



Interfaces

Golang

- # Definicion

- Es una colección de **firmas de métodos** que se puede **implementar** en un objeto, definiendo su comportamiento.

El tipo interface está enfocado en lo que puede hacer, sin tener en cuenta el objeto(type) en el cual se implementa.

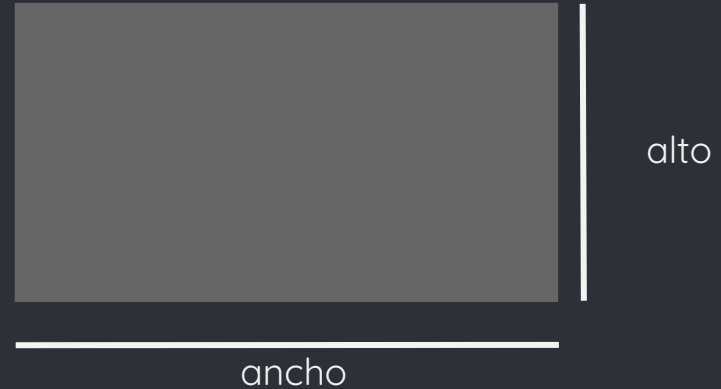
Figuras Geometricas

Circulo



```
type circulo struct {  
    radio float64  
}
```

Rectangulo



```
type rect struct {  
    ancho, largo float64  
}
```

Declaración de la Interface

Definición **abstracta** del comportamiento sin importar el tipo de figura geométrica

Firmas de
Métodos

```
type geometria interface {  
    area() float64  
    perim() float64  
}
```

Nombre

Parametros

Tipo dato
retorno

Implementación de la Interface

Objeto
Receptor

```
func (r rect) area() float64 {  
    return r.ancho * r.alto  
}  
func (r rect) perim() float64 {  
    return 2*r.ancho + 2*r.alto  
}
```

Metodo de
Interface

```
func (c circulo) area() float64 {  
    return math.Pi * c.radio * c.radio  
}  
func (c circulo) perim() float64 {  
    return 2 * math.Pi * c.radio  
}
```

Tipo de dato
retorno

Definición
particular del
comportamiento,
según el tipo de
figura geométrica

Polimorfismo

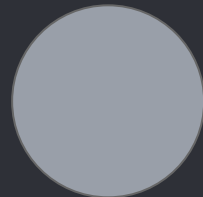
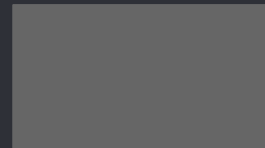
Interface en Accion

2. Recibe el objeto de tipo interface como parámetro

```
func medir(g geometria) {  
    fmt.Println(g)  
    fmt.Println(g.area())  
    fmt.Println(g.perim())  
}
```

1. Llama a función con objetos de tipo particular

```
func main() {  
    r := rect{ancho: 3, alto: 4}  
    c := circulo{radio: 5}  
  
    medir(r)  
    medir(c)  
}
```



{3 4}

{5}

12

78.53981

14

31.41592

Los objetos de tipo **rect** y **circulo** son a la vez de tipo **geometria**