**LA PROGRAMACIÓN ORIENTADA A OBJETOS TIENE CUATRO CARACTERÍSTICAS PRINCIPALES:**

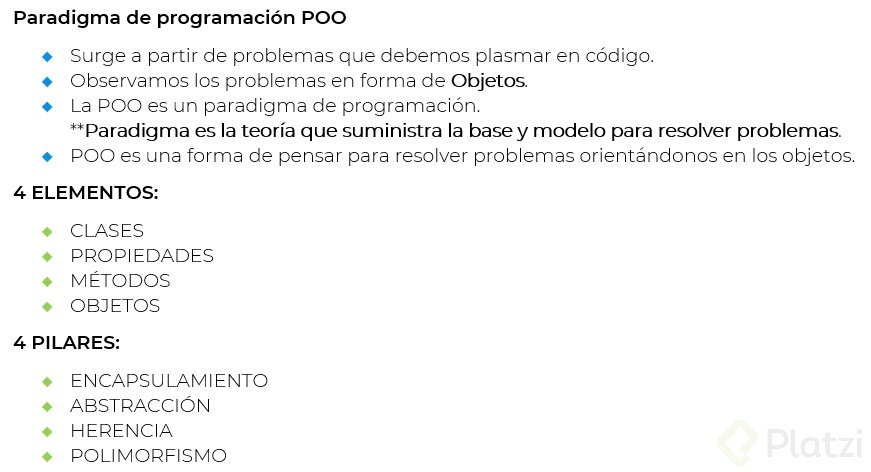
* **ENCAPSULAMIENTO.** Quiere decir que oculta datos mediante código.
* **ABSTRACCIÓN.** Es como se pueden representar los objetos en modo de código.
* **HERENCIA.** Es donde una clase nueva se crea a partir de una clase existente.
* **POLIMORFISMO.** Se refiere a la propiedad por la que es posible enviar mensajes sintácticamente iguales a objetos de tipos distintos.

**¿QUÉ RESUELVE LA PROGRAMACIÓN ORIENTACIÓN A OBJETOS?**

- Principalmente esos problemas y huecos que nos deja la programación estructurada tales como:

1. Código muy largo:
2. Si algo falla todo se rompe
3. Código Spaguetti: Muchas sentencias de control anidadas y pérdida de control sobre el código.

- Simplifica la programación, la hace más práctica, divertida y amigable.



**ALGUNOS DE LOS LENGUAJES DE PROGRAMACIÓN ORIENTADOS A OBJETOS SON:**

* **JAVA:**

– Orientado a Objetos naturalmente

– Es muy útilizado en Android

– Y es usado del lado del servidor o Server Side

* **PHP**

– Lenguaje interpretado

– Pensado para la Web

* **PYTHON**

– Diseñado para ser fácil de usar

– Múltiples usos: Web, Server Side, Análisis de Datos, Machine Learning, etc

* **JAVASCRIPT**

– Lenguaje interpretado

– Orientado a Objetos pero basado en prototipos

– Pensado para la Web

* **C#**
* **RUBY**
* **KOTLIN**

****

**OBJETO:** Son aquellos que tienen propiedades y comportamientos. Un objeto puede ser algo físico o conceptual.

* **UNA PROPIEDAD:** También llamado atributo, son las características o las propiedades que describen a un objeto. Estas propiedades siempre deben ser de forma sustantiva. ej: color, tamaño, etc.
* **UN \*\*COMPORTAMIENTO \*\*O MÉTODO:** Son las acciones que puede realizar un objeto, es decir lo que puede hacer dicho objeto.
* Una **CLASE** Es el modelo sobre el cual nuestros objetos se construyen. A este análisis se le conoce como Abstracción, simplemente consiste en generar un molde en base a esas propiedades y métodos de los objetos, abstraemos todos esos datos para generar dicho molde.

Es decir si tenemos un objeto llamado perro y este tiene sus atributos que lo describen generalmente y a su vez tiene métodos donde se define las acciones que pueda hacer ese perrito. Una clase me permite generar mas objetos (mas perros) con mismos atributos y métodos pero con resultados diferentes.

* **MODULAR:** Dividir un sitema y así crear módulos independientes, lo que permite evitar un colapso masivo en nuestro código y mejorar la legibilidad.

**La modularidad nos permite:**

1. Reutilizar código
2. Evitar Colapsos
3. Mantenible
4. Legibilidad
5. Resolución Rápida de Problemas

**Clases en UML y su sintaxis en código**

Recuerda que el proceso es:

• Identificar el problema, y objetos

• Definir las clases

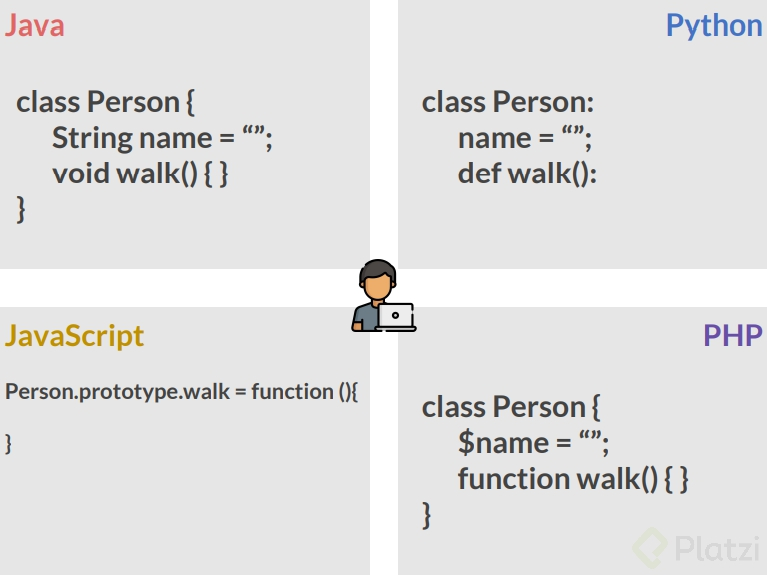
• Plasmarlas en un diagrama

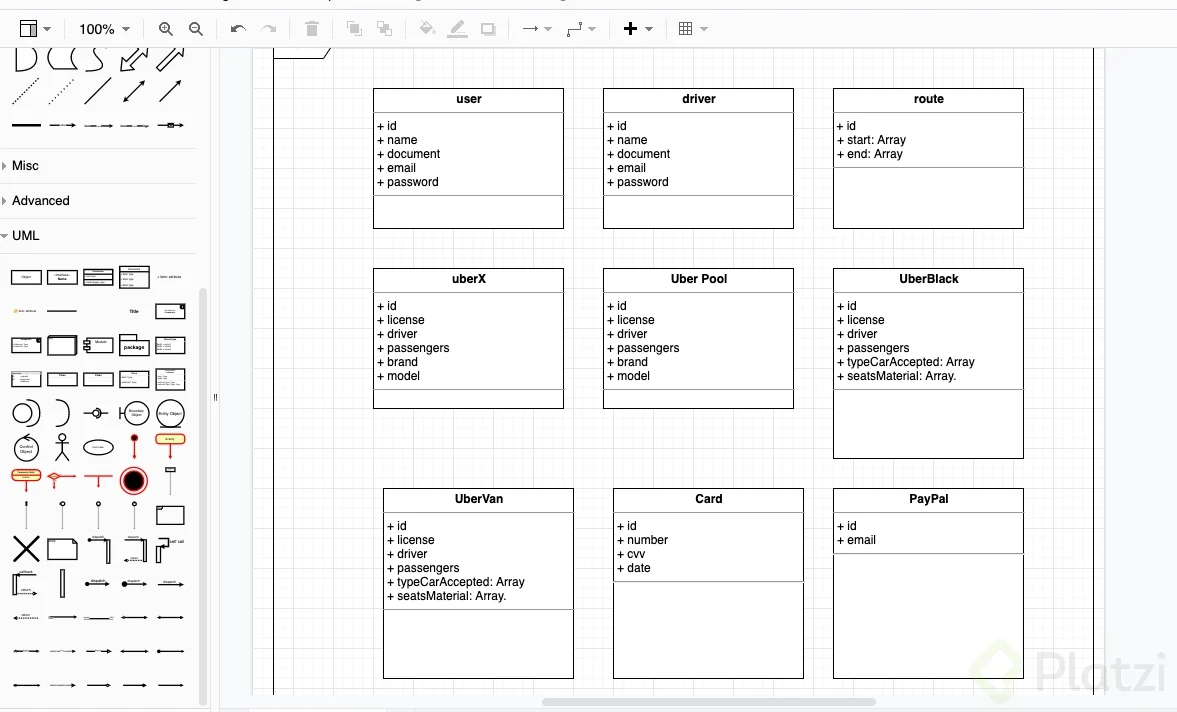
**Se lo puede plasmar en UML :**

• Identidad, que será el nombre de la clase.

• Estado, que serán los atributos de la clase.

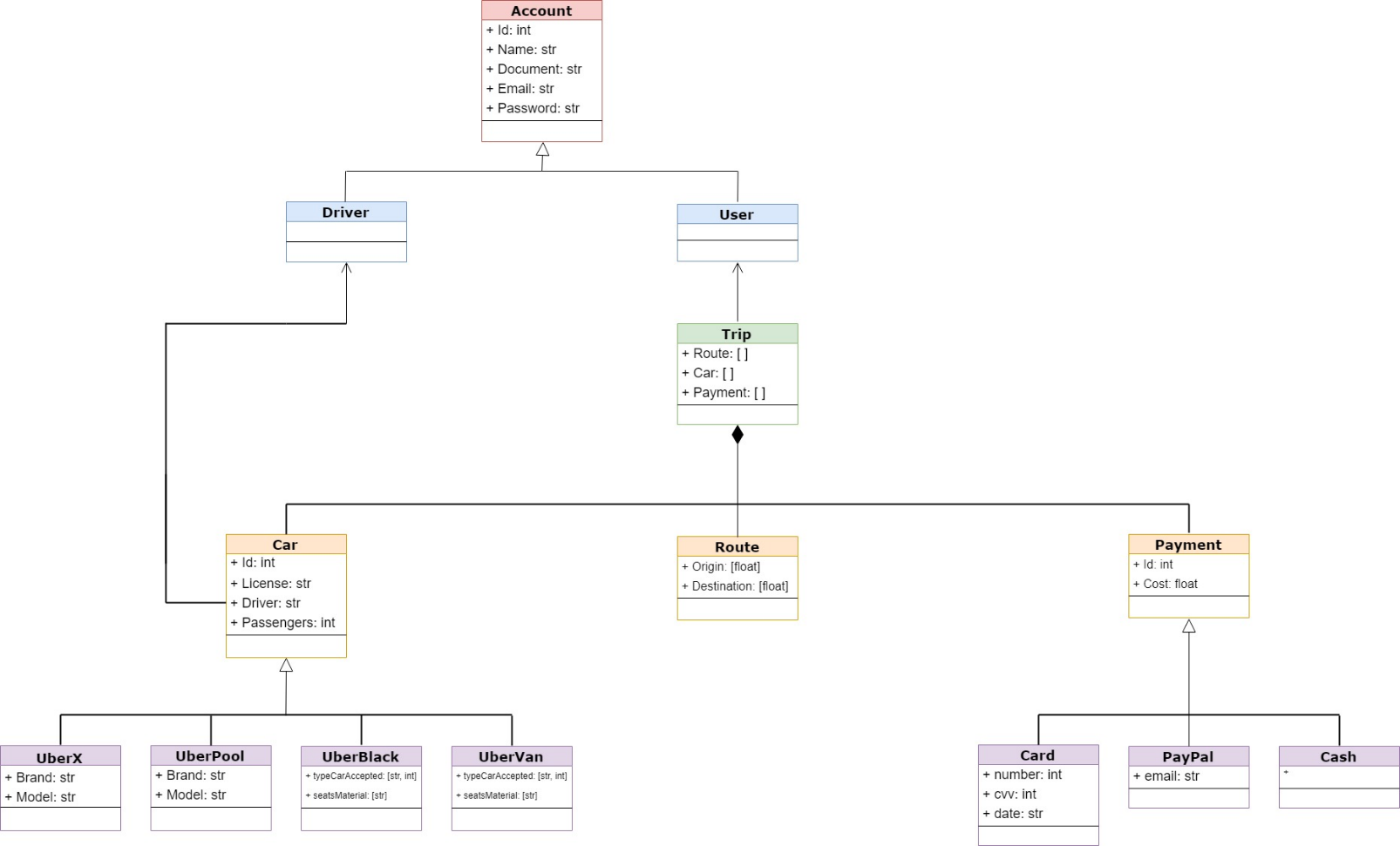
• Comportamiento, que serán las operaciones de la clase.



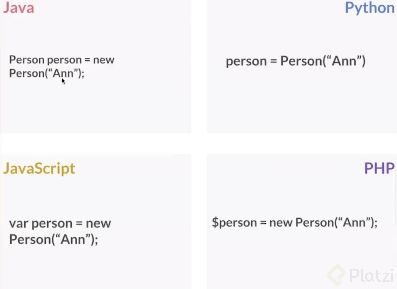


* **HERENCIA:** Mecanismo utilizado en la programación que nos permite crear nuevas clases a partir de una existente.
* Diagrama

  Descripción generada automáticamente**ABSTRACCIÓN:** Práctica de aislar un elemento de su contexto, o el resto de elementos que lo acompañan consiguiendo de ese modo, reducir ese elemento a sus características mas esenciales



**OBJETO:** Un objeto es una instancia de la clase. Es decir, el objeto es el resultado de lo que modelamos o los parámetros que dejamos declarados en la clase.



* **Método constructor**

1. Da un estado inicial de objeto (Crea un espacio en memoria para nuestro objeto).
2. Tiene el mismo nombre de la clase.
3. Son los parámetros mínimos que necesita el objeto para que pueda vivir.