

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FUNDAMENTOS DE BASES DE DATOS 2026-1

PRÁCTICA 08

Equipo Excelovers • Octubre de 2025

Torneo de Pokémon Go

JDBC & Psycopg



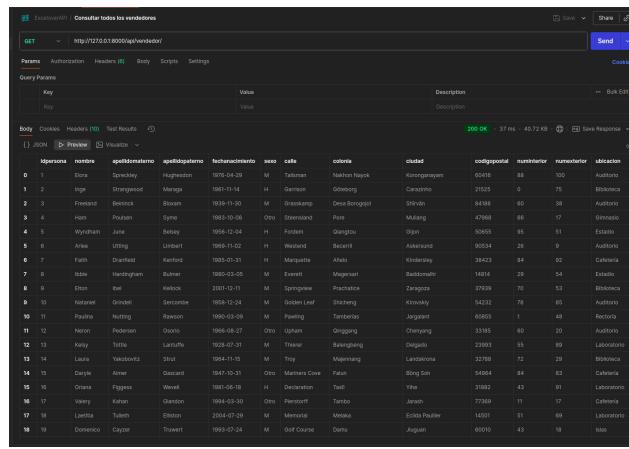
Para esta práctica decidimos usar *Psycopg2* y *Django*. En el proyecto hay instrucciones de cómo instalar y correr el proyecto.

API'S EN POSTMAN

Vendedor

GET

Probamos la API para la tabla **Vendedor** consultando a todos los alimentos con GET. Recibimos el código 200, lo que significa que la solicitud se procesó correctamente y el servidor devolvió una respuesta exitosa. En SQL haríamos `SELECT * FROM vendedor;`



idpersona	nombre	apellidomaterno	apellidopaterno	fechanacimiento	sexo	calle	colonia	ciudad	codigopostal	numinterior	numexterior	ubicacion
0	Ella	Spreckley	Hughesdon	1976-04-29	M	Tallman	Nahon Nayak	Korsenayang	60416	88	100	Australia
1	Igor	Strongwood	Marge	1965-11-14	M	Gernase	Goneberg	Cansene	21025	0	75	Biblioteca
2	Freeland	Beelock	Bosan	1939-11-00	M	Grasskamp	Desa Borongjol	Shihdin	84988	80	88	Australia
3	Horn	Poulton	Syme	1983-10-08	Otro	Steensland	Pore	Mulang	47966	86	17	Germany
4	Wynghan	June	Belsay	1956-12-04	H	Forsen	Qiangtou	Qijon	92665	95	51	Estado
5	Arina	Uttung	Urbelt	1989-11-02	H	Woulard	Bacerril	Aksarund	95534	28	9	Australia
6	Fath	Drafield	Kentford	1985-01-31	H	Marquette	Afalo	Kinshale	35423	84	92	Cafeleira
7	Isabe	Hastingsham	Bulmer	1980-03-05	M	Everett	Magerant	Baddsmith	14014	29	54	Estado
8	Elton	Ibel	Kelock	2001-12-11	M	Springewe	Prachette	Zhangera	27908	70	53	Biblioteca
9	Nazare	Grindell	Seconne	1958-12-24	M	Gosen Leaf	Shicheng	Kirovsky	54222	78	85	Australia
10	Paulha	Nutting	Rawson	1990-03-09	M	Pawling	Tombolas	Jargant	65865	1	48	Rectoria
11	Neron	Pedersen	Osorio	1986-08-27	Otro	Uptan	Genggang	Chengyang	53195	80	20	Australia
12	Kelly	Totte	Lamotte	1928-07-31	M	Thivier	Baekgang	Delgado	23993	55	89	Laboratorio
13	Laura	Yakovlevitz	Stuf	1984-11-15	M	Troy	Majengang	Lantidena	32780	72	29	Biblioteca
14	Danyle	Armer	Gascard	1947-10-23	Otro	Manners Cove	Falun	Bong San	54984	84	83	Cafeleira
15	Orson	Figgere	Wewel	1981-06-18	H	Decoratton	Tell	Yihe	31982	43	91	Laboratorio
16	Vanry	Kahan	Gaston	1984-03-20	Otro	Pienroff	Tambo	Jarech	77360	11	17	Cafeleira
17	Leontia	Tullem	Elston	2004-07-29	M	Memorial	Meraka	Erdos Poulder	14501	51	69	Laboratorio
18	Domenico	Cayzer	Trovett	1993-07-24	M	Golf Course	Darus	Jugian	65010	43	19	Isas

Figura 1: Operación GET para obtener a todos los vendedores

POST

Recibimos el código 201, lo que significa que el servidor creó de forma exitosa un nuevo recurso.

En SQL:

```
INSERT INTO Vendedor (
    idpersona, nombre, apellidomaterno, apellidopaterno, fechanacimiento,
    sexo, calle, colonia, ciudad, codigopostal, numinterior, numexterior, ubicacion
) VALUES (
    151, 'Laura', 'Gomez', 'Perez', '1995-06-14',
    'M', 'Av. Juarez', 'Norte', 'CDMX', 11000, 5, 33, 'Cafeteria'
);
```

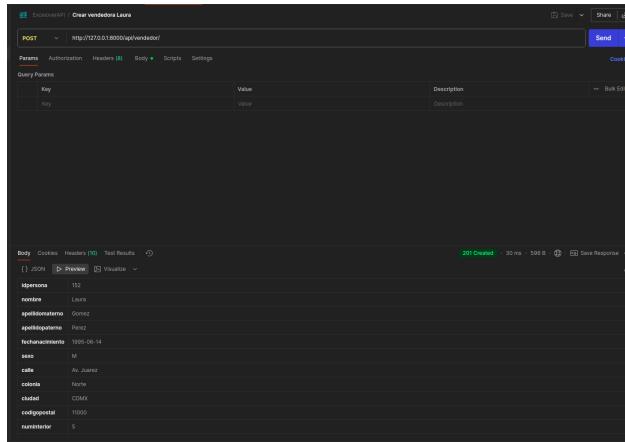


Figura 2: Operación POST para crear a la vendedora Laura

GET

Recibimos el código 200, lo que significa que la solicitud se procesó correctamente y el servidor devolvió una respuesta exitosa.

En SQL: `SELECT * FROM Vendedor WHERE idpersona = 151;`

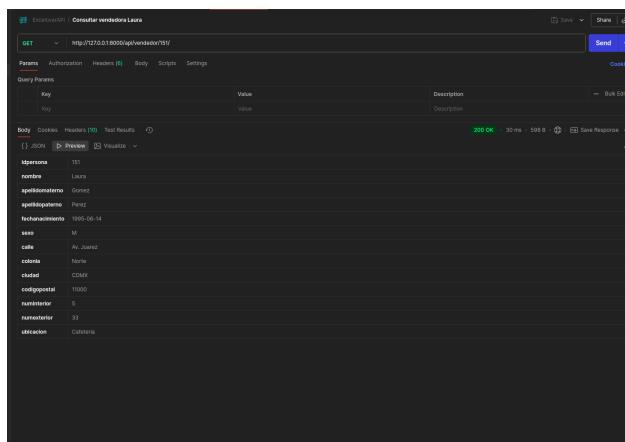


Figura 3: Operación GET para obtener a la vendedora Laura

PUT

Recibimos el código 200, lo que significa que la solicitud se procesó correctamente y el servidor devolvió una respuesta exitosa.

En SQL: `UPDATE Vendedor SET ubicacion = 'Biblioteca' WHERE idpersona = 151;`

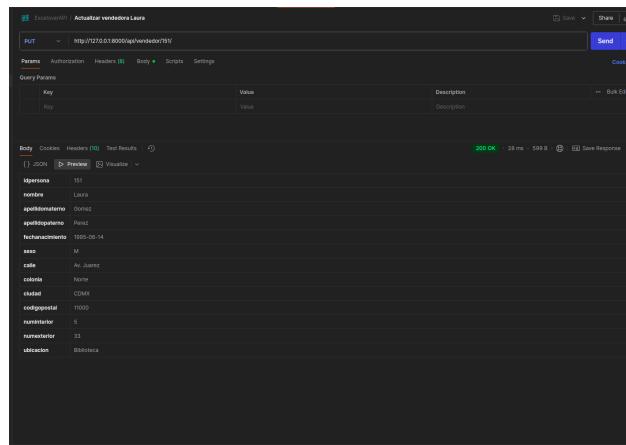


Figura 4: Operación PUT para actualizar a la vendedora Laura

GET Recibimos el código 200, lo que significa que la solicitud se procesó correctamente y el servidor devolvió una respuesta exitosa.

En SQL: `SELECT * FROM Vendedor WHERE idpersona = 151;`

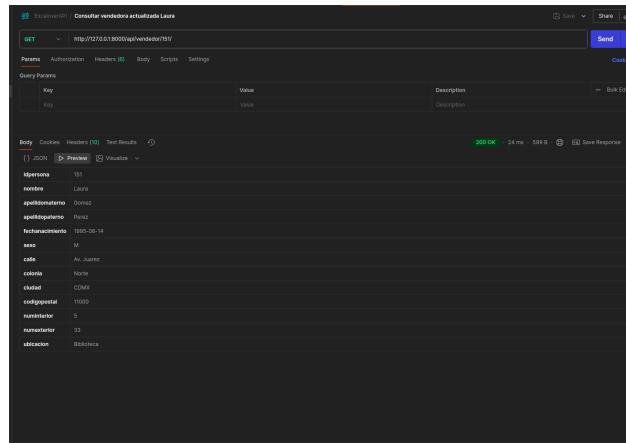


Figura 5: Operación GET para consultar a la vendedora Laura tras actualización

DELETE

Recibimos el código 204, lo que significa que el servidor borró de forma exitosa.

En SQL: `DELETE FROM Vendedor WHERE idpersona = 151;`

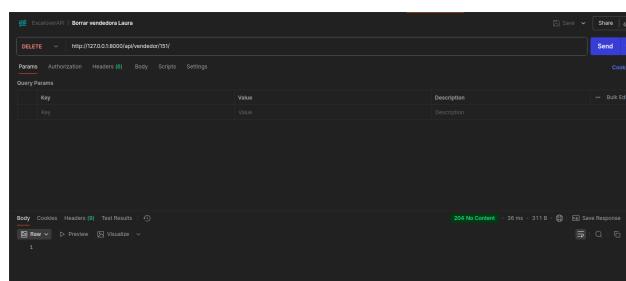


Figura 6: Operación DELETE para borrar a la vendedora Laura

GET

Recibimos el código 404, lo que significa que el servidor no encontró el recurso, tal como esperábamos.

En SQL: `SELECT * FROM Vendedor WHERE idpersona = 151;`

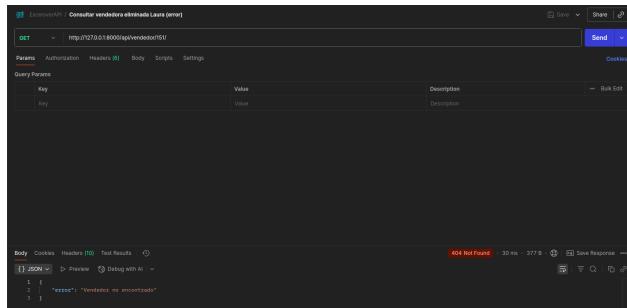


Figura 7: Operación GET para consultar a la vendedora Laura después de borrarse

Alimento

GET

Recibimos el código 200, lo que significa que la solicitud se procesó correctamente y el servidor devolvió una respuesta exitosa.

En SQL: `SELECT * FROM Alimento;`

Consultar todos los alimentos					
GET http://127.0.0.1:8000/api/alimento/					
Params Authorization Headers Body Scripts Settings Cookies					
Query Params					
Key	Value	Description			
Key	Value	Description			
Body	Cookies	Headers (10)	Test Results	404 OK	34 ms - 18.29 kB
(JSON) { "idalimento": 1, "fechadecaducidad": "2023-12-20", "nombre": "Mazapan reja", "tipo": "Dulce", "precio": 57.0, "idpersona": 122 }, { "idalimento": 2, "fechadecaducidad": "2027-02-25", "nombre": "Helado de vainilla", "tipo": "Postre", "precio": 222.00, "idpersona": 13 }, { "idalimento": 3, "fechadecaducidad": "2028-01-11", "nombre": "Galletas de avena", "tipo": "Embudo", "precio": 276.64, "idpersona": 145 }, { "idalimento": 4, "fechadecaducidad": "2025-11-09", "nombre": "Café molido", "tipo": "Sopa", "precio": 89.98, "idpersona": 109 }, { "idalimento": 5, "fechadecaducidad": "2026-04-21", "nombre": "Tomate", "tipo": "Verdura", "precio": 308.73, "idpersona": 56 }, { "idalimento": 6, "fechadecaducidad": "2029-04-06", "nombre": "Pasta espagueti", "tipo": "Grano", "precio": 143.76, "idpersona": 37 }, { "idalimento": 7, "fechadecaducidad": "2027-11-30", "nombre": "Tomate", "tipo": "Fruta", "precio": 476.21, "idpersona": 61 }, { "idalimento": 8, "fechadecaducidad": "2027-10-18", "nombre": "Chocolate en barra", "tipo": "Fruta", "precio": 481.5, "idpersona": 82 }, { "idalimento": 9, "fechadecaducidad": "2029-01-08", "nombre": "Plátano", "tipo": "Ensalada", "precio": 390.89, "idpersona": 101 }, { "idalimento": 10, "fechadecaducidad": "2027-01-15", "nombre": "Taco al pastor", "tipo": "Panadería", "precio": 147.9, "idpersona": 134 }, { "idalimento": 11, "fechadecaducidad": "2027-09-28", "nombre": "Helado de vainilla", "tipo": "Dulce", "precio": 83.09, "idpersona": 65 }, { "idalimento": 12, "fechadecaducidad": "2027-02-30", "nombre": "Sopa instantánea", "tipo": "Condimento", "precio": 15.429, "idpersona": 98 }, { "idalimento": 13, "fechadecaducidad": "2027-01-31", "nombre": "Salsa de tomate", "tipo": "Panadería", "precio": 235.92, "idpersona": 101 }, { "idalimento": 14, "fechadecaducidad": "2029-09-14", "nombre": "Emparedado", "tipo": "Lácteos", "precio": 278.84, "idpersona": 26 }, { "idalimento": 15, "fechadecaducidad": "2027-06-05", "nombre": "Yogur natural", "tipo": "Cereal", "precio": 413.13, "idpersona": 3 }, { "idalimento": 16, "fechadecaducidad": "2027-09-24", "nombre": "Hamburguesa", "tipo": "Fruta", "precio": 129.34, "idpersona": 140 }, { "idalimento": 17, "fechadecaducidad": "2025-11-02", "nombre": "Sopa instantánea", "tipo": "Condimento", "precio": 64.34, "idpersona": 89 }, { "idalimento": 18, "fechadecaducidad": "2029-01-09", "nombre": "Atún en lata", "tipo": "Verdura", "precio": 360.77, "idpersona": 143 }, { "idalimento": 19, "fechadecaducidad": "2029-12-24", "nombre": "Copa de Nutella", "tipo": "Azafrán", "precio": 127.15, "idpersona": 16 }					

Figura 8: Operación GET para obtener todos los alimentos

POST

Recibimos el código 201, lo que significa que el servidor creó de forma exitosa un nuevo recurso.

En SQL:

```
INSERT INTO Alimento (
    idalimento, fechadecaducidad, nombre, tipo, precio, idpersona
) VALUES (
    151, '2025-11-30', 'Pan de muerto', 'Pan', 55.5, 122
);
```

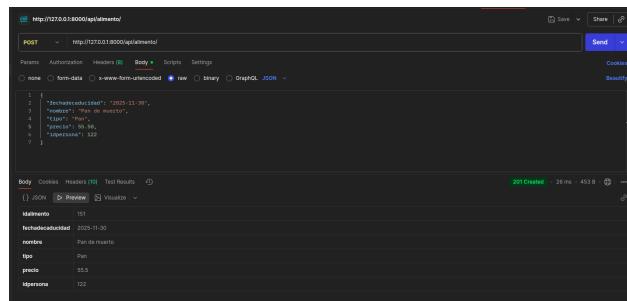


Figura 9: Operación POST para crear el alimento Pan de muerto

GET

Recibimos el código 200, lo que significa que la solicitud se procesó correctamente y el servidor devolvió una respuesta exitosa.

En SQL: `SELECT * FROM Alimento WHERE idalimento = 151;`

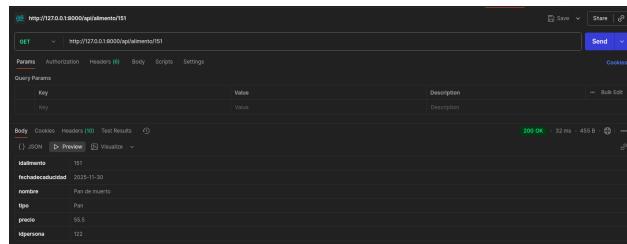


Figura 10: Operación GET para obtener el alimento Pan de muerto

PUT

Recibimos el código 200, lo que significa que la solicitud se procesó correctamente y el servidor devolvió una respuesta exitosa.

En SQL: `UPDATE Alimento SET precio = '30.5' WHERE idalimento = 151;`

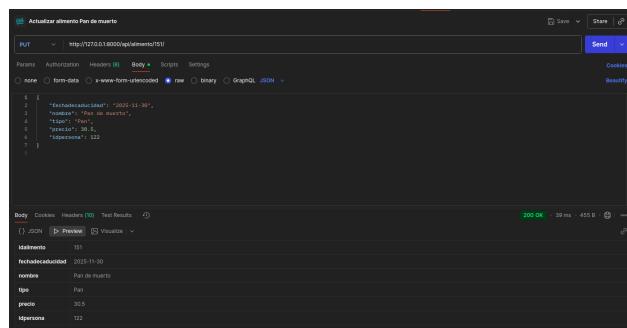


Figura 11: Operación PUT para actualizar el alimento Pan de muerto

GET

En SQL: `SELECT * FROM Alimento WHERE idalimento = 151;`

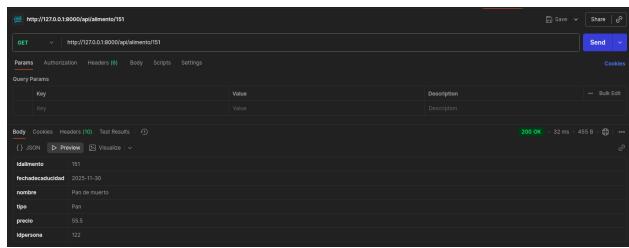


Figura 12: Operación GET para obtener el alimento Pan de muerto

DELETE

Recibimos el código 204, lo que significa que el servidor borró de forma exitosa.

En SQL: DELETE FROM Alimento WHERE idalimento = 151;

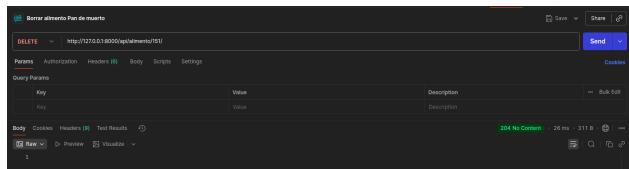


Figura 13: Operación DELETE para borrar el alimento Pan de muerto

GET

Recibimos el código 404, lo que significa que el servidor no encontró el recurso, tal como esperábamos.

En SQL: SELECT * FROM Alimento WHERE idalimento = 151;

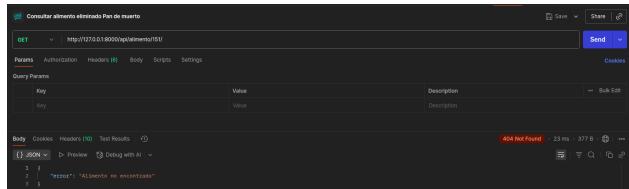


Figura 14: Operación GET para consultar el alimento eliminado Pan de muerto