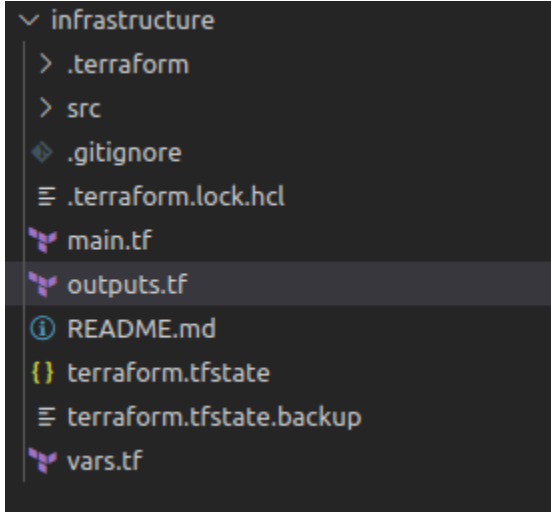


# Implementación API: Terraform

Vamos a explicar brevemente como aprovisionar la infraestructura utilizando la herramienta IaC (Infraestructure as a Code) de Terraform.

## Introducción

Toda la implementación la podemos encontrar en el repositorio de Gitlab llamado [infraestructura](#). En este repositorio encontramos este árbol de directorios:



Donde:

- **vars.tf**: son las variables de entrada para terraform.
  - **table\_name**: nombre de la tablade DynamoDB, si cambiamos este nombre deberemos de modificar el nombre de la tabla utilizada en el lambda que implementa el CRUD de layouts
  - **table\_billing\_mode**: método de pago para la lambda que implementa el CRUD de layouts
  - **enviroment**: versión del API desplegada
- **main.tf**: representa el script principal que aprovisiona todos los recursos de AWS para gestionar los layouts:
  - Tabla de DynamoDB
  - IAM policies
  - Lambda Functios
  - API Gateway
- **outputs.tf**: representa la salida tras el aprovisionamiento de Terraform. Sacamos la URL base que se utilizará para acceder al recurso de layout através de todos los métodos implementados por el lambda de CRUD. Un ejemplo será del tipo:
  - **findAll** (GET) <URL\_BASE>/layouts
  - **findById** (GET) <URL\_BASE>/layouts/{id}
  - **save** (PUT) <URL\_BASE>/layouts
  - **delete** (DELETE) <URL\_BASE>/layouts/{id}

**Un ejemplo podría ser:** recuperar todos los layouts de la tienda con código MANGO002

<https://qq5map9ju9.execute-api.us-east-2.amazonaws.com/v1/layouts/MANGO002>

## Configuración

Antes de ejecutar el proyecto de Terraform debemos de configura la conexión a nuestra infraestructura AWS correctamente. Para ello podemos editar el fichero `~/.aws/credentials` y creamos un nuevo perfil

```
[default]
aws_access_key_id = <ACCESS_KEY_DEFAULT_PROFILE>
aws_secret_access_key = <SECRET_KEY_DEFAULT_PROFILE>

[mango]
aws_access_key_id = <ACCESS_KEY_MANGO_PROFILE>
aws_secret_access_key = <SECRET_KEY_MANGO_PROFILE>
```

Una vez configurado ya podemos planificar nuestro proyecto de Terraform

## Despliegue

Para aprovisionar la infraestructura simplemente deberemos de:

- Instalar correctamente la el CLI de AWS y correctamente conectado a la cuenta de AWS correspondiente.
- Instalado el CLI de terraform correctamente en nuestro equipo.

### STEP 01:

Primero deberemos de inicializar el proyecto de Terraform para bajarse las dependencias necesarias para conectarnos a la infraestructura de AWS, ejecutando este comando:

```
terraform init
```

### STEP 02:

Deberemos de planificar nuestro proyecto, chequeando que el mismo es correcto ante de iniciar el aprovisionamiento, ejecutando este comando pasando el perfil a utilizar.

```
terraform plan -var 'aws_profile=mango' -var 'aws_region=eu-west-1'
```

Estas son las variables de terraform configurables:

variable	default	descripción
aws_profile	default	Perfil de conexión a AWS con sus credenciales
aws_region	us-east-2	Región de AWS donde aprovisionar infraestructura
table_name	Layout	Nombre de la tabla de DynamoDB. Cualquier cambio implica modificar el lambda CRUD
table_billing_name	PAY_PER_REQUEST	Controla como se cargarán las lecturas y escrituras en las tablas de DynamoDB
environment	api_v1	Representa la versión de las APIs utilizadas por el sistema

### STEP 03:

Si la planificación anterior es correcta podemos iniciar nuestro aprovisionamiento ejecutando este comando:

```
terraform apply -var 'aws_profile=mango' -var 'aws_region=eu-west-1'
```

Tras terminar obtendremos la URL base para conectarnos a nuestras API recién desplegada, como por ejemplo esta salida:

```
aws_apigatewayv2_stage.layout_api_stage: Destroying... [id=production]
aws_apigatewayv2_route.layout_put_api_route: Modifying... [id=n652cxi]
aws_apigatewayv2_route.layout_get_by_id_api_route: Modifying... [id=7lakoc9]
aws_apigatewayv2_route.layout_delete_api_route: Modifying... [id=dm6xrlt]
aws_apigatewayv2_route.layout_get_api_route: Modifying... [id=360m619]
aws_dynamodb_table.layout_dynamodb_table: Modifying... [id=Layout]
aws_apigatewayv2_stage.layout_api_stage: Destruction complete after 1s
aws_apigatewayv2_stage.layout_api_stage: Creating...
aws_apigatewayv2_route.layout_get_api_route: Modifications complete after 1s [id=360m619]
aws_apigatewayv2_route.layout_delete_api_route: Modifications complete after 1s [id=dm6xrlt]
aws_apigatewayv2_route.layout_get_by_id_api_route: Modifications complete after 1s [id=7lakoc9]
aws_apigatewayv2_route.layout_put_api_route: Modifications complete after 1s [id=n652cxi]
aws_dynamodb_table.layout_dynamodb_table: Modifications complete after 1s [id=Layout]
aws_apigatewayv2_stage.layout_api_stage: Creation complete after 1s [id=v1]

Apply complete! Resources: 1 added, 5 changed, 1 destroyed.

Outputs:
base_url = "https://qq5map9ju9.execute-api.us-east-2.amazonaws.com/v1"
```

#### STEP 04:

Después de aprovisionar la infraestructura, podemos modificar la misma en cualquier momento ejecutando esta combinación de comandos:

```
terraform plan -var "aws_profile=mango"
terraform apply
```

O bien podemos liberar toda la infraestructura aprovisionada, incluyendo la tabla de DynamoDB.

```
terraform destroy
```

⚠ Realizar esto con sumo cuidado si antes no hemos hecho un backup de los datos de DynamoDB