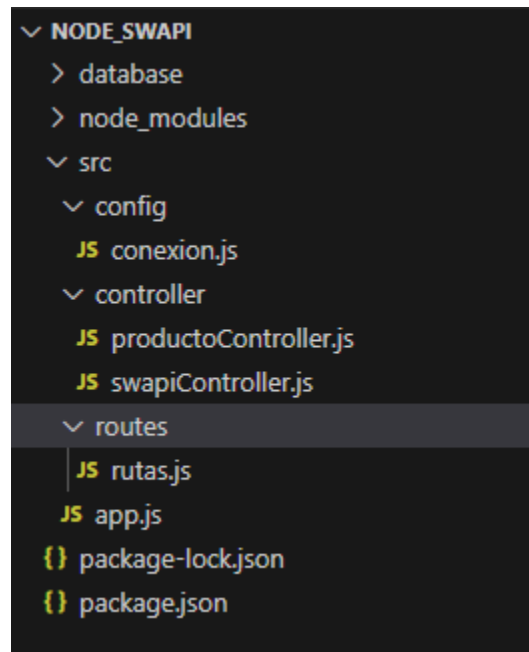


DOCUMENTACIÓN DE USO

APLICACIÓN NODE_SWAPI

DANIEL FERNANDO LADINO

COMPOSICIÓN DE LA APLICACIÓN



El api está conformado por los siguientes archivos y carpetas:

- Carpeta database: contiene el archivo db.sql que está compuesto por los scripts necesarios para crear una base de datos y la tabla producto (modelo de elección).
- Carpeta node_modules: contiene todas las dependencias requeridas en la aplicación.
- Carpeta src: contiene los archivos necesarios para el funcionamiento de la aplicación.
 - conexión.js: contiene la configuración necesaria para establecer la conexión con la base de datos mysql empresa_productos.
 - productoController: contiene las funcionalidades para obtener una conexión mysql y listar los registros de la tabla producto, obtener registros por id e insertar productos.
 - swapiController: contiene la lógica necesaria para listar la url expuesta: <https://swapi.py4e.com/api/>, también obtiene por id las personas, películas, naves, vehículos, especies y planetas, cada funcionalidad convierte los atributos del json obtenido a español y usa fetch api para la obtención de los datos.
 - rutas: contiene las rutas get y post para acceder a la api swapi u obtener/insertar la tabla producto mysql.
 - app.js: contiene la configuración inicial de la aplicación, la definición de la instancia de express con su configuración e inicio y la importación de las rutas.

- Package.json: contiene metadata de la aplicación, configuración de scripts y dependencias.

USO DE BASE DE DATOS EN LA APLICACIÓN

La aplicación utiliza el motor de base de datos mysql, utiliza la base de datos llamada empresa_productos y una tabla llamada producto con la siguiente estructura:

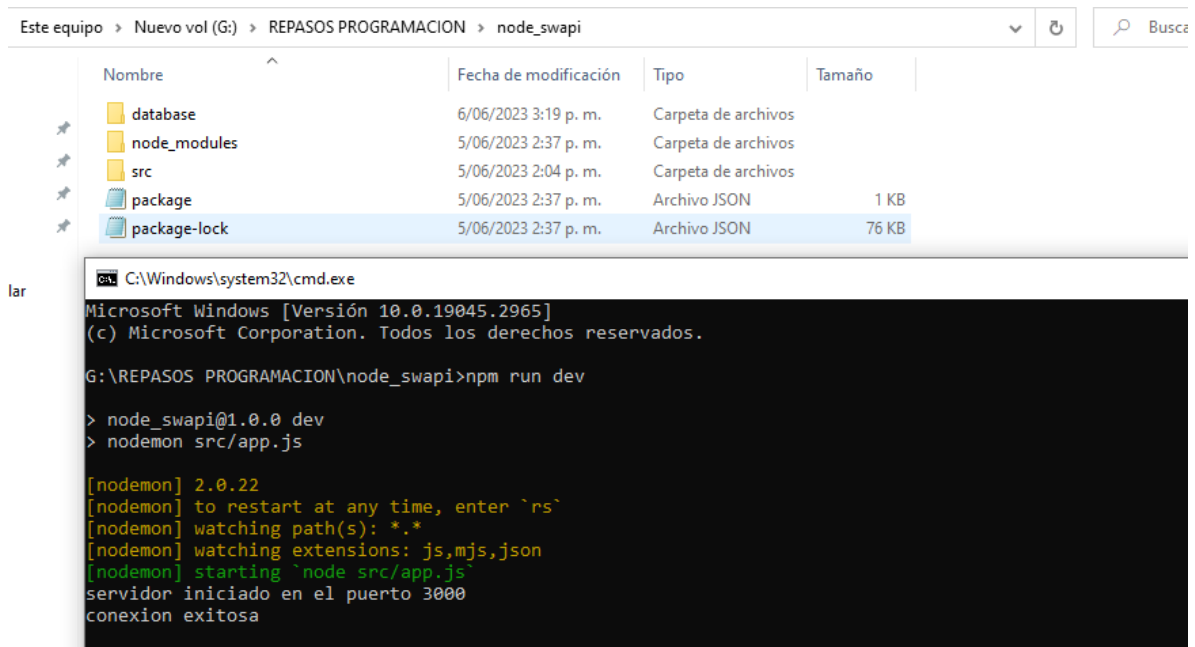
```
CREATE TABLE `empresa_productos`.`producto` (  
  `idproducto` INT NOT NULL AUTO_INCREMENT,  
  `nombre` VARCHAR(45) NOT NULL,  
  `descripcion` VARCHAR(500) NOT NULL,  
  `precio` DOUBLE NOT NULL,  
  `existencias` INT NOT NULL,  
  `foto` VARCHAR(2000) NULL,  
  `fecharegistro` DATE NULL,  
  PRIMARY KEY (`idproducto`));
```

idproducto	nombre	descripcion	precio	existencias	foto	fecharegistro
1	Morral	Moral totto	200000	24	NULL	2023-05-03
2	Dog Chow	Comida húmeda para perros	2900	10	NULL	2018-01-01
3	Baffle	Parlante para equipo	500000	7	NULL	2018-01-01

USO DE LA APLICACIÓN

Para la creación de la carpeta node_modules, iniciar una terminal de comandos dentro de la aplicación y ejecutar npm install.

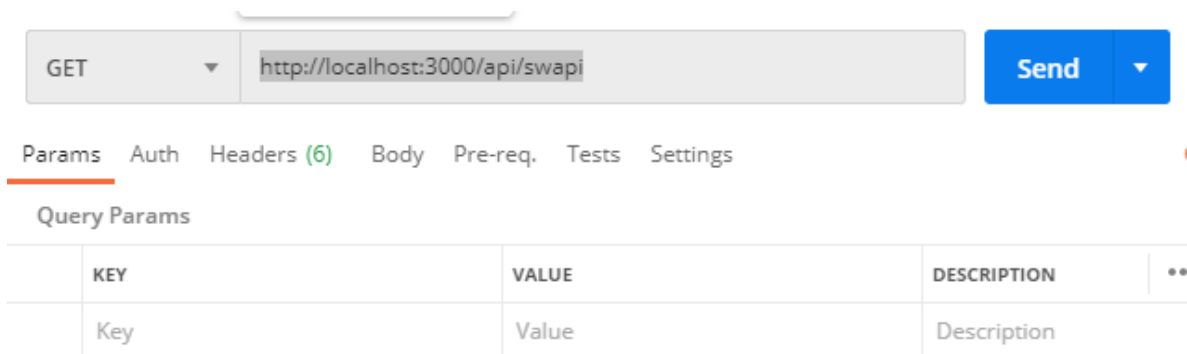
Iniciar una terminal de comandos y ubicarse dentro de la carpeta de la aplicación, ejecutar: npm run dev



Iniciar la aplicación Postman para el consumo de las solicitudes:

PETICIONES SWAPI

- Listado general del api SWAPI: realizar la petición get de la url: <http://localhost:3000/api/swapi> y enviar:



Body ▾ 200 OK 442 ms 540 B

Pretty Raw Preview Visualize JSON ▾ 

```
1 {  
2   "gente": "https://swapi.py4e.com/api/people/",  
3   "planetas": "https://swapi.py4e.com/api/planets/",  
4   "películas": "https://swapi.py4e.com/api/films/",  
5   "especies": "https://swapi.py4e.com/api/species/",  
6   "vehículos": "https://swapi.py4e.com/api/vehicles/",  
7   "naves estelares": "https://swapi.py4e.com/api/starships/"  
8 }
```

- Búsqueda de personas por id: realizar la petición get de ejemplo: <http://localhost:3000/api/swapi/personas/1> y enviar:

GET ▾ Send ▾

Params Auth Headers (6) Body Pre-req. Tests Settings

Query Params

	KEY	VALUE	DESCRIPTION	
	Key	Value	Description	•

Body ▾ 200 OK 466 ms 1.03 KB Sav

Pretty Raw Preview Visualize JSON ↗

```
1 {
2   "nombre": "Luke Skywalker",
3   "altura": "172",
4   "masa": "77",
5   "color de cabello": "blond",
6   "color de piel": "fair",
7   "color de ojos": "blue",
8   "año de nacimiento": "1988Y",
9   "género": "male",
10  "mundo natal": "https://swapi.py4e.com/api/planetas/1/",
11  "películas": [
12    "https://swapi.py4e.com/api/films/1/",
13    "https://swapi.py4e.com/api/films/2/",
14    "https://swapi.py4e.com/api/films/3/",
15    "https://swapi.py4e.com/api/films/6/",
16    "https://swapi.py4e.com/api/films/7/"
17  ],
18  "especies": [
19    "https://swapi.py4e.com/api/species/1/"
20  ],
21  "vehículos": [
22    "https://swapi.py4e.com/api/vehicles/14/",
23    "https://swapi.py4e.com/api/vehicles/30/"
24  ],
25  "naves estelares": [
26    "https://swapi.py4e.com/api/starships/12/",
27    "https://swapi.py4e.com/api/starships/22/"
28  ],
29  "creado": "2014-12-09T13:50:51.644000Z",
30  "editado": "2014-12-20T21:17:56.891000Z",
31  "url": "https://swapi.py4e.com/api/gente/1/"
32 }
```

- Búsqueda de películas por id: realizar la petición get de ejemplo: <http://localhost:3000/api/swapi/peliculas/3> y enviar:

GET

http://localhost:3000/api/swapi/peliculas/3

Send

Params

Auth

Headers (6)

Body

Pre-req.

Tests

Settings

Query Params

KEY	VALUE	DESCRIPTION	
Key	Value	Description	

Raw

Parsed

```

{
  "título": "Return of the Jedi",
  "número de episodio": 6,
  "rastreo de apertura": "Luke Skywalker has returned to\r\nhis home planet of Tatooine in\r\nan attempt to rescue his\r\nfriend Han Solo from the\r\nclutches of the vile gangster\r\nJabba the Hutt.\r\nLittle does Luke know that the\r\nGALACTIC EMPIRE has secretly\r\nbegun construction on a new\r\narmored space station even\r\nmore powerful than the first\r\n dreaded Death Star.\r\nWhen completed, this ultimate\r\nweapon will spell certain doom\r\nfor the small band of rebels\r\nstruggling to restore freedom\r\ninto the galaxy...",
  "director": "Richard Marquand",
  "productor": "Howard G. Kazanjian, George Lucas, Rick McCallum",
  "fecha de realización": "1983-05-25",
  "protagonistas": [ ... ], // 20 items
  "planetas": [ ... ], // 5 items
  "naves estelares": [ ... ], // 12 items
  "vehículos": [ ... ], // 8 items
  "especies": [
    "https://swapi.py4e.com/api/species/1/",
    "https://swapi.py4e.com/api/species/2/",
    "https://swapi.py4e.com/api/species/3/",
    "https://swapi.py4e.com/api/species/5/",
    "https://swapi.py4e.com/api/species/6/",
    "https://swapi.py4e.com/api/species/8/",
    "https://swapi.py4e.com/api/species/9/",
    "https://swapi.py4e.com/api/species/10/",
    "https://swapi.py4e.com/api/species/15/"
  ],
  "creado": "2014-12-18T10:39:33.255000Z",
  "editado": "2014-12-20T09:48:37.462000Z",
  "url": "https://swapi.py4e.com/api/peliculas/3/"
}

```

- Búsqueda de naves por id: realizar la petición get de ejemplo: <http://localhost:3000/api/swapi/naves/3> y enviar:

GET

http://localhost:3000/api/swapi/naves/3

Send

Params

Auth

Headers (6)

Body

Pre-req.

Tests

Settings

Query Params

KEY	VALUE	DESCRIPTION	
Key	Value	Description	

Body ▾

200 OK 485 ms 882 B

Pretty

Raw

Preview

Visualize

JSON ▾



```
1 {
2   "nombre": "Star Destroyer",
3   "modelo": "Imperial I-class Star Destroyer",
4   "fabricante": "Kuat Drive Yards",
5   "costo en créditos": "150000000",
6   "longitud": "1,600",
7   "velocidad atmosférica máxima": "975",
8   "tripulación": "47,060",
9   "pasajeros": "n/a",
10  "capacidad de carga": "36000000",
11  "consumibles": "2 years",
12  "calificación del hiperimpulsor": "2.0",
13  "MGLT": "60",
14  "clase de nave": "Star Destroyer",
15  "pilotos": [],
16  "películas": [
17    "https://swapi.py4e.com/api/films/1/",
18    "https://swapi.py4e.com/api/films/2/",
19    "https://swapi.py4e.com/api/films/3/"
20  ],
21  "creado": "2014-12-10T15:08:19.848000Z",
22  "editado": "2014-12-20T21:23:49.870000Z",
23  "url": "https://swapi.py4e.com/api/starships/3/"
24 }
```

- Búsqueda de vehículos por id: realizar la petición get de ejemplo: <http://localhost:3000/api/swapi/vehiculos/3> y enviar:

GET ▾

<http://localhost:3000/api/swapi/vehiculos/3>


Send ▾

Params Auth Headers (6) Body Pre-req. Tests Settings

Query Params

	KEY	VALUE	DESCRIPTION	••
	Key	Value	Description	

Body ▾ 200 OK 386 ms

Pretty Raw Preview Visualize JSON ▾ 

```
1 {
2   "nombre": "Star Destroyer",
3   "modelo": "Imperial I-class Star Destroyer",
4   "fabricante": "Kuat Drive Yards",
5   "costo en créditos": "150000000",
6   "longitud": "1,600",
7   "velocidad atmosférica máxima": "975",
8   "personal": "47,060",
9   "pasajeros": "n/a",
10  "capacidad de carga": "36000000",
11  "consumibles": "2 years",
12  "calificación del hiperimpulsor": "2.0",
13  "MGLT": "60",
14  "clase de nave": "Star Destroyer",
15  "pilotos": [],
16  "películas": [
17    "https://swapi.py4e.com/api/films/1/",
18    "https://swapi.py4e.com/api/films/2/",
19    "https://swapi.py4e.com/api/films/3/"
20  ],
21  "creado": "2014-12-10T15:08:19.848000Z",
22  "editado": "2014-12-20T21:23:49.870000Z",
23  "url": "https://swapi.py4e.com/api/starships/3/"
24 }
```

- Búsqueda de especies por id: realizar la petición get de ejemplo:
<http://localhost:3000/api/swapi/especies/3> y enviar:

GET ▾ Send ▾

Params Auth Headers (6) Body Pre-req. Tests Settings

Query Params

	KEY	VALUE	DESCRIPTION	•
	Key	Value	Description	

Body ▾

200 OK 668 r

Pretty

Raw

Preview

Visualize

JSON ▾



```
1 {
2   "nombre": "Wookiee",
3   "clasificación": "mammal",
4   "designación": "sentient",
5   "altura promedio": "210",
6   "colores de piel": "gray",
7   "colores de cabello": "black, brown",
8   "colores de ojos": "blue, green, yellow, brown, golden, red",
9   "promedio de vida": "400",
10  "mundo natal": "https://swapi.py4e.com/api/planets/14/",
11  "idioma": "Shyriiwook",
12  "personas": [
13    "https://swapi.py4e.com/api/people/13/",
14    "https://swapi.py4e.com/api/people/80/"
15  ],
16  "películas": [
17    "https://swapi.py4e.com/api/films/1/",
18    "https://swapi.py4e.com/api/films/2/",
19    "https://swapi.py4e.com/api/films/3/",
20    "https://swapi.py4e.com/api/films/6/",
21    "https://swapi.py4e.com/api/films/7/"
22  ],
23  "creado": "2014-12-10T16:44:31.486000Z",
24  "editado": "2014-12-20T21:36:42.142000Z",
25  "url": "https://swapi.py4e.com/api/species/3/"
26 }
```

- Búsqueda de planetas por id: realizar la petición get de ejemplo: <http://localhost:3000/api/swapi/planetas/3> y enviar:

GET ▾

<http://localhost:3000/api/swapi/planetas/3>

Send ▾

Params Auth Headers (6) Body Pre-req. Tests Settings

Query Params

	KEY	VALUE	DESCRIPTION	..
	Key	Value	Description	

Body ▾

200 OK

Pretty

Raw

Preview

Visualize

JSON ▾



```
1 {
2   "nombre": "Yavin IV",
3   "periodo de rotación": "24",
4   "periodo de órbita": "4818",
5   "diámetro": "10200",
6   "clima": "temperate, tropical",
7   "gravedad": "1 standard",
8   "terreno": "jungle, rainforests",
9   "superficie del agua": "8",
10  "población": "1000",
11  "residentes": [],
12  "películas": [
13    "https://swapi.py4e.com/api/films/1/"
14  ],
15  "creado": "2014-12-10T11:37:19.144000Z",
16  "editado": "2014-12-20T20:58:18.421000Z",
17  "url": "https://swapi.py4e.com/api/planets/3/"
18 }
```

PETICIONES A BASE DE DATOS

- Búsqueda de todos los productos: realizar la petición get: <http://localhost:3000/api/productos> y enviar:

Untitled Request

BUILD

GET ▾

http://localhost:3000/api/productos

Send ▾

Body ▾

200 OK 8 ms 1.89 KB Save

Pretty

Raw

Preview

Visualize

JSON ▾



```
1  [
2    {
3      "idproducto": 1,
4      "nombre": "Morral",
5      "descripcion": "Moral totto",
6      "precio": 200000,
7      "existencias": 24,
8      "foto": null,
9      "fecharegistro": "2023-05-03T05:00:00.000Z"
10   },
11   {
12     "idproducto": 2,
13     "nombre": "Dog Chow",
14     "descripcion": "Comida húmeda para perros",
15     "precio": 2900,
16     "existencias": 10,
17     "foto": null,
18     "fecharegistro": "2018-01-01T05:00:00.000Z"
19   },
20   {
21     "idproducto": 3,
22     "nombre": "Bafle",
23     "descripcion": "Parlante para equipo",
24     "precio": 500000,
25     "existencias": 7,
26     "foto": null,
27     "fecharegistro": "2018-01-01T05:00:00.000Z"
28   },
29   {
30     "idproducto": 4,
31     "nombre": "Mug",
32     "descripcion": "Pocillo con icono empresarial",
33     "precio": 10000,
34     "existencias": 25,
```

- Búsqueda de productos por id: realizar la petición get de ejemplo: <http://localhost:3000/api/swapi/planetas/3> y enviar:

Untitled Request BUILD

GET ▼ <http://localhost:3000/api/productos/17> Send ▼

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION	**
Key	Value	Description	

Body Cookies Headers (7) Test Results ⊕ Status: 200 OK Time: 7 ms Size: 580 B Save Response ▼

Pretty Raw Preview Visualize JSON ↕

```
1 {
2   "idproducto": 17,
3   "nombre": "pantalón",
4   "descripcion": "Pantalón beige talla 31",
5   "precio": 60000,
6   "existencias": 55,
7   "foto": "https://cdn.shopify.com/s/files/1/0311/9398/9260/products/true_cargo_multipocket_pants_black_pantalones_sudaderas_1_776423a6-c4ba-43aa-be88-cf671e38a3da_720x720.jpg?v=1669821939",
8   "fecharegistro": "2023-05-29T05:00:00.000Z"
9 }
10
11
```

- Inserción de productos: realizar la petición post: <http://localhost:3000/api/productos> en el body de la petición, opción raw/JSON agregar un nuevo registro en formato json tal como se muestra de ejemplo y enviar

POST ▼ <http://localhost:3000/api/productos> Send ▼ Save ▼

Params Authorization Headers (8) Body ● Pre-request Script Tests Settings Cookies Code

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL JSON ▼ Beautify

```
1 {
2   "nombre": "PC",
3   "descripcion": "Computador portátil",
4   "precio": "6000000",
5   "existencias": "5",
6   "foto": "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcR-5TZsP6oC9ExyLmIpev1zY4C0cFeHc9xYUgqPjVC-6wyfYkPkb0j5cbIK54_tBzuCwI0&usqp=CAU",
7   "fecharegistro": "2022-10-04"
8 }
```

Body Cookies Headers (7) Test Results ⊕ Status: 200 OK Time: 33 ms Size: 265 B Save Response ▼

Pretty Raw Preview Visualize JSON ↕

```
1 {
2   "status": "producto agregado"
3 }
```

GET

http://localhost:3000/api/productos/21

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Query Params

KEY	VALUE	DESCRIPTION	...
-----	-------	-------------	-----

Body

Cookies

Headers (7)

Test Results

Status: 200 OK

Time: 6 ms

Size: 525 B

Saved

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

7

8

9

10

11

{

"producto": 21,

"nombre": "PC",

"descripcion": "Computador portátil",

"precio": 6000000,

"existencias": 5,

"foto": "https://encrypted-tbn0.gstatic.com/images?q=tbn:AND9GcR-5TZsPGoC9ExyLmNpevlzY4C0cFeHc9xYUgqPjVC-6wyfYkPkb0j5cbIK54_tBzuCWI0&usqp=CAU",

"fecha_registro": "2022-10-04T05:00:00.000Z"

}