



# DESARROLLO DE APLICACIONES WEB

ING. GEOVANNY BRAVO, MG.

## ACTIVIDAD PRÁCTICA # 1

UNIDAD 2: PROGRAMACIÓN WEB EN EL SERVIDOR

ANTONIO JOSÉ VILLÓN YAGUAL

14 DE DICIEMBRE DE 2024

## REALIZAR UNA API REST

### 1. LA API SERÁ ESTRUCTURADA EN PHP (CODEIGNITER)

#### Configuración de CodeIgniter

Configuración de la Base de Datos (app/Config/Database.php):

Se modifica las credenciales de PostgreSQL:

```
27 public $default = [  
28     'DSN' => '',  
29     'hostname' => 'localhost',  
30     'username' => 'postgres', // Usuario de PostgreSQL  
31     'password' => 'postgre', // Contraseña de PostgreSQL  
32     'database' => 'tienda',  
33     'DBDriver' => 'Postgre',  
34     'DBPrefix' => '',  
35     'pConnect' => false,  
36     'DBDebug' => (ENVIRONMENT !== 'production'),  
37     'cacheOn' => false,  
38     'cacheDir' => '',  
39     'charset' => 'utf8',  
40     'DBCollat' => 'utf8_general_ci',  
41     'swapPre' => '',  
42     'encrypt' => false,  
43     'compress' => false,  
44     'strictOn' => false,  
45     'failover' => [],  
46     'port' => 5432  
47 ];  
48
```

#### Creación del Controlador

se crea ApiController.php en app/Controllers.

```
app > Controllers > ApiController.php  
1  <?php  
2  
3  namespace App\Controllers;  
4  use CodeIgniter\RESTful\ResourceController;  
5  use App\Models\ClienteModel;  
6  use App\Models\PedidoModel;  
7  
8  class ApiController extends ResourceController  
9  {  
10     // Obtener todos los clientes  
11     public function getClientes()  
12     {  
13         $model = new ClienteModel();  
14         $clientes = $model->findAll();  
15         return $this->respond($clientes);  
16     }  
17  
18     // Obtener un cliente por ID  
19     public function getCliente($id = null)  
20     {  
21         $model = new ClienteModel();  
22         $cliente = $model->find($id);  
23     }  
24 }
```

```

23  ✓      if ($cliente) {
24          return $this->respond($cliente);
25  ✓      } else {
26          return $this->failNotFound('Cliente no encontrado');
27      }
28  }
29
30  // Obtener los pedidos de un cliente
31  public function getPedidos($cliente_id)
32  ✓  {
33      $model = new PedidoModel();
34      $pedidos = $model->getPedidosByCliente($cliente_id);
35      return $this->respond($pedidos);
36  }
37  }
38

```

## Crear Modelos para Clientes y Pedidos

Se crea los modelos ClienteModel.php y PedidoModel.php en app/Models.

```

app > Models > 🐞 ClienteModel.php
1  <?php
2
3  namespace App\Models;
4  use CodeIgniter\Model;
5
6  class ClienteModel extends Model
7  {
8      protected $table = 'clientes';
9      protected $primaryKey = 'id';
10     protected $allowedFields = ['nombre', 'correo', 'direccion'];
11 }
12

```

```

app > Models > 🐞 PedidoModel.php
1  <?php
2
3  namespace App\Models;
4  use CodeIgniter\Model;
5
6  class PedidoModel extends Model
7  {
8      protected $table = 'pedidos';
9      protected $primaryKey = 'id';
10     protected $allowedFields = ['cliente_id', 'producto', 'cantidad', 'fecha'];
11
12     public function getPedidosByCliente($cliente_id)
13     {
14         return $this->where('cliente_id', $cliente_id)->findAll();
15     }
16 }
17

```

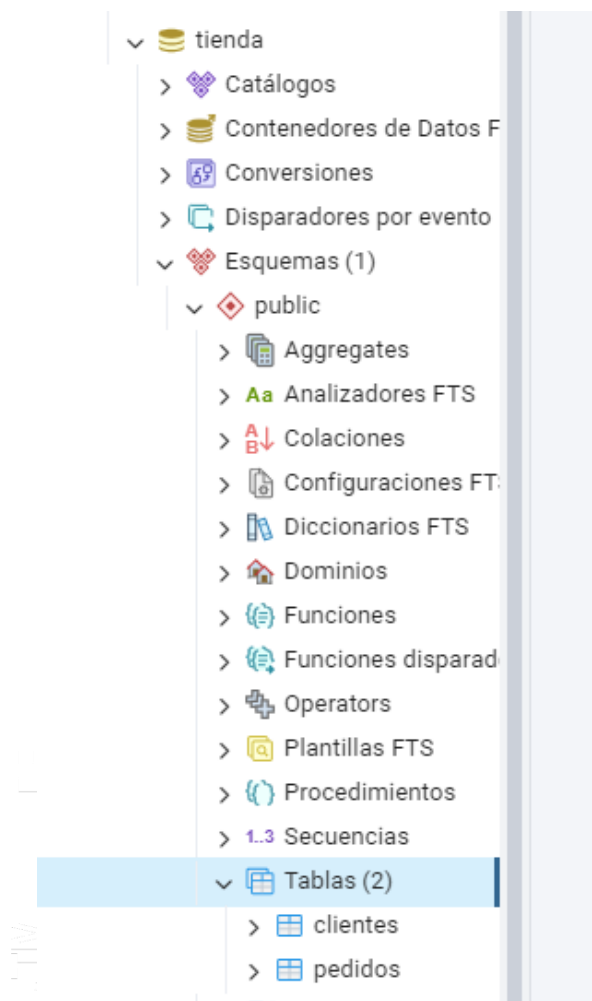
## Configurar las Rutas

Se define las rutas para la API en app/Config/Routes.php:

```
app > Config > Routes.php
30 // We get a performance increase by specifying the default
31 // route since we don't have to scan directories.
32
33 $routes->get('clientes', 'ApiController::getClientes');
34 $routes->get('cliente/(:num)', 'ApiController::getClient/$1');
35 $routes->get('pedidos/(:num)', 'ApiController::getPedidos/$1');
36
```

## 2. DEBE SER REALIZADA CON DOS TABLAS Y VARIAS CONSULTAS

Creación de la base de datos TIENDA y 2 tablas: CLIENTES Y PEDIDOS



## Datos almacenados en las tablas

tienda/postgres@PostgreSQL 17

Consulta Historial de Consultas

```
1 select * from clientes
2
```

Data Output Mensajes Notificaciones

Showing rows: 1 to 5

	id [PK] integer	nombre character varying (100)	email character varying (100)	telefono character varying (15)
1	1	Juan Pérez	juan.perez@example.com	123456789
2	2	María López	maria.lopez@example.com	987654321
3	3	Carlos Gómez	carlos.gomez@example.co...	456123789
4	4	Ana Torres	ana.torres@example.com	789321654
5	5	Luis Sánchez	luis.sanchez@example.com	321654987

tienda/postgres@PostgreSQL 17

Consulta Historial de Consultas

```
1 select * from pedidos
2
```

Data Output Mensajes Notificaciones

Showing rows: 1 to 7

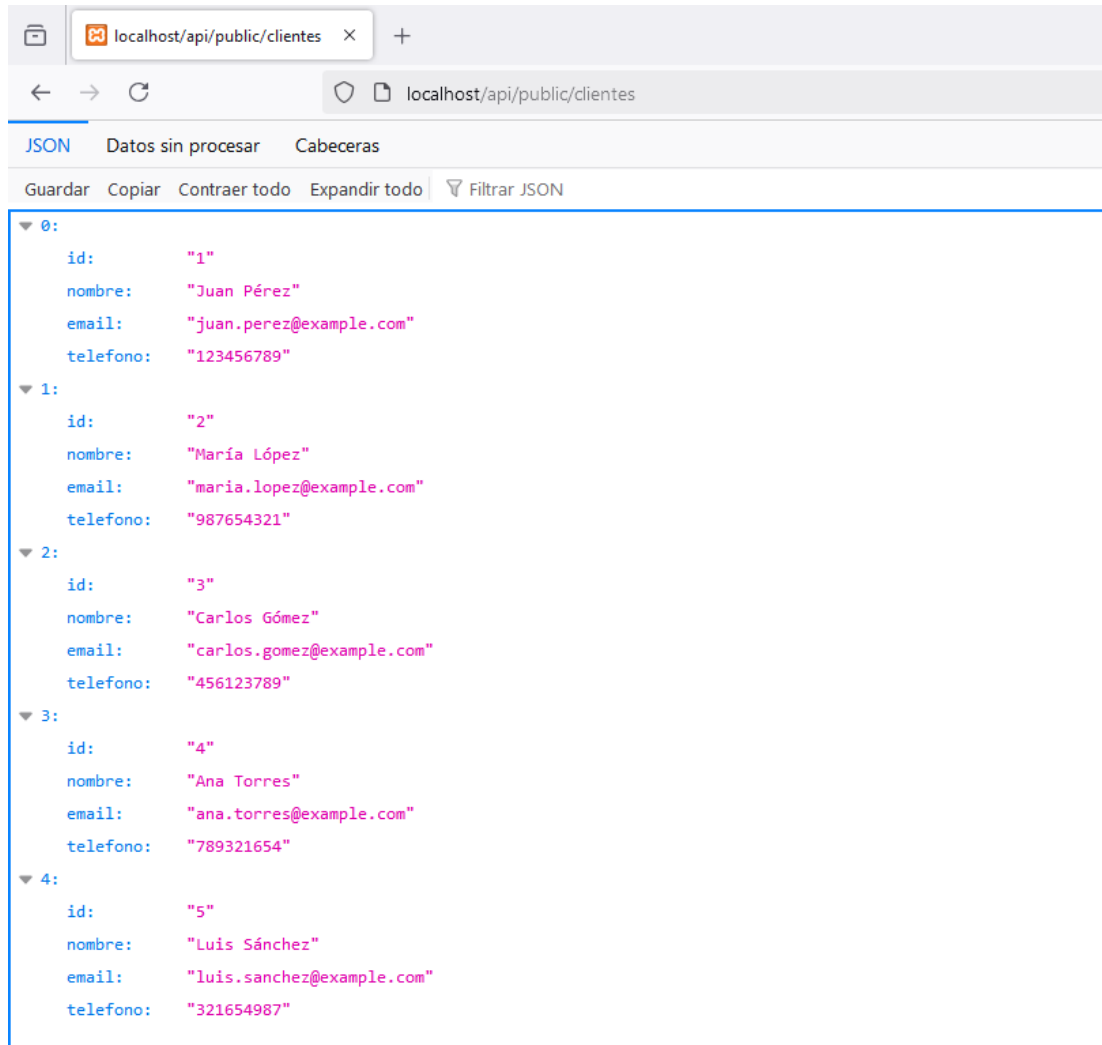
	id [PK] integer	cliente_id integer	descripcion text	total numeric (10,2)	fecha timestamp without time zone
1	1	1	Compra de electrodomésticos	1200.50	2024-12-14 01:34:22.281789
2	2	1	Compra de muebles	850.00	2024-12-14 01:34:22.281789
3	3	2	Pedido de ropa	250.75	2024-12-14 01:34:22.281789
4	4	3	Pedido de libros	100.00	2024-12-14 01:34:22.281789
5	5	4	Compra de alimentos	300.25	2024-12-14 01:34:22.281789
6	6	5	Pedido de electrónicos	450.99	2024-12-14 01:34:22.281789
7	7	5	Compra de herramientas	620.50	2024-12-14 01:34:22.281789

### 3. DEBE DEVOLVER LOS VALORES O RETORNAR LOS MISMOS EN FORMATO JSON

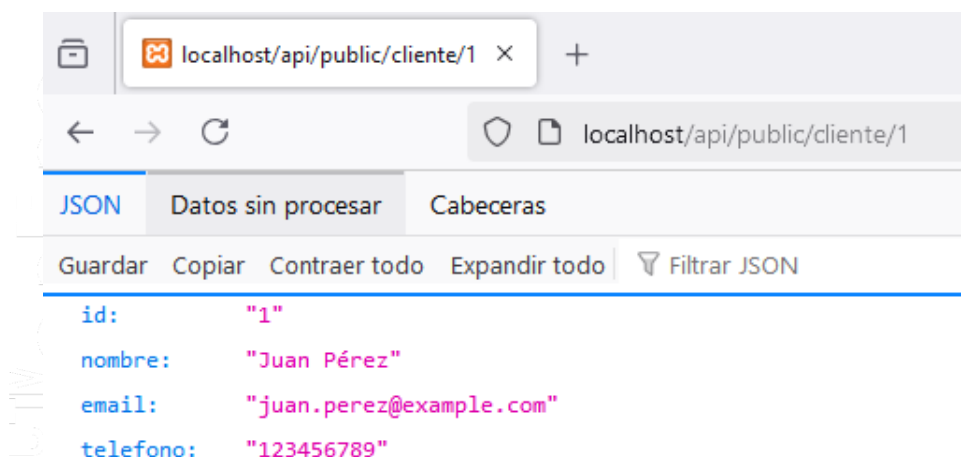
#### Probar la API

En el navegador se prueba las siguientes URLs:

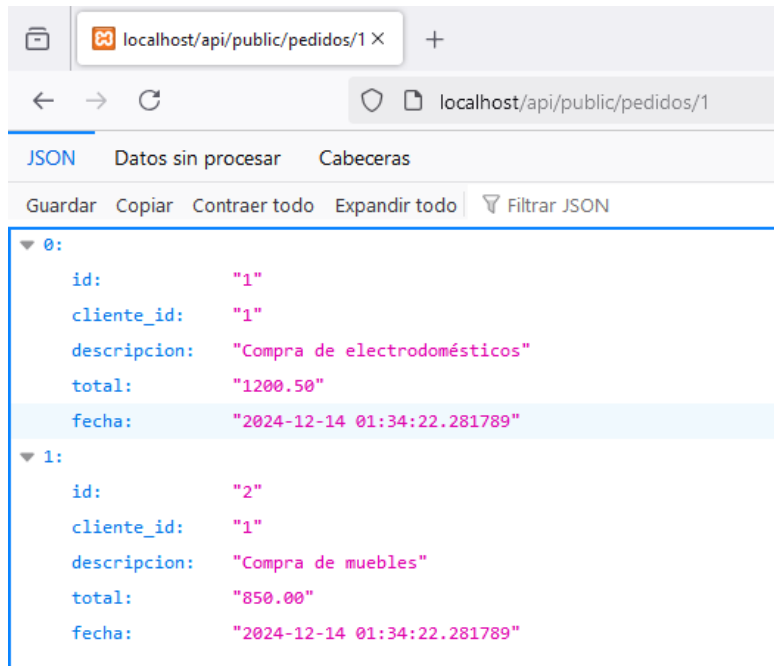
- <http://localhost/api/public/clientes> (obtendrá todos los clientes)



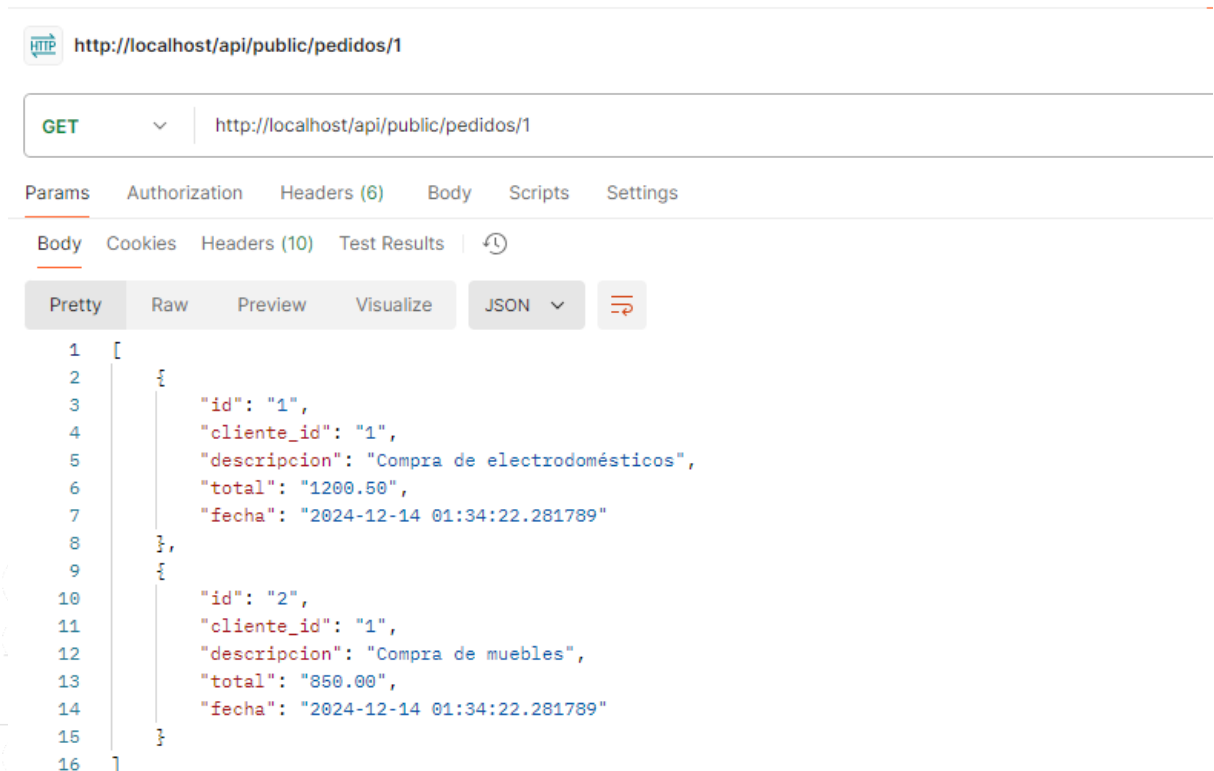
- <http://localhost/api/public/cliente/1> (obtendrá el cliente con ID 1)



- <http://localhost/api/public/pedidos/1> (obtendrá los pedidos del cliente con ID 1)



También se realiza pruebas en POSTMAN



#### 4. TAREA SUBIDA A UN GIT Y ENLACE DEL REPOSITORIO

[https://github.com/AntonioVillon/Api\\_Rest.git](https://github.com/AntonioVillon/Api_Rest.git)