

🕒 UD03 · Variables, Outputs y Estado en Terraform

Una guía visual y práctica para dominar el uso de **variables**, **outputs** y el **archivo de estado** en tus proyectos con Terraform.



1 · Parametrización en Terraform

- ⟳ **Reutilización:** Cambia valores sin tocar el código.
- ↗ **Escalabilidad:** Úsalo en diferentes entornos.

```
variable "imagen" {  
    default = "nginx:latest"  
}  
  
resource "docker_container" "web" {  
    name   = "webserver"  
    image  = var.imagen  
}
```

💡 Puedes sobrescribir desde CLI: `terraform apply -var="imagen=httpd:latest"`



2 · Variables

Define parámetros reutilizables:

```
variable "puerto_externo" {  
    type    = number  
    default = 8080  
}
```

- ⤒ **Tipos:** `string` , `number` , `bool` , `list` , `map` , `object` , `any`
- ⤓ **.tfvars:** para entornos (`dev.tfvars`, `prod.tfvars...`)

- ✓ CLI tiene prioridad más alta: `-var="clave=valor"`

🛡️ Validación

```
variable "imagen" {  
  type = string  
  validation {  
    condition      = contains(["nginx:latest", "httpd:latest"], var.imagen)  
    error_message = "Solo nginx o httpd están permitidos"  
  }  
}
```

{*} 3 · Locals

Valores internos reutilizables para evitar repetición:

```
locals {  
  prefijo = "${var.project}-${var.env}"  
}
```

- ✓ Código más limpio
- ⌚ Cálculos y condiciones

```
locals {  
  imagen = var.env == "prod" ? "nginx:stable" : "nginx:alpine"  
}
```

⬇️ 4 · Outputs

Valores de salida tras aplicar infraestructura:

```
output "ip_publica" {  
  value = aws_instance.web.public_ip  
}
```

- Ver: `terraform output`
- Sensible: `sensitive = true`

```
output "secret" {  
    value      = random_password.secret.result  
    sensitive = true  
}
```

5 · Estado (`terraform.tfstate`)

Archivo donde Terraform guarda lo que existe realmente:

- Ver con `terraform show`
- Ver recursos: `terraform state list`

Estado remoto

```
terraform {  
    backend "s3" {  
        bucket = "tf-estado"  
        key    = "dev/infra.tfstate"  
        region = "us-east-1"  
    }  
}
```

Nunca subas `terraform.tfstate` a Git

6 · Providers útiles: `random` & `templatefile`

Provider `random`

```
resource "random_pet" "name" {  
    length = 2  
}  
output "nombre" {
```

```
    value = random_pet.name.id  
}
```

templatefile()

Usa archivos `.tmpl` externos con variables:

```
locals {  
    mensaje = templatefile("${path.module}/saludo tmpl", {  
        nombre = "Carlos"  
        id     = "123"  
    })  
}
```

Bonus: Funciones útiles

- `join("-", ["web", "01"])` → "web-01"
- `format("srv-%03d", 7)` → "srv-007"
- `file("config.json")` → contenido del archivo

 [Cheat Sheet UD03](#)