

Curso de **Programación Orientada a Objetos con C++**

Diana Martínez



Introducción al curso





Bjarne Stroustrup

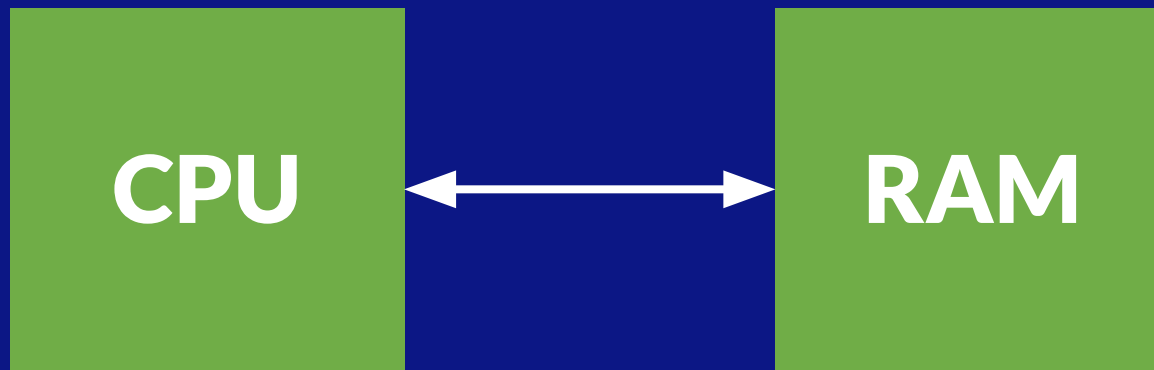


Manejo de memoria

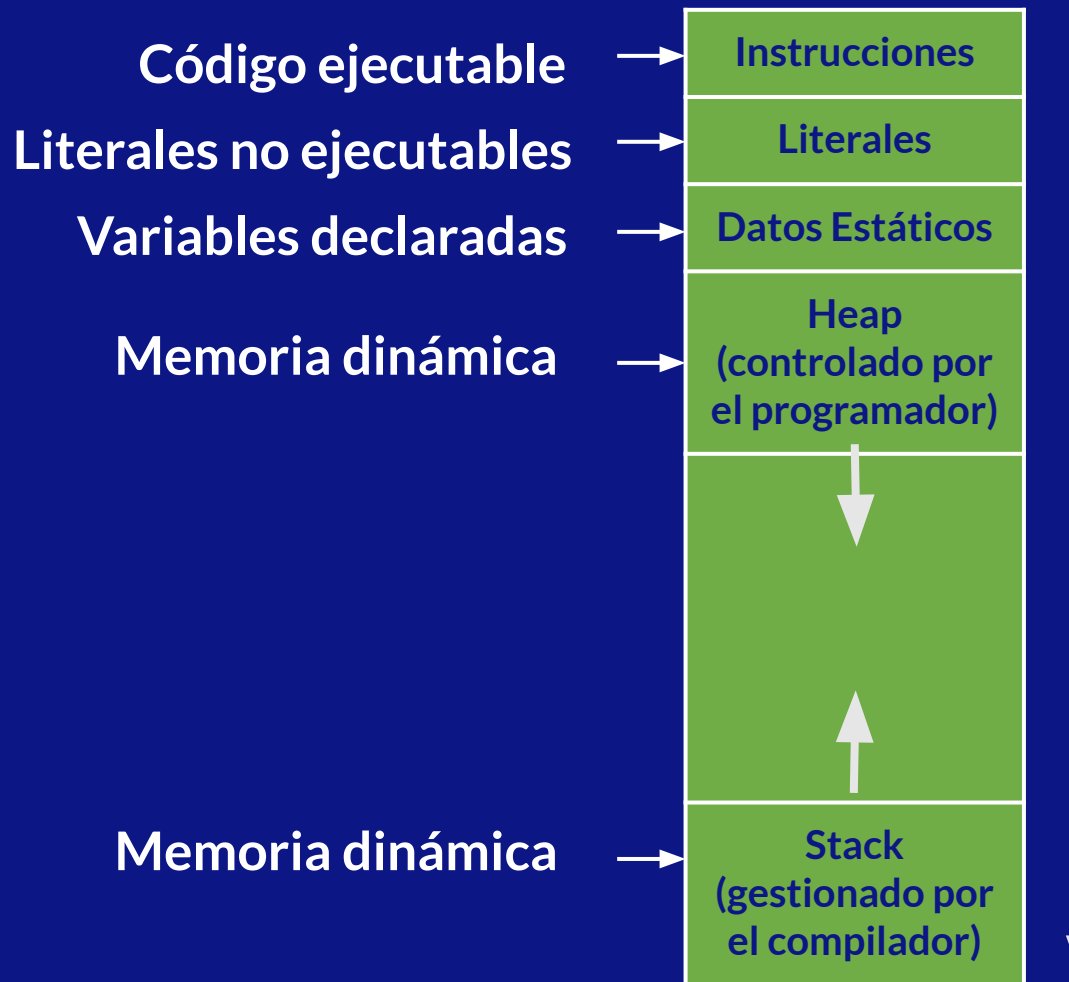
Punteros



Conexión entre CPU y RAM



Segmentación de la memoria



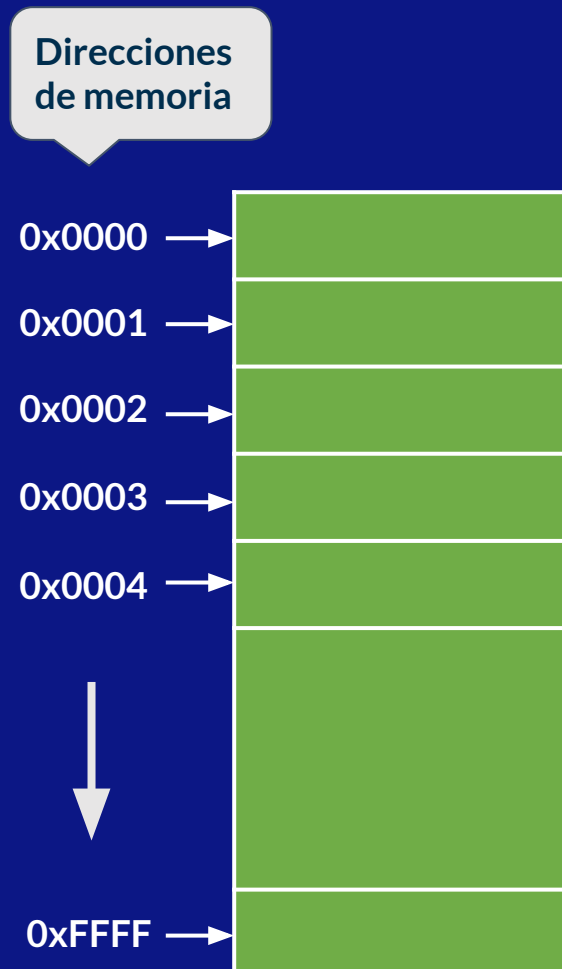


Punteros

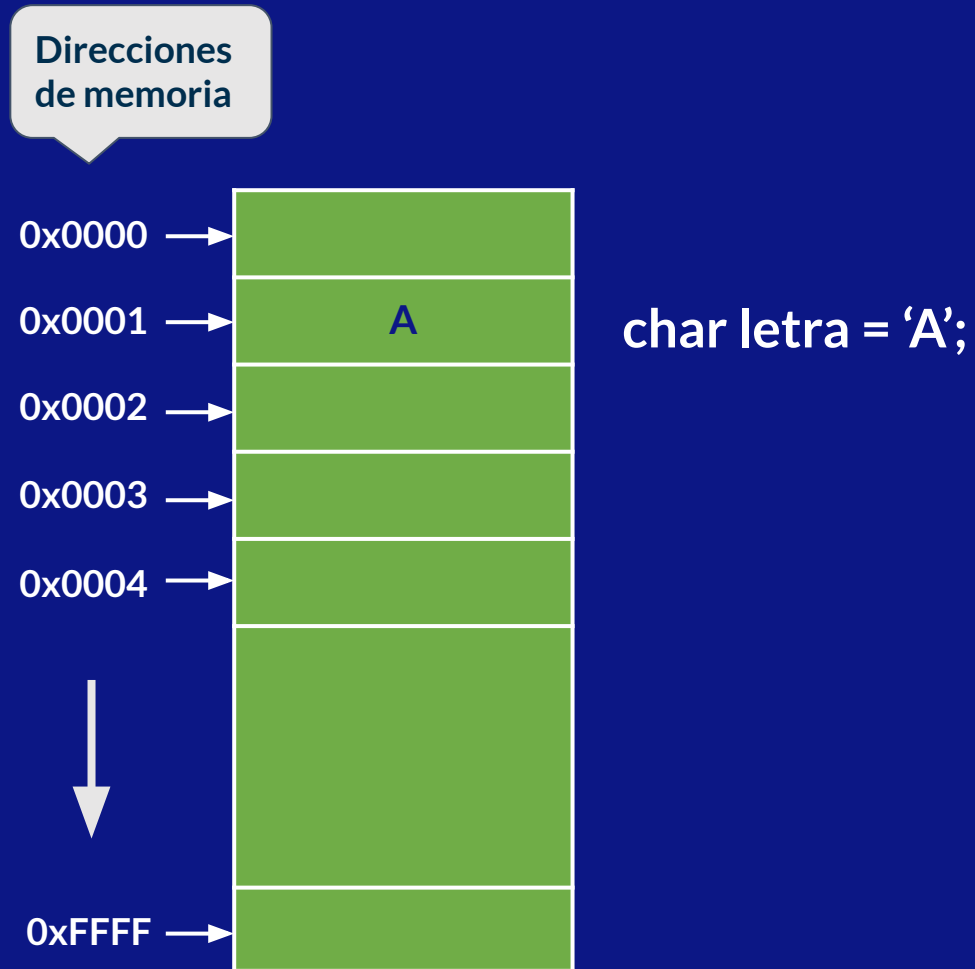
Un puntero es una variable que almacena una dirección de memoria.



Estructura básica de la RAM

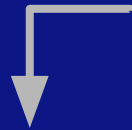


Estructura básica de la RAM



Puntero a una variable

obtiene el valor almacenado
en una dirección de memoria

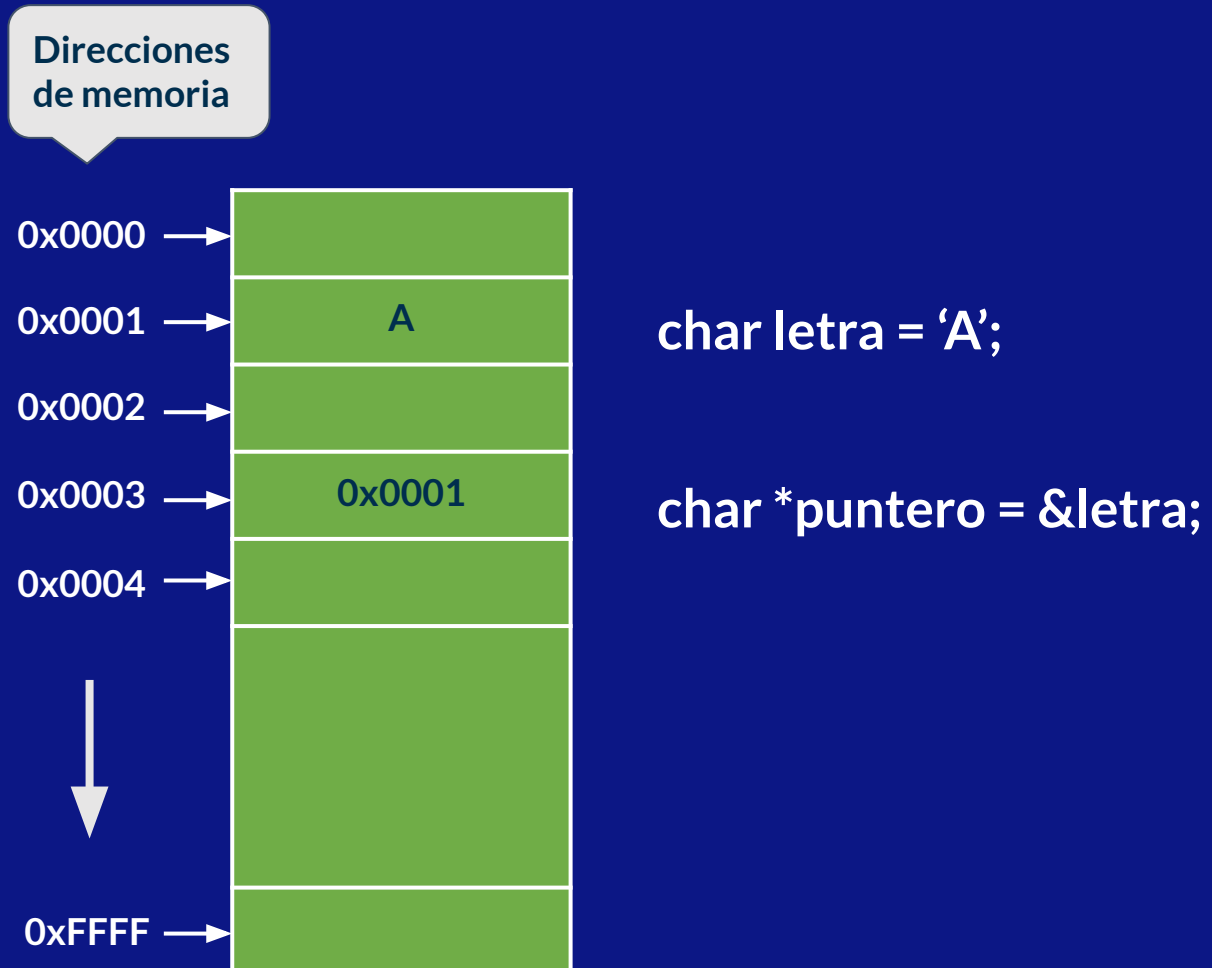


```
char *puntero = &letra;
```

obtiene la dirección de
memoria de una variable

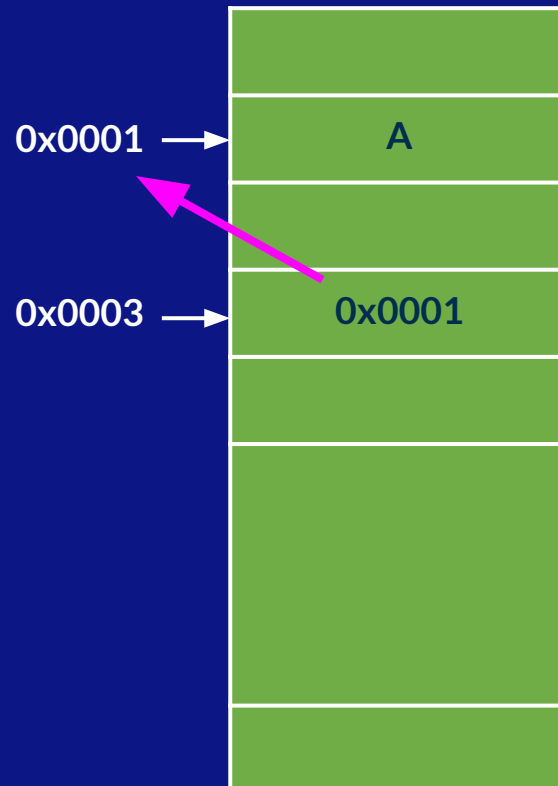


Puntero a una variable



Puntero a una variable

Direcciones
de memoria

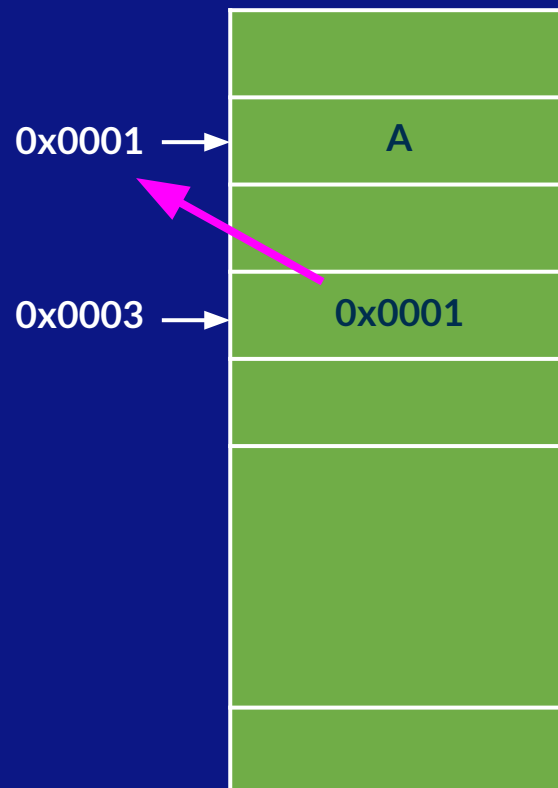


`char letra = 'A';`

`char *puntero = &letra;`

Puntero a una variable

Direcciones
de memoria



```
char letra = 'A';
```

```
char *puntero = &letra;
```

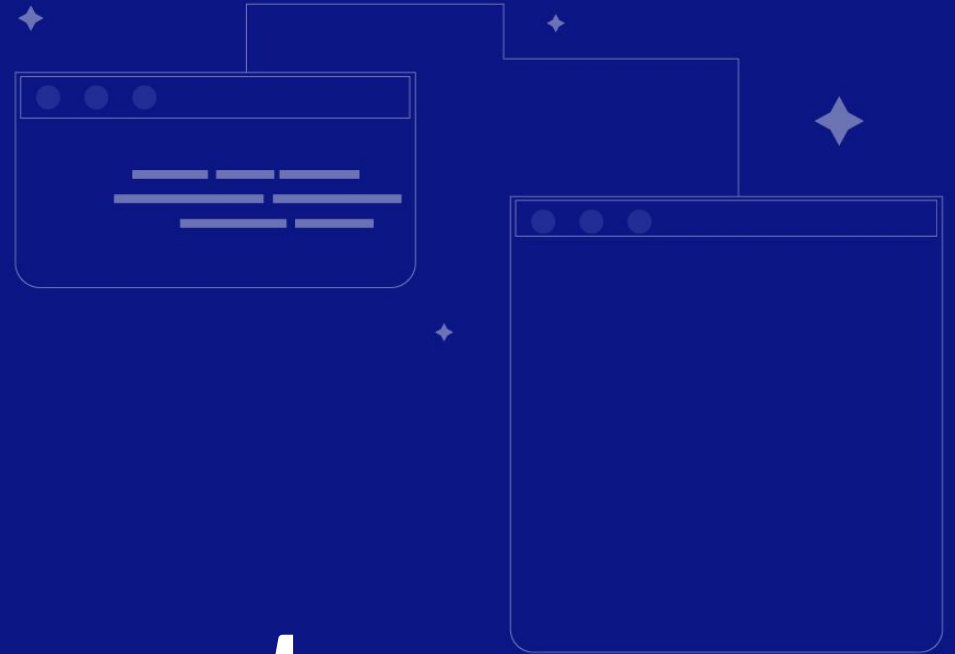
Por lo tanto:

```
&letra = 0x0001
```

```
puntero == 0x0001
```

```
*puntero == 'A'
```

```
&puntero == 0x0003
```



Usando punteros





Estructuras de datos

y tipos personalizados



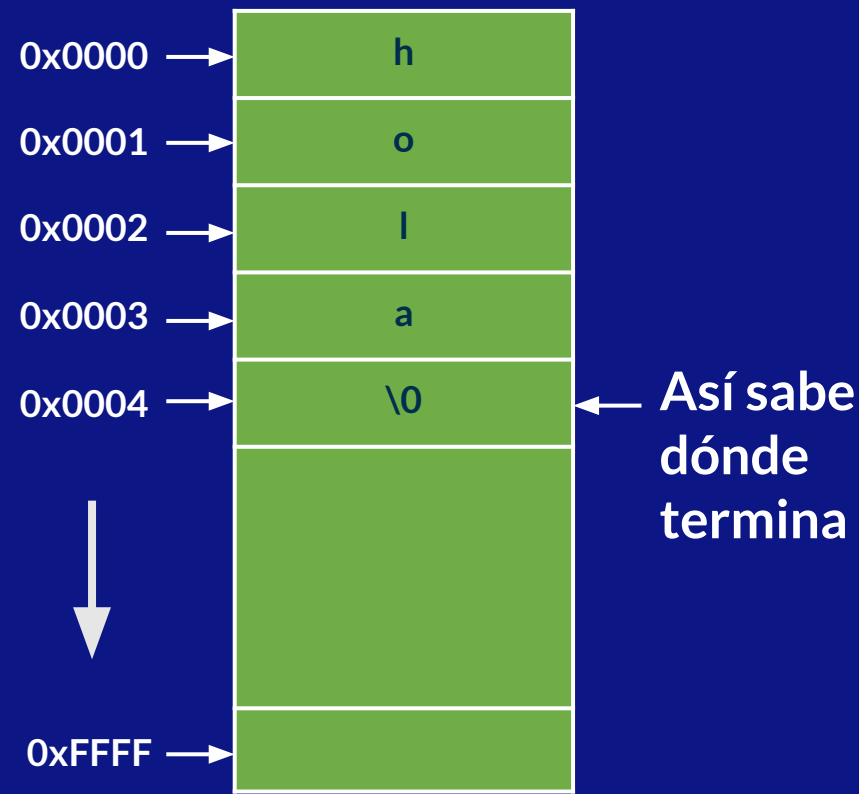
```
char texto[] = "hola";
```

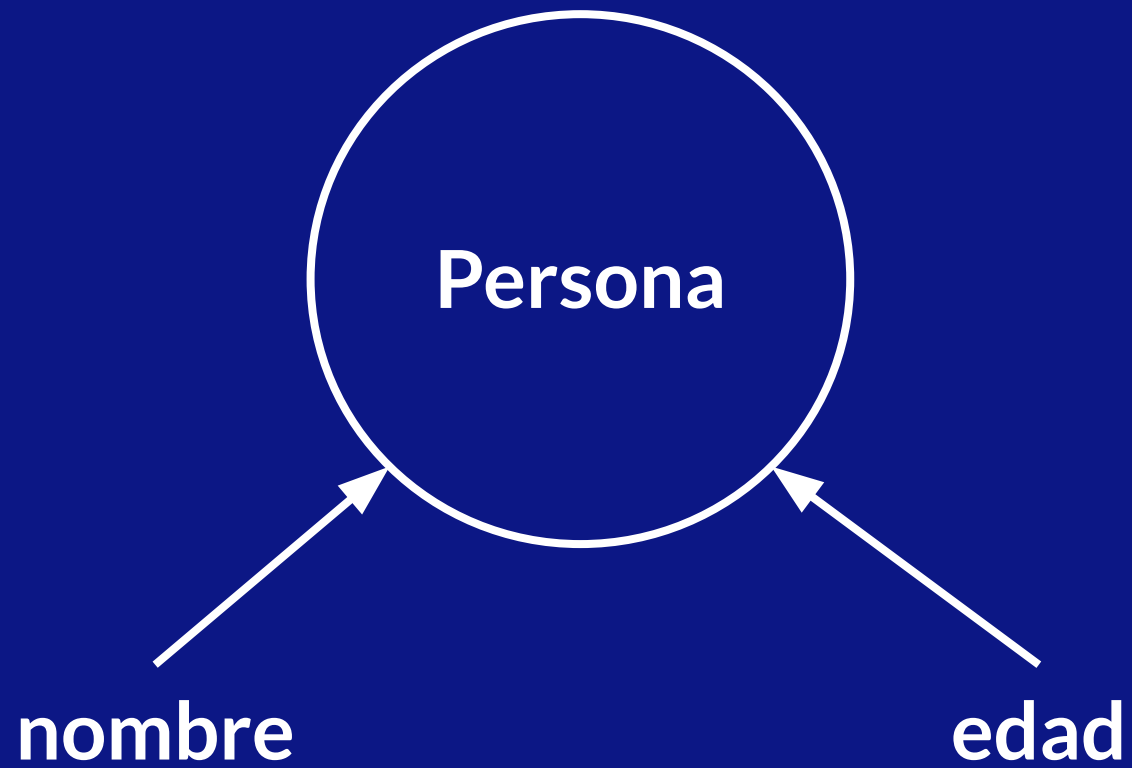


```
char texto[] = { 'h', 'o', 'l', 'a' };
```

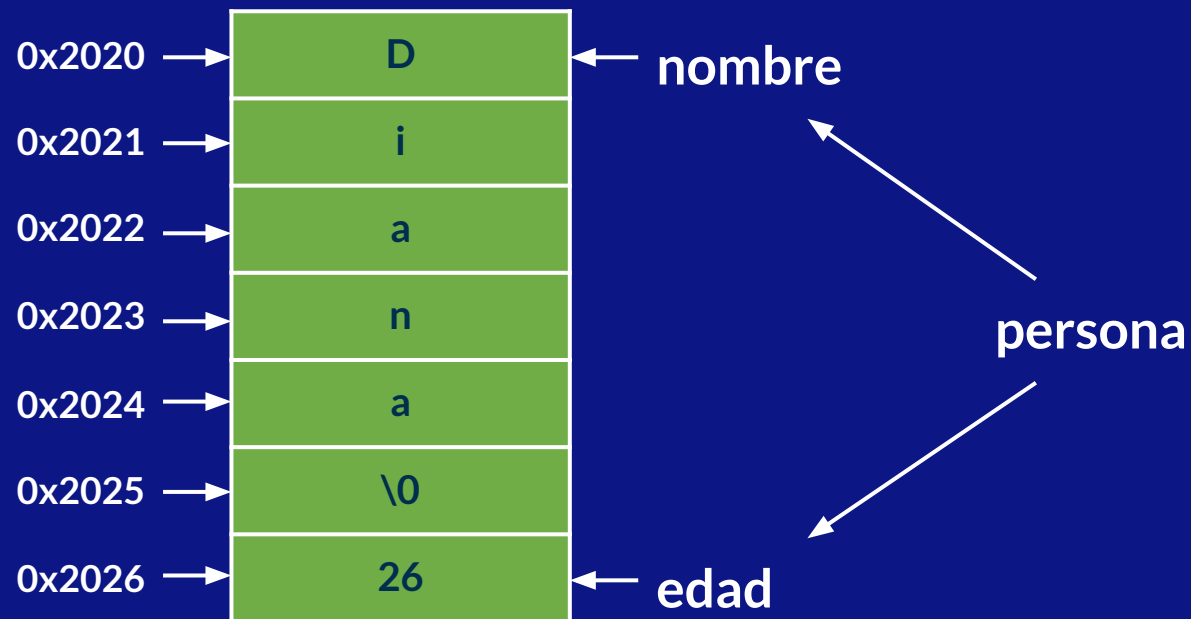
```
char *texto = (char *) "hola";
```


Representación de una cadena de caracteres en memoria





Estructura de una “persona” en memoria





Creando nuestra primera estructura




```
struct Persona {  
    char *nombre;  
    int edad;  
};
```

```
Persona *P = new Persona;
```

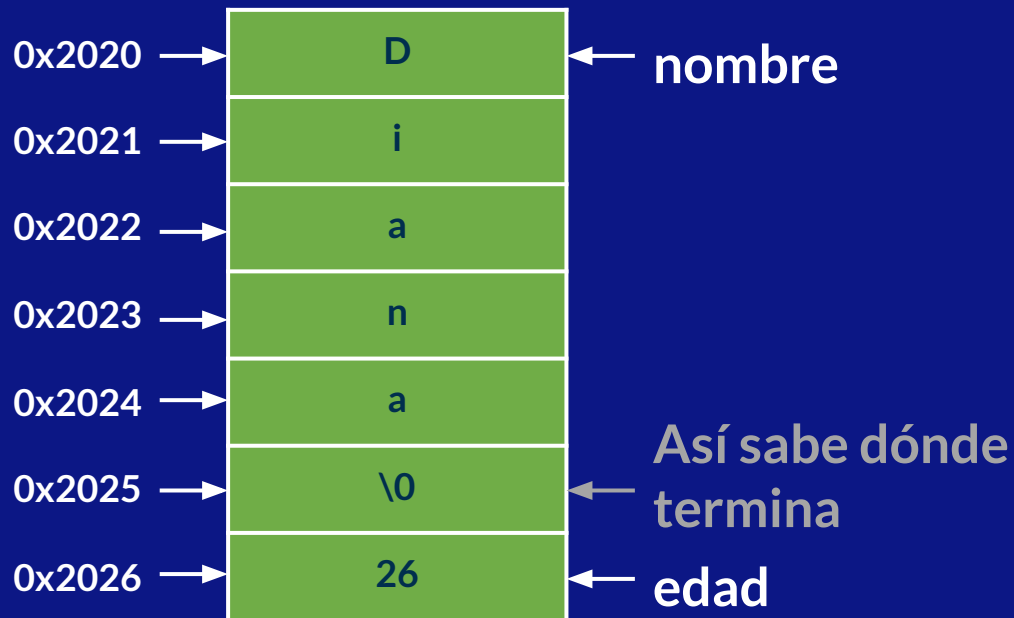


Reserva el espacio necesario en
memoria

 Direcciona al puntero
del valor miembro

```
P->nombre = (char *) "Diana";  
P->edad = 26;
```

Estructura de una “persona” en memoria



Por lo tanto:

```
P == 0x2020
```

```
&P->nombre == 0x2020 &P->edad  
== 0x2026
```

```
sizeof P == 8 bytes
```

```
sizeof &P->nombre == 8 bytes
```

```
sizeof &P->edad == 8 bytes
```

```
sizeof Persona == 16 bytes
```

* En este ejemplo el compilador es capaz de inferir el tamaño del nombre dado que es un valor literal.

Uniones y enumeraciones



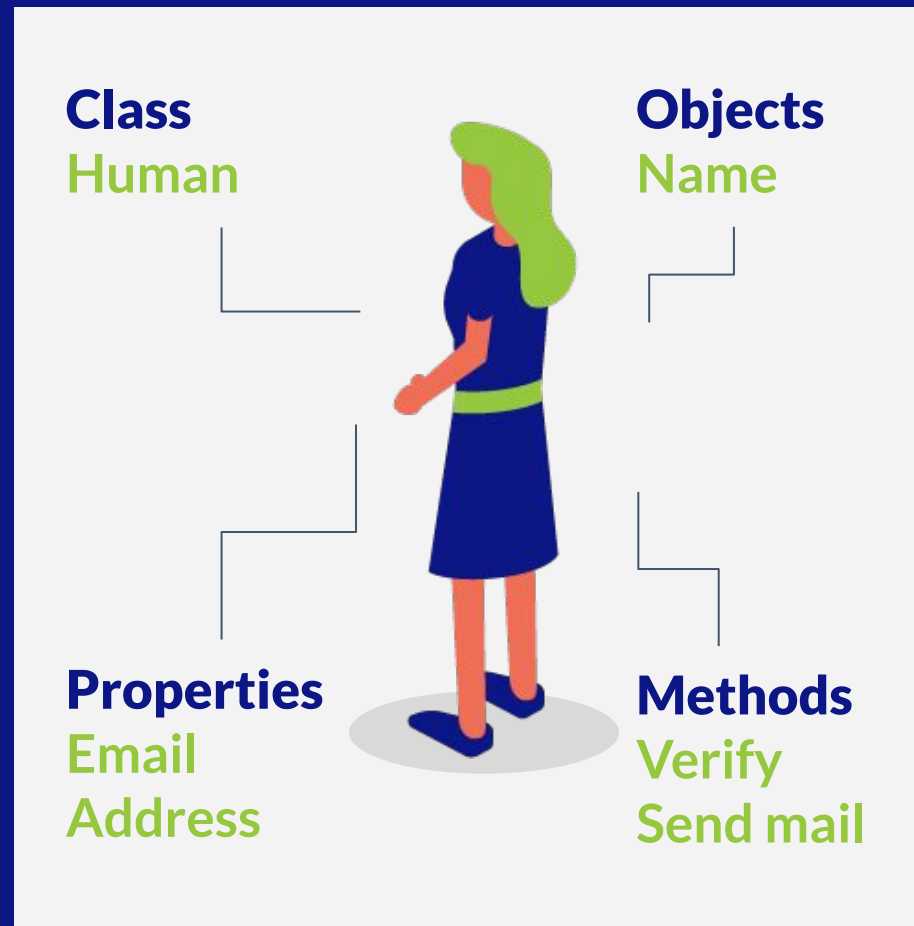


¿Qué es la POO?

Programación Orientada a Objetos



Object-oriented programming





Objetos en C++





Abstracción

Es la capacidad de realizar programas a partir de definiciones abstractas, por medio de solo mostrar la información esencial y ocultar los detalles de implementación.





Encapsulación

Es la capacidad de agrupar datos bajo una misma unidad.

Una clase puede ocultar características para hacerlas inaccesibles a las demás.



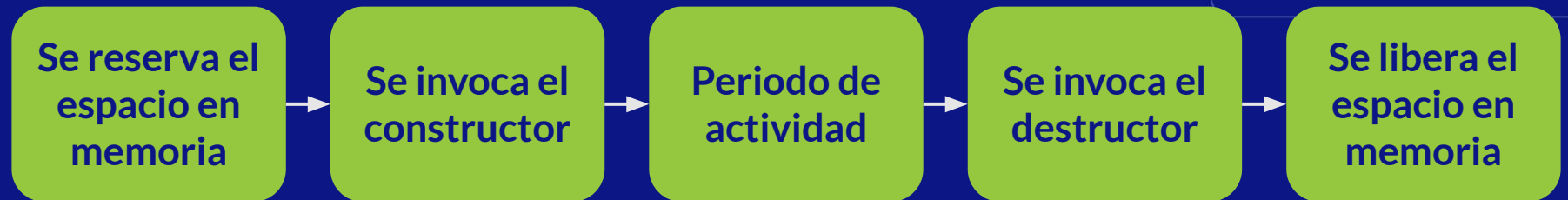
Creando nuestro primer objeto



El ciclo de vida de un objeto



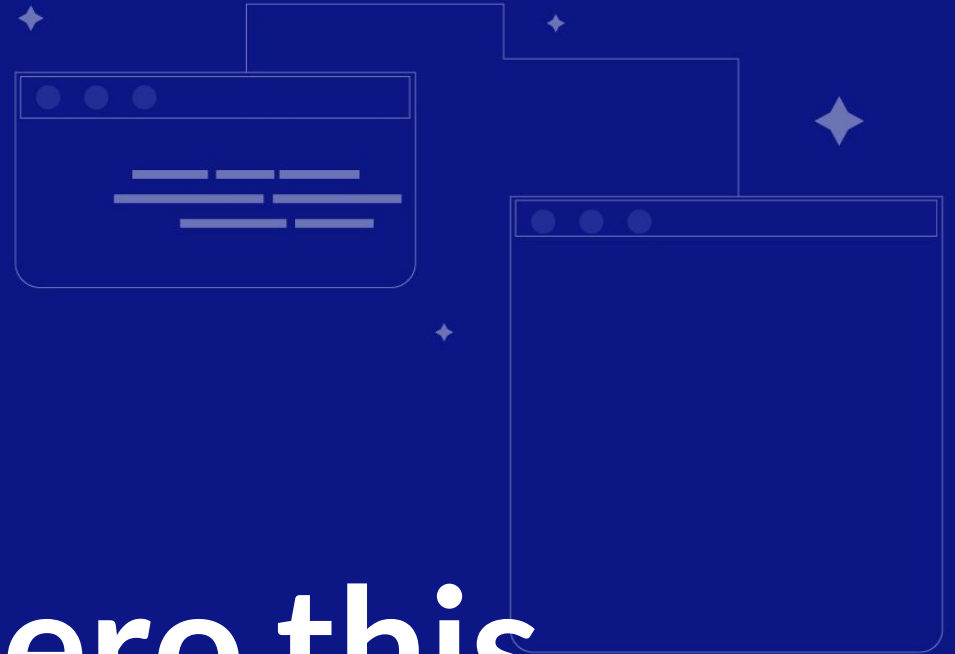
El ciclo de vida de un objeto

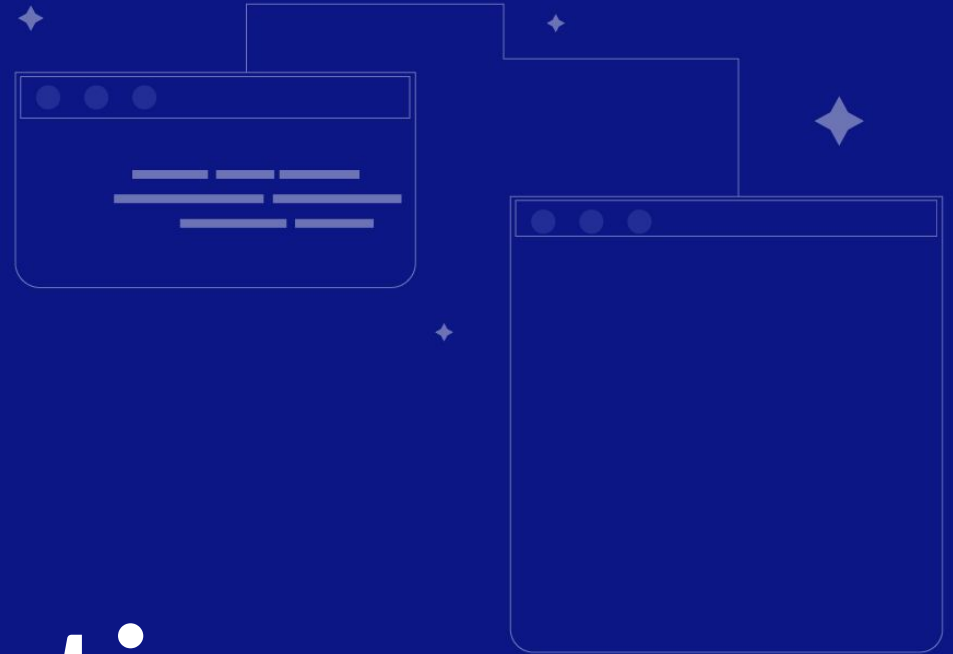


Encapsulación



El puntero this





Static



Herencia

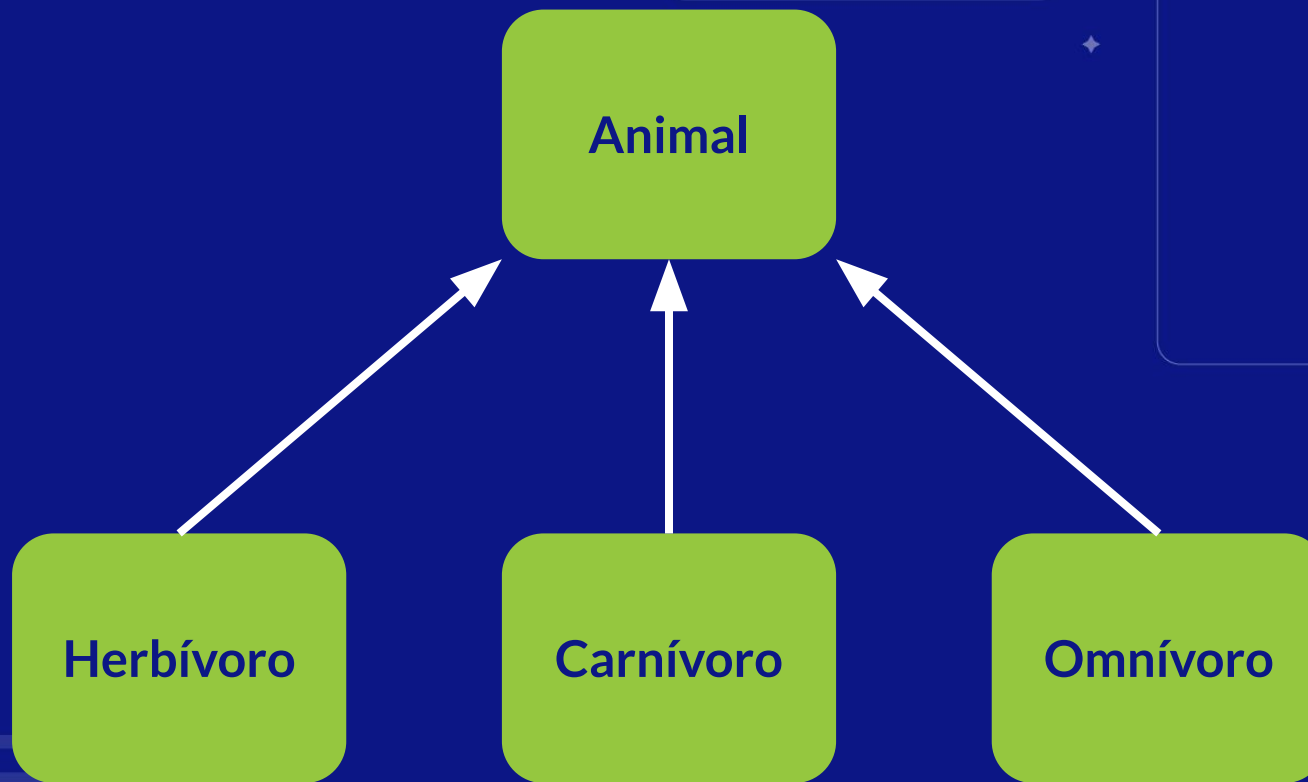




Herencia

Es la capacidad de una clase de heredar características de una o más clases base.



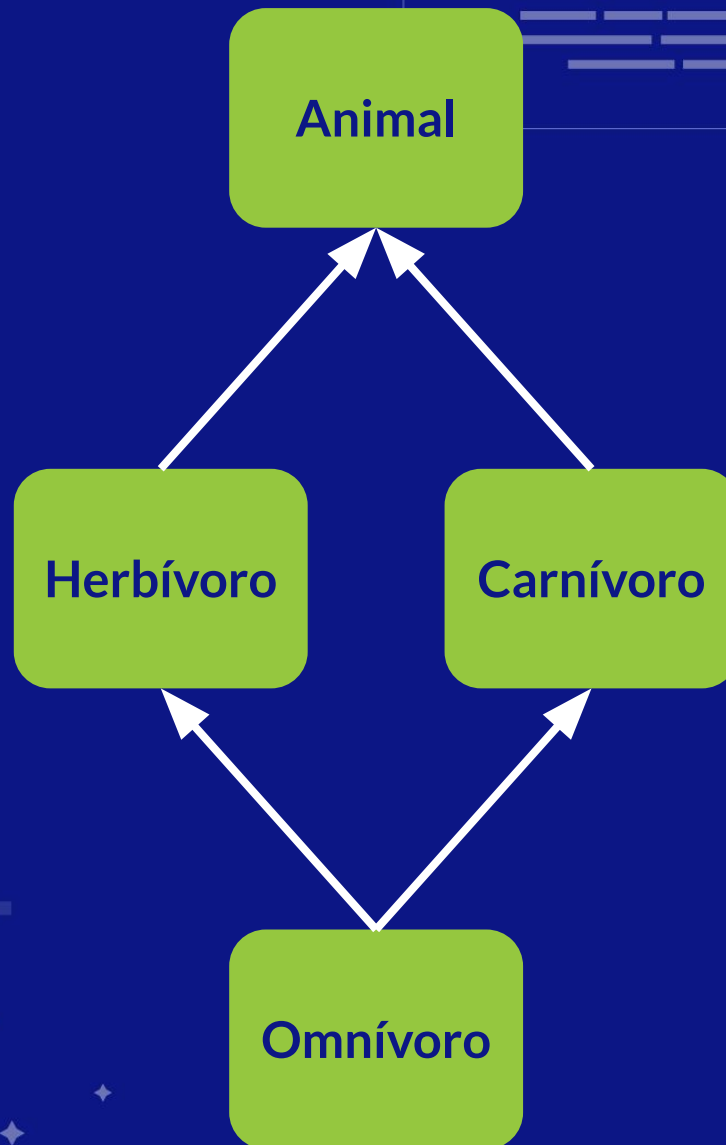




Herencia múltiple

Objetos que heredan de múltiples
padres







Polimorfismo

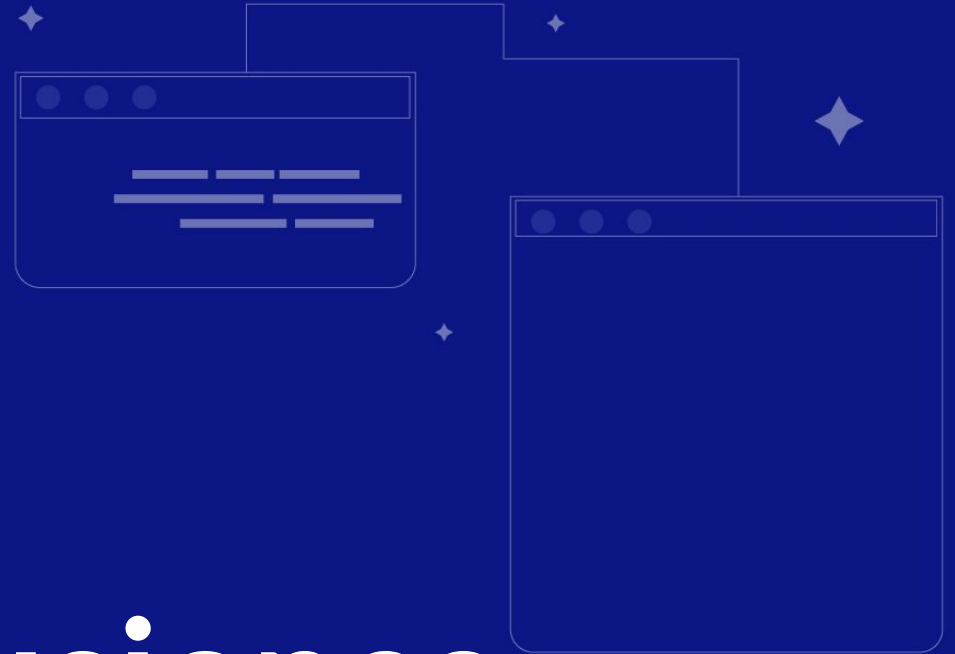




Polimorfismo

Es la característica que permite a objetos de diferente clase intercambiar mensajes con una misma estructura.





Conclusiones

