

Estructuras

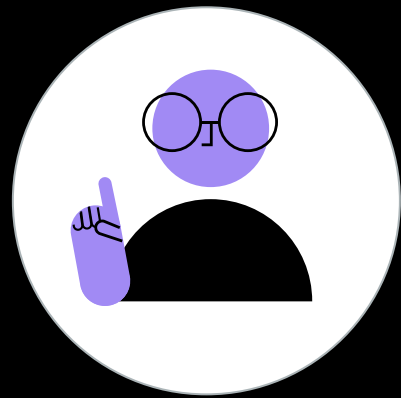
Índice

- 01** [¿Qué son las estructuras?](#)
- 02** [Aplicar las estructuras en Go](#)



01

¿Qué son las
estructuras?



Una **estructura** es
un conjunto tipado
de campos de datos.

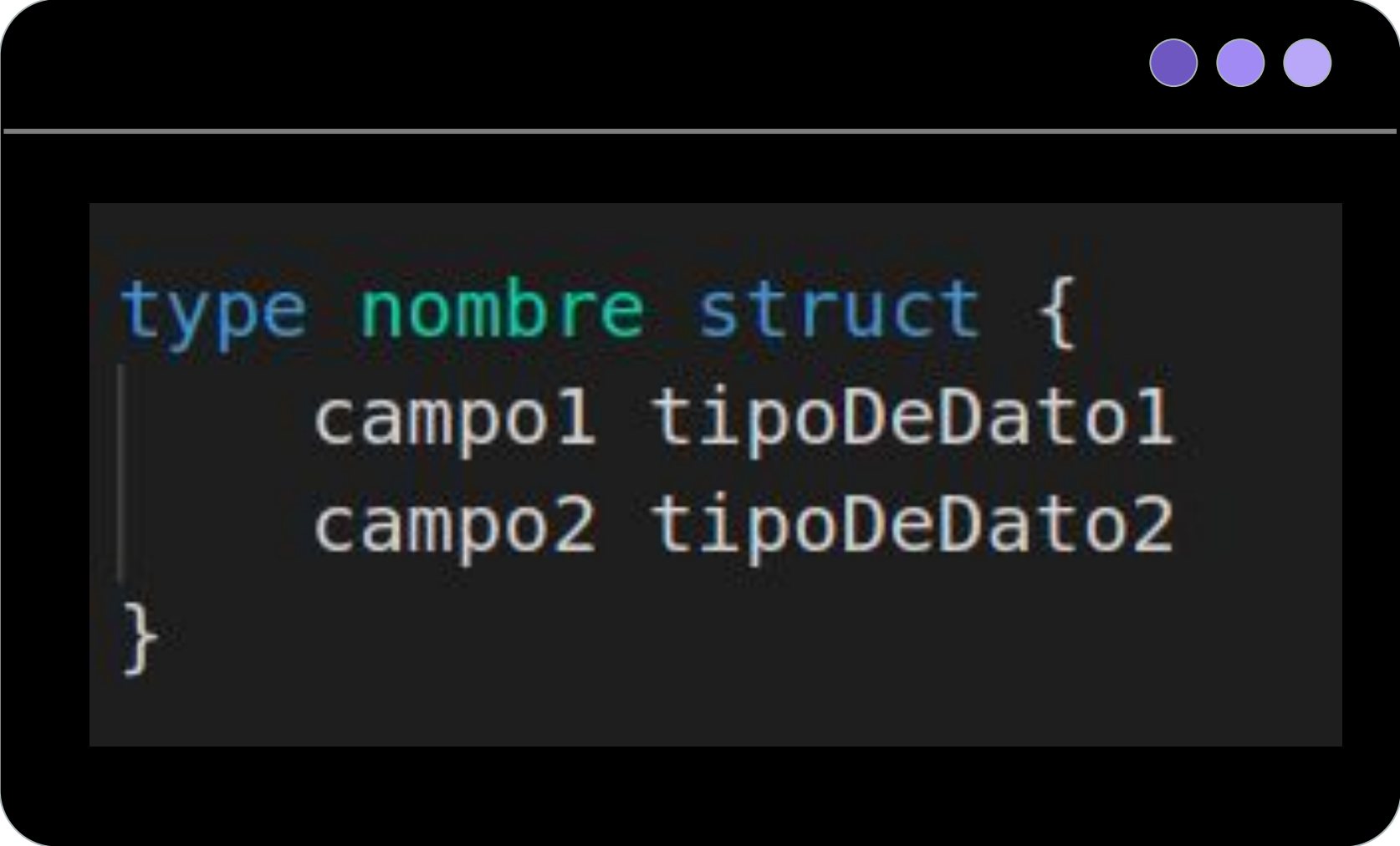
Por ejemplo, podemos definir una estructura “persona” y en ella tener valores como edad, peso, género, profesión, etc. Estas son útiles para agrupar datos, como formar registros personalizados.

Una estructura consta de tipos integrados y definidos por el usuario (la estructura en sí es un tipo definido por el usuario).

Declaración de estructuras

Ahora que sabemos qué es una estructura y por qué se utiliza, es hora de aprender cómo declarar estructuras. En la imagen de la derecha, podemos ver el esqueleto básico de una estructura:

type y **struct** son palabras clave, mientras que la estructura contiene varios campos con su tipo de datos definido.

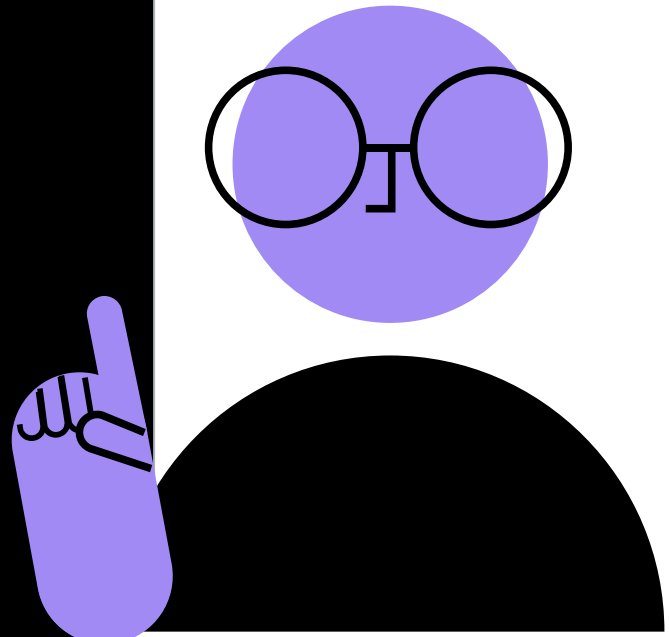


```
type nombre struct {  
    campo1 tipoDeDato1  
    campo2 tipoDeDato2  
}
```

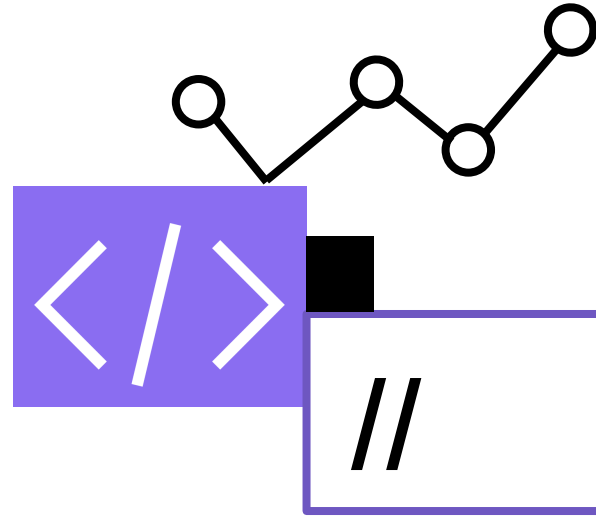
02

Aplicar las estructuras en Go

Definimos una estructura de la siguiente manera:
determinamos sus campos, seguido de un espacio y el tipo de dato. Para separar cada campo utilizamos un salto de línea. Si la primera letra es mayúscula, significa que es accesible desde otro paquete, y aplica la misma lógica para los atributos.



```
type Persona struct {  
    Nombre    string  
    Genero    string  
    Edad      int  
    Profesion string  
}
```



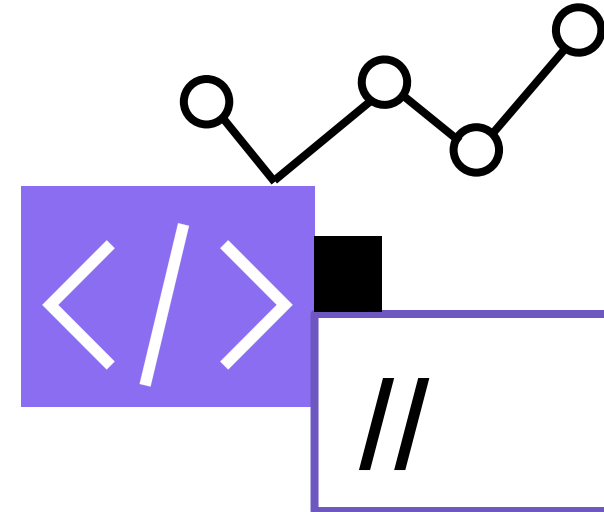


Para instanciar una estructura, podemos utilizar distintas formas:

- Indicar todos los valores que queremos que tengan los campos:
- Definir los valores para el campo que corresponda. De esta manera podemos no asignar valores a todos los campos y, de ser así, los valores quedarán por defecto según el tipo de dato.

```
{  
  p1 := Persona{"Celeste",  
    "Mujer", 34, "Ingeniera"}  
}
```

```
{  
  p2 := Persona{  
    Nombre: "Nahuel",  
    Genero: "Hombre",  
    Edad: 30,  
    Profesión: "Ingeniero",  
  }  
}
```

Para acceder a un campo de la estructura procedemos de la siguiente manera:

```
{ } p2.Edad
```

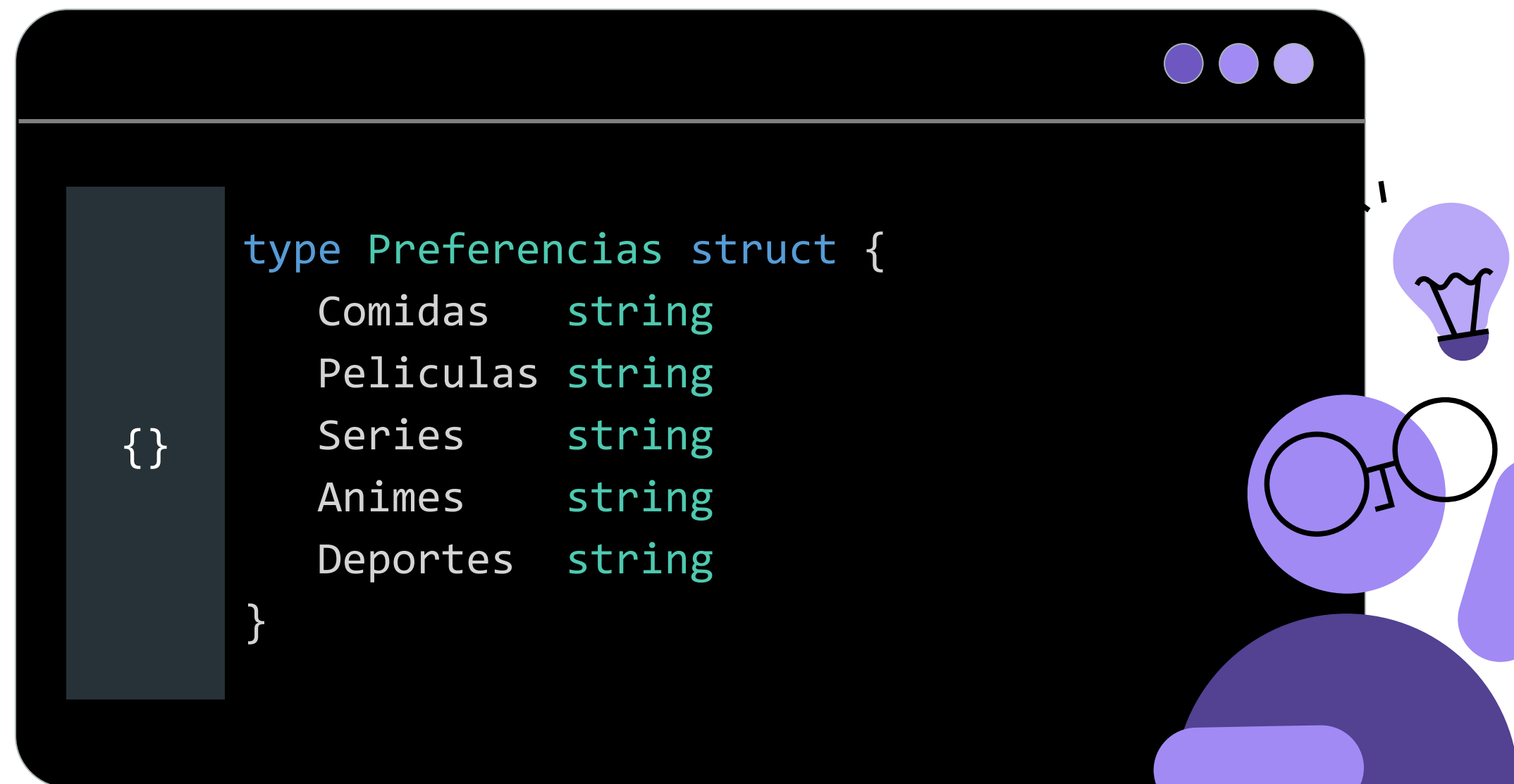
Podemos asignar o modificar un valor a un campo de la estructura de la siguiente manera:

```
{ } p2.Edad = 33
```

También podemos definir una estructura vacía e ir asignando los valores.

```
{ }  
var p3 Persona  
p3.Nombre = "Ulises"  
p3.Edad = 15
```

Podemos utilizar las estructuras como un tipo de dato. Por ende, podríamos tener estructuras como campos dentro de otra estructura. A eso se lo denomina “composición”. Por ejemplo, podemos tener una estructura **gustos** dentro de nuestra estructura **persona**. Para eso debemos declarar nuestra estructura **gustos**.



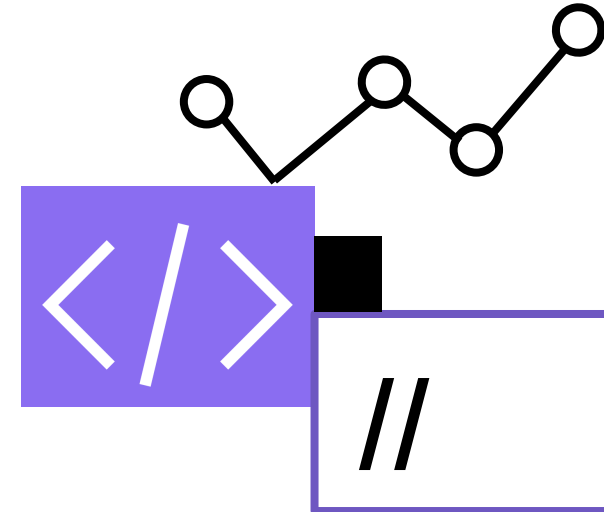


Asignaremos un campo de tipo gustos a nuestra estructura persona:

```
{}  
  
type Persona struct {  
    Nombre    string  
    Género    string  
    Edad      int  
    Profesion string  
    Peso      float64  
    Gustos    Preferencias  
}
```

Hacemos lo siguiente para instanciar nuestra estructura:

```
{}  
  
p1 := Persona{"Celeste", "Mujer",  
34, "Ingeniera", 65.5,  
Preferencias{"pollo", "Titanic",  
"", "", ""}}
```



También podemos
instanciarla haciendo
referencia a cada
campo:



```
{ }  
p2 := Persona{  
  Nombre:    "Nahuel",  
  Genero:    "Hombre",  
  Edad:      30,  
  Profesion: "Ingeniero",  
  Peso:      77,  
  Gustos: Preferencias{  
    Comidas:  "asado, pollo",  
    Peliculas: "Coco",  
    Animes:   "Shingeki no Kyojin",  
  },  
}
```



De la misma forma, para acceder a un valor o modificarlo dentro de la estructura “gustos” desde “persona”:

O podríamos agregarle directamente la estructura completa:

```
{}
```

```
fmt.Println(p2.Gustos.Animes)  
p2.Gustos.Deportes = "fútbol"
```

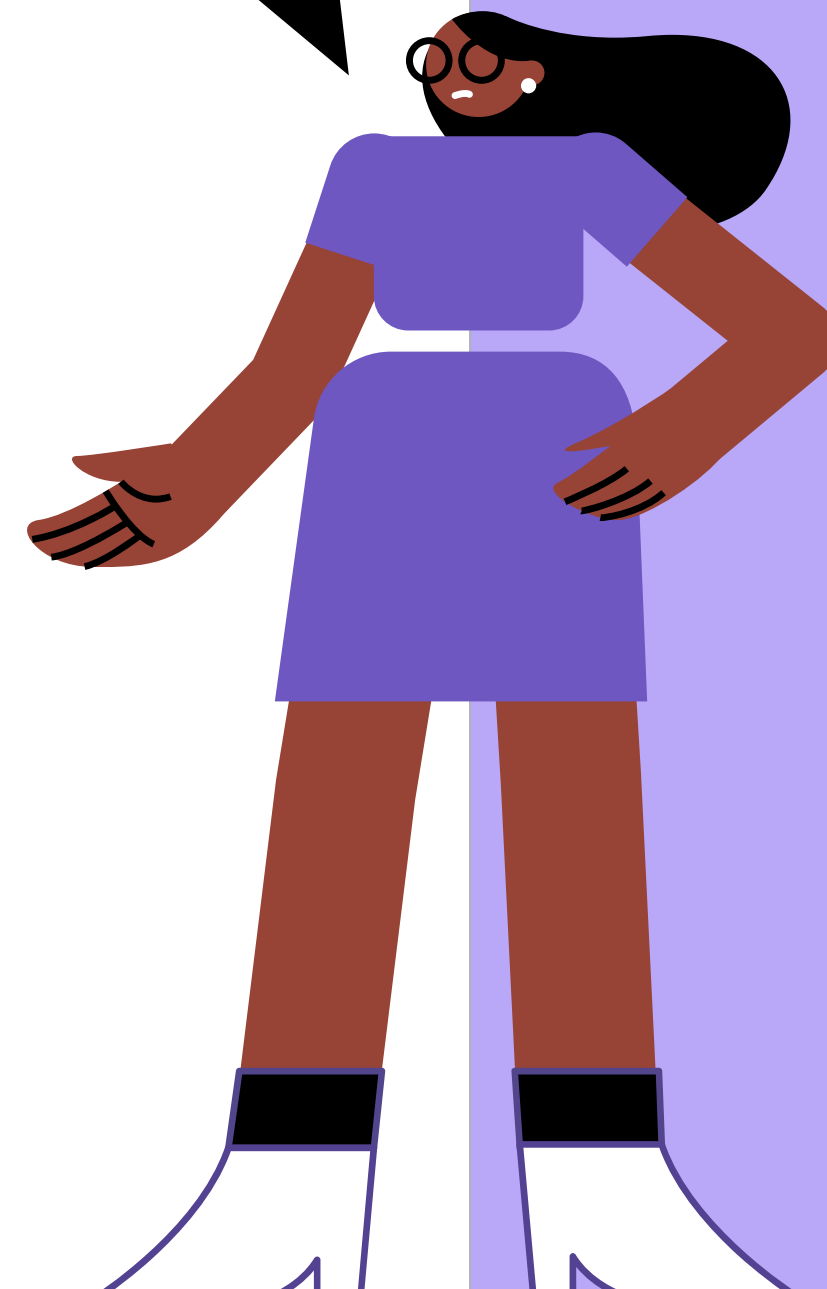
```
{}
```

```
p3 := Persona{  
  p3.Nombre = "Ulises"  
  p3.Edad = 15  
  p3.Gustos = Preferencias{Comidas:  
    "verduras", Películas: "Entrenando  
    a mi dragón"}  
}
```

Conclusiones

Ahora estamos listo para usar estructuras en Go. Cubrimos todos los conceptos básicos, como la declaración, la inicialización y el acceso a los campos de estructura. También vimos cómo implementar una estructura anidada.

Hay mucho más que aprender acerca de las estructuras, pero es un buen comienzo.



¡Muchas gracias!