

PROGRAMACIÓN ORIENTADA A OBJETOS

Profesor: Alexei Ochoa




Grupo: 6 POO

Integrantes: Juan Orozco - Camilo Cuello - David Solís

TALLER 1

SOLUCIÓN:

1)

	Tipo	Paradigma	Características	Sintaxis Hello world
	Interpretado	Orientado a objetos, imperativo, funcional, reflectivo, procedural	Multi-Plataforma, tipado dinámico, lenguaje indentado	<code>print("Hello world")</code>
	Compilado e interpretado	Orientado a objetos (Class-based), imperativo, estructurado, genérico	Multi-plataforma (Máquina virtual), estático,	<code>System.out.println("Hello world");</code>
	Compilado	Orientado a objetos, funcional, procedural, genérico	Permite la manipulación de memoria, fuertemente tipado	<code>cout<<"Hello world";</code>

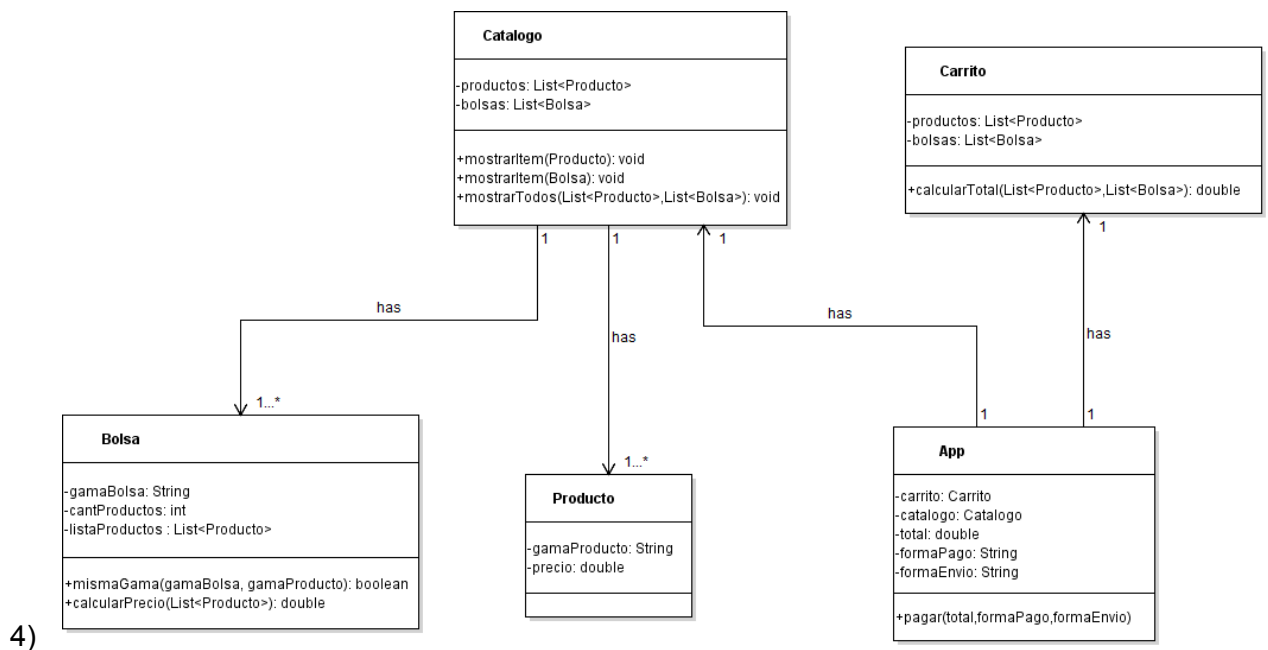
2) La programación orientada a objetos se puede definir como un paradigma de programación, donde se organiza el código fuente en clases. Al programar orientado a objetos se debe pensar la forma de resolver los problemas de una manera distinta a la común. Ahora tendremos que redactar nuestros propios programas en términos de clases, objetos, características, procedimientos, entre otras. Desde este punto se debe pensar en términos de objetos, es decir, de la manera que vemos las cosas en la vida real, en otras palabras materializamos los elementos del código organizándose esquemáticamente. La definición de clase y las propiedades de la misma se explicarán más adelante.

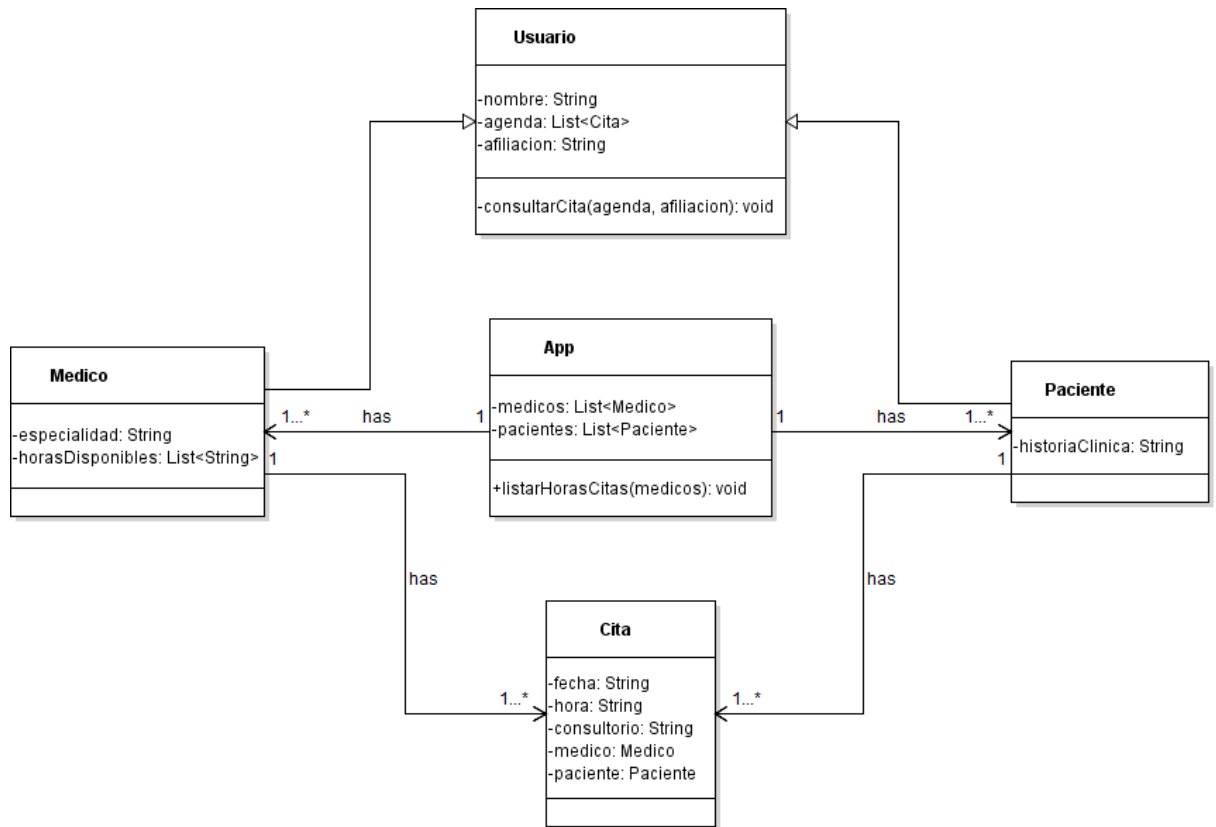
3)

- **Clase:** Una clase puede entenderse como un plano que sirve para crear objetos individuales, y se le deben asignar atributos especiales, así como unos lineamientos de comportamiento [1]. **Ejemplo:** Una clase Bicicleta.
- **Instancia:** Corresponde a un objeto particular que pertenece a una clase [2]. **Ejemplo:** Una bicicleta roja con frenos deportivos, de tamaño mediano.
- **Abstracción:** Corresponde al proceso de capturar las características esenciales y el comportamiento de un objeto [3]. **Ejemplo:** Una bicicleta tiene un color, unos frenos de determinada calidad, un tamaño, y se mueve hacia adelante.
- **Encapsulamiento:** Corresponde al proceso de unir los atributos y comportamientos de una clase, en una sola entidad, lo que a su vez se utiliza para ocultar del exterior de la clase el estado de un objeto que pertenezca a la misma [4]. **Ejemplo:** Una bicicleta tiene un color, unos frenos de

determinada calidad, un tamaño, y se mueve hacia adelante. No obstante, el funcionamiento interno del sistema de frenado, así como su país de origen, es información que le interesa solamente al fabricante o a un mecánico de bicicletas.

- **Herencia:** Corresponde a un mecanismo por el que un objeto adquiere todas las propiedades y comportamientos de un objeto padre [5]. **Ejemplo:** Una bicicleta de montaña tendrá las mismas características y comportamientos esenciales de una bicicleta común, con la diferencia de que la bicicleta de montaña tendrá algunas propiedades y comportamientos adicionales que no están presentes en una bicicleta convencional.
- **Polimorfismo:** Es un concepto a través del cual es posible llevar a cabo una misma acción de maneras diferentes, según la clase en donde quiera ejecutarse la acción [6]. **Ejemplo:** El habla es una habilidad presente tanto en un hispanohablante como en un anglohablante, no obstante, al comunicarse de manera oral utilizan un idioma y un lenguaje distintos.
- **Multiplicidad:** Corresponde al número de objetos que participan en una asociación [7]. **Ejemplo:** En una universidad, puede plantearse una asociación entre profesor, estudiante y monitor de la asignatura, en donde cada individuo está relacionado con los dos restantes. Por ello, la multiplicidad de la asociación será tres sin importar qué papel se escoja entre profesor, estudiante y monitor.





5)

REFERENCIAS

- [1] K. Moore, E. Njeru, y M. Pocevlclus, «Classes (OOP) | Brilliant Math & Science Wiki». <https://brilliant.org/wiki/classes-oop/>.
- [2] «Creating Objects (The Java™ Tutorials > Learning the Java Language > Classes and Objects)». <https://docs.oracle.com/javase/tutorial/java/javaOO/objectcreation.html>.
- [3] «Fundamentos de programación/La Abstracción y el Encapsulamiento - Wikilibros». https://es.wikibooks.org/wiki/Fundamentos_de_programaci%C3%B3n/La_Abstracci%C3%B3n_y_el_Encapsulamiento.
- [4] T. Janssen, «OOP Concept for Beginners: What is Encapsulation – Stackify». <https://stackify.com/oop-concept-for-beginners-what-is-encapsulation/>.
- [5] «Inheritance in Java - Javatpoint». <https://www.javatpoint.com/inheritance-in-java>.
- [6] «Polymorphism in Java - javatpoint». <https://www.javatpoint.com/runtime-polymorphism-in-java>.
- [7] G. Greeff y R. Ghoshal, Eds., «5 - Business process and system modeling tools and packages», en *Practical E-Manufacturing and Supply Chain Management*, Oxford: Newnes, 2004, pp. 112-145.
- [8] Alvarez, M., 2019. Qué es la programación orientada a objetos. [online] Desarrolloweb.com. Available at: <<https://desarrolloweb.com/articulos/499.php>> [Accessed 15 April 2021].