# Módulos en Terraform

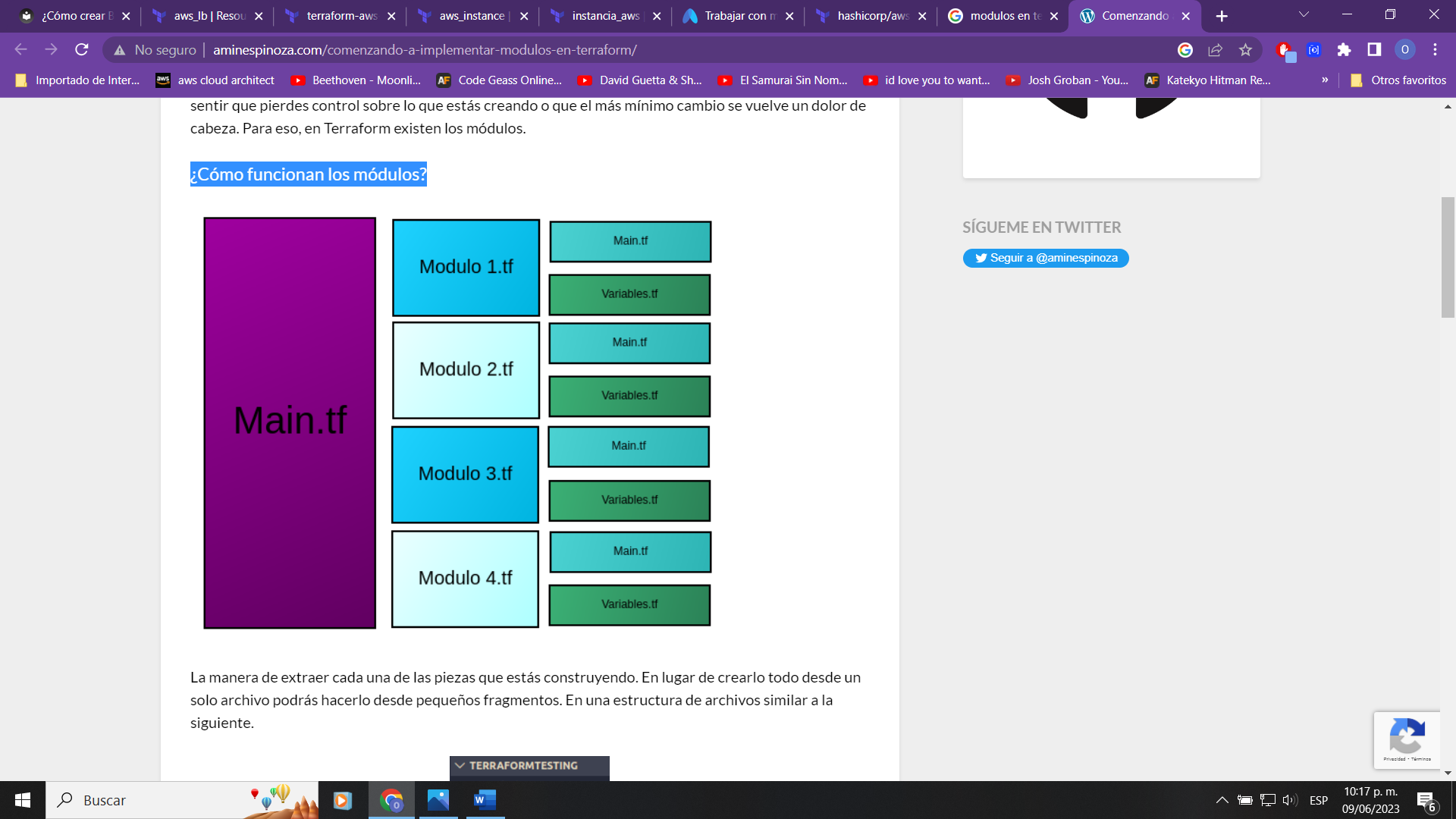
# ¿Qué son los módulos?

Podemos entenderlo como un conjunto de ficheros de terraform que, al ser aplicados con ciertos argumentos, aprovisionan los recursos en la nube y generan una información de salida.

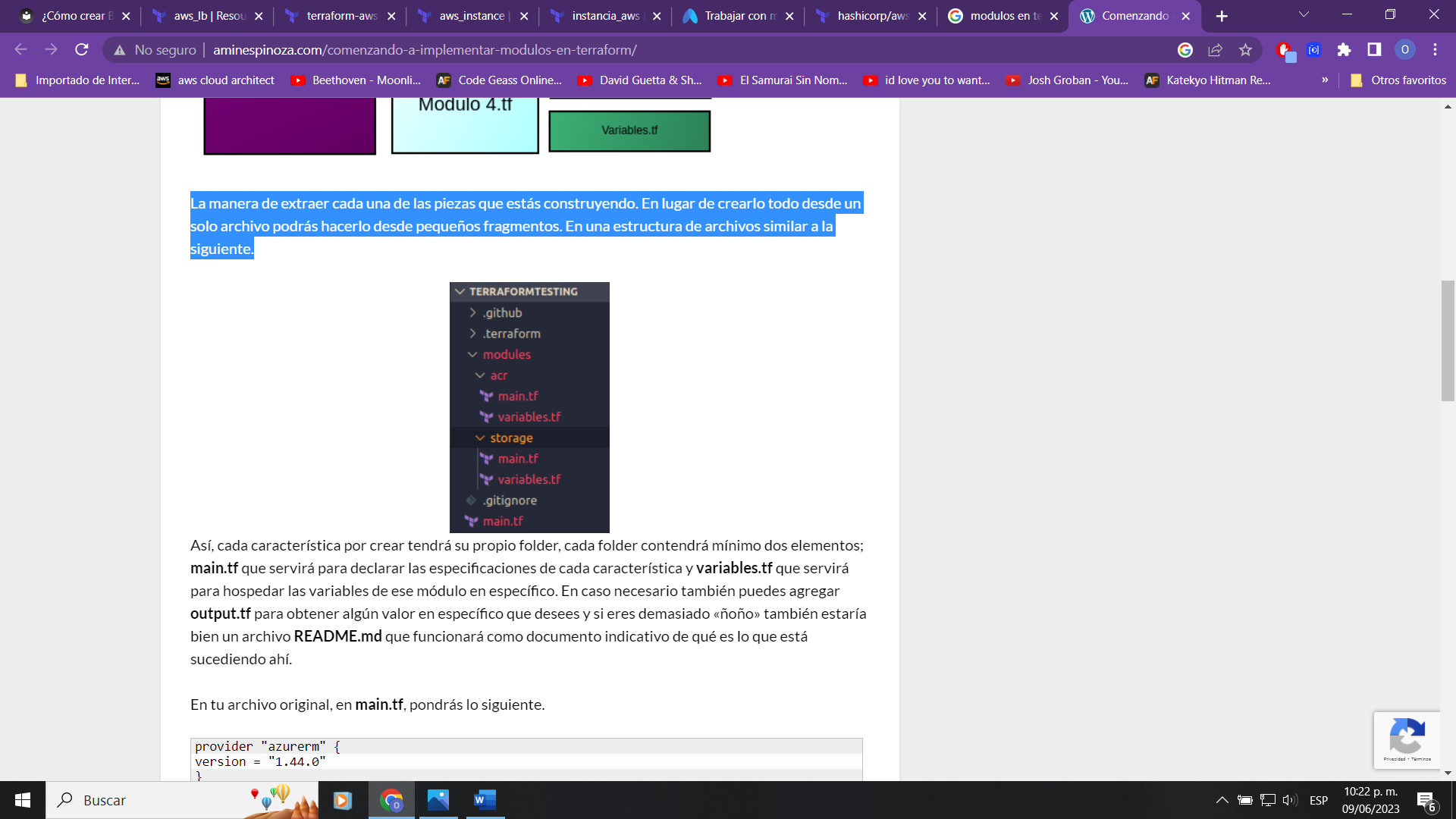
Existen ficheros con extensión .tf dentro de un directorio como un modulo

Lo que debemos tener en cuenta que existen argumentos de entradas y de salidas como son los inputs y los outputs.

### ¿Cómo funcionan los módulos?



En lugar de crearlo todo desde un solo archivo se podrá hacer desde pequeños fragmentos. En una estructura de archivos similar a la siguiente.



# Crear recursos en terraform

Primero crearemos dos archivos con la extensión **.tf** en un folder vacío. Uno con el nombre **main.tf** y otro llamado **variables.tf**.

**Dentro el archivo main.tf se colocara lo siguiente :**

|  |
| --- |
| provider "aws" {  region = "us-west-2"  }  resource "aws\_resource\_group" "rg"{  name = var.resource\_group\_name  tags = {  Name = "MyResourceGroup" } } |

**Y en el archivo variables.tf coloca lo siguiente.**

|  |
| --- |
| variable "resource\_group\_name" {  type = string description = "Desired name for your group" default = "testingGroup"  }  variable "location" {  type = string  description = "Desired area location"  default = "us-west-2"  }  provider "aws" { region = var.location  }  resource "aws\_resource\_group" "rg" {  name = var.resource\_group\_name  tags = {  Name = "MyResourceGroup"  }  } |

# Recursos para desplegar un ALB en AWS

## Aws\_lb:

* Se puede hacer que el balanceador de carga sea privado, al que solo se puede acceder dentro de VPC ,o hacerlo público para aceptar solicitudes de Internet.
* Subnets: Define las subredes en las que se desplegará el balanceador de carga.
* Enable\_deletion\_protection: Habilita la protección contra eliminación accidental para el balanceador de carga, estableciéndolo en true.
* Access\_logs: Configura los registros de acceso y q Opcionalmente se le puede habilitar y almacenar en un depósito S3 .

resource "aws\_lb" "test" {

name = "test-lb-tf"

internal = false

load\_balancer\_type = "application"

security\_groups = [aws\_security\_group.lb\_sg.id]

subnets = [for subnet in aws\_subnet.private : subnet.id]

enable\_deletion\_protection = true

access\_logs {

bucket = aws\_s3\_bucket.lb\_logs.id

prefix = "test-lb"

enabled = true

}

tags = {

Environment = "production"

}

}

## Aws\_Lb\_Listener

Estas entradas de información necesitan ser escuchadas y en este caso las escuchan por el puerto 80 en donde utiliza el protocolo http . cuando el trafico llega a este puerto se redirecciona al puerto 443 usando protocolos de https .

resource "aws\_lb\_listener" "front\_end" {

load\_balancer\_arn = aws\_lb.front\_end.arn

port = "80"

protocol = "HTTP"

default\_action {

type = "redirect"

redirect {

port = "443"

protocol = "HTTPS"

status\_code = "HTTP\_301"

}

}

}

## Route 53

Pera proteger la aplicación con https necesitaremos crear un dominio personalizado y obtener un certificado TLS

data "aws\_route53\_zone" "selected" {

name = "test.com."

private\_zone = true

}

resource "aws\_route53\_record" "www" {

zone\_id = data.aws\_route53\_zone.selected.zone\_id

name = "www.${data.aws\_route53\_zone.selected.name}"

type = "A"

ttl = "300"

records = ["10.0.0.1"]

}