio remoto y esto lo hacemos con la instrucción **"git push origin main"**.
- 7._ Nos dirigimos a Github.com, iniciamos sesión y ya podremos ver nuestros archivos subidos en el repositorio creado.