

# Prueba técnica NODRIZATech

---

Solución propuesta a la prueba técnica por **Toño Ramos**

## Instalación

---

Para facilitar el proceso de instalación se usa un archivo `Makefile`, para la instalación de la aplicación basta con ejecutar:

```
$ git clone git@github.com:rulzg/nodriza_planets.git
$ cd nodriza_planets
$ make build
```

El comando `make build` realiza varios procesos:

- Construir los contenedores `docker`
- Instalará el software necesario (`php`, `nginx`, `mysql`, etc..)
- Instalará la última versión de `Symfony` y el resto de componentes especificados en el archivo `composer.json`
- Ejecuta la migración que creará la bbdd y el schema necesario
- Ejecuta un servidor web que escucha en `http://localhost:8080`

## Requisitos

La aplicación está pensada para ejecutarse en un sistema `linux`, para que funcione será necesario tener instalado:

- `docker-compose`
- `make`

Para probar los endpoints también es recomendable tener instalado **Postman**.

## Recursos y estructura aplicación

---

La estructura de la aplicación es la siguiente:

```
.
├─ app                // Código fuente de la aplicación
├─ docker-compose.yml // Configuración de docker
├─ Makefile           // Definición comandos make
├─ nginx              // Configuración nginx
├─ php                // Configuración php
├─ Planets.postman_collection.json // Colección de peticiones para Postman
├─ README.md          // Este documento
└─ README.pdf         // Documentación en formato PDF
```

## Makefile

Para mayor comodidad durante el desarrollo se han definido una serie de comandos en el archivo `Makefile` para interactuar con **docker**, para ver un listado de las opciones disponibles ejecutar `make help`:

```
$ make help
```

```
-----
---
Están disponibles las siguientes opciones:
-----
---
```

<code>make help</code>	- Muestra esta ayuda
<code>make build</code>	- Construye los contenedores docker e instala la aplicación
<code>make run</code>	- Ejecuta la aplicación
<code>make stop</code>	- Detiene la aplicación
<code>make status</code>	- Muestra status de los contenedores
<code>make composer-install</code>	- Ejecuta ' <b>composer install</b> ' en la aplicación
<code>make composer-update</code>	- Ejecuta ' <b>composer update</b> ' en la aplicación
<code>make shell</code>	- Inicia una shell en el contenedor php

## Colección peticiones Postman

El archivo `Planets.postman_collection.json` contiene ejemplos de llamadas a los dos endpoints implementados en la aplicación listos para importar desde **Postman**.

## API

Descripción de los endpoints implementados:

# GET /api/planets/{id}

## Request

```
curl -i -X GET http://localhost:8080/api/planets/1 -H 'Content-Type: application/json'
```

## Response

```
HTTP/1.1 200 OK
Server: nginx/1.22.1
Content-Type: application/json
Transfer-Encoding: chunked
Connection: keep-alive
X-Powered-By: PHP/8.0.24
Cache-Control: no-cache, private
Date: Tue, 01 Nov 2022 19:30:34 GMT
X-Robots-Tag: noindex

{
  "name": "Tatooine",
  "rotation_period": "23",
  "orbital_period": "304",
  "diameter": "10465",
  "climate": "arid",
  "gravity": "1 standard",
  "terrain": "desert",
  "surface_water": "1",
  "population": "200000",
  "residents": [
    "https://swapi.dev/api/people/1/",
    "https://swapi.dev/api/people/2/",
    "https://swapi.dev/api/people/4/",
    "https://swapi.dev/api/people/6/",
    "https://swapi.dev/api/people/7/",
    "https://swapi.dev/api/people/8/",
    "https://swapi.dev/api/people/9/",
    "https://swapi.dev/api/people/11/",
    "https://swapi.dev/api/people/43/",
    "https://swapi.dev/api/people/62/"
  ],
  "films": [
    "https://swapi.dev/api/films/1/",
    "https://swapi.dev/api/films/3/",
    "https://swapi.dev/api/films/4/",
    "https://swapi.dev/api/films/5/",
    "https://swapi.dev/api/films/6/"
  ],
  "created": "2014-12-09T13:50:49.641000Z",
  "edited": "2014-12-20T20:58:18.411000Z",
  "url": "https://swapi.dev/api/planets/1/"
}
```

# POST /api/planet

## Request

```
curl -i -X POST http://localhost:8080/api/planet -H 'Content-Type: application/json' -d '{"id": 1, "name": "test", "rotation_period": 1, "orbital_period": 2, "diameter": 3}'
```

## Response OK

```
HTTP/1.1 201 Created
Server: nginx/1.22.1
Content-Type: application/json
Transfer-Encoding: chunked
Connection: keep-alive
X-Powered-By: PHP/8.0.24
Cache-Control: no-cache, private
Date: Tue, 01 Nov 2022 19:36:21 GMT
X-Robots-Tag: noindex
```

```
{
  "id": 1,
  "name": "test",
  "rotation_period": 1,
  "orbital_period": 2,
  "diameter": 3
}
```

## Response KO

```
HTTP/1.1 500 Internal Server Error
Server: nginx/1.22.1
Content-Type: application/json
Transfer-Encoding: chunked
Connection: keep-alive
X-Powered-By: PHP/8.0.24
Cache-Control: no-cache, private
Date: Tue, 01 Nov 2022 19:36:51 GMT
X-Robots-Tag: noindex
```

```
{"errors":["id: This value is already used."]}
```