

# MICROSERVICIO CLIENTES

## 1. Solicitar pedido

The screenshot shows the Postman interface for a POST request to the endpoint `http://localhost:3000/api/cliente/solicitar-pedido`. The request body is a JSON object:

```
{  "restaurante": "Del Puente",  "comida": ["hamburguesa", "papas", "fresco"]}
```

The response is a 200 OK status with the following JSON body:

```
{  "error": false,  "codigo": 200,  "mensaje": "Pedido recibido",  "pedido": {    "id": 5,    "restaurante": "Del Puente",    "comida": [      "hamburguesa",      "papas",      "fresco"    ]  }}
```

## 2. Verificar pedido restaurante

The screenshot shows the Postman interface for a GET request to the endpoint `http://localhost:3000/api/cliente/verificar-pedido-restaurante?pedidoId=1`. The request has a query parameter `pedidoId=1`. The response is a 200 OK status with the following JSON body:

```
{  "error": false,  "codigo": 200,  "mensaje": "Pedido en proceso",  "pedido": {    "id": "1"  }}
```

### 3. Verificar pedido repartidor

The screenshot shows the Postman interface with a GET request selected. The request is for the endpoint `http://localhost:3000/api/cliente/verificar-pedido-repartidor?pedidoId=7`. The 'Params' tab is active, showing a query parameter `pedidoId` with the value `7`. The 'Response' section is empty, displaying a 'Hit Send to get a response' message with a rocket icon.

collections / verificar-pedido-repartidor

GET `http://localhost:3000/api/cliente/verificar-pedido-repartidor?pedidoId=7` Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies </>

Query Params

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> pedidoId	7	
Key	Value	Description

Response

Hit Send to get a response

## MICROSERVICIO REPARTIDOR

### 1. Recibir pedido

The screenshot shows the Postman interface with a POST request selected. The request is for the endpoint `http://localhost:3002/api/repartidor/recibir-pedido`. The 'Body' tab is active, showing a JSON payload. The 'Response' section shows a successful status of 200 OK with a response body.

repartidor / recibir-pedido

POST `http://localhost:3002/api/repartidor/recibir-pedido` Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "restaurante": "Del Puente",
3   "comida": ["hamburguesa", "papas", "fresco"]
4 }
```

Body Cookies Headers (7) Test Results Status: 200 OK Time: 7 ms Size: 379 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "error": false,
3   "codigo": 200,
4   "mensaje": "Pedido recibido",
5   "pedido": {
6     "id": 6,
7     "restaurante": "Del Puente",
8     "comida": [
9       "hamburguesa",
10      "papas",
11      "fresco"
12    ]
13   }
14 }
```

## 2. Informar estado cliente

The screenshot shows the Postman application interface. On the left sidebar, under 'My Workspace', the 'clientes' collection is expanded, showing a sub-collection 'repartidor' with the endpoint 'POST informar-estado-cliente' selected. The main panel displays the request details for this endpoint. The URL is 'http://localhost:3002/api/repartidor/informar-estado-cliente'. The request body is a JSON object: 

```
{  "pedidoId": 1}
```

. The response is shown in the 'Body' tab, displaying a JSON object: 

```
{  "error": false,  "codigo": 200,  "mensaje": "EN PROCESO",  "pedido": {    "id": 1  }}
```

. The status is 200 OK, and the response is beautified.

## 3. Marcar pedido entregado

The screenshot shows the Postman application interface. On the left sidebar, under 'My Workspace', the 'clientes' collection is expanded, showing a sub-collection 'repartidor' with the endpoint 'POST marcar-pedido-entregado' selected. The main panel displays the request details for this endpoint. The URL is 'http://localhost:3002/api/repartidor/marcar-pedido-entregado'. The request body is a JSON object: 

```
{  "pedidoId": 1}
```

. The response is shown in the 'Body' tab, displaying a JSON object: 

```
{  "error": false,  "codigo": 200,  "mensaje": "ENTREGADO",  "pedido": {    "id": 1  }}
```

. The status is 200 OK, and the response is beautified.

# MICROSERVICIO RESTAURANTE

## 1. Recibir pedido

The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar lists collections: 'clientes' (with 'solicitar-pedido', 'verificar-pedido-restaurante', 'verificar-pedido-repartidor') and 'restaurante' (with 'recibir-pedido', 'informar-estado-cliente', 'marcar-pedido-entregado'). The main panel displays a POST request to 'http://localhost:3001/api/restaurante/recibir-pedido'. The request body is a JSON object: 

```
{  "restaurante": "Del Puente",  "comida": ["hamburguesa", "papas", "algo mas"]}
```

. The response is a 200 OK status with a JSON body: 

```
{  "error": false,  "codigo": 200,  "mensaje": "Pedido recibido",  "pedido": {    "id": 5,    "restaurante": "Del Puente",    "comida": [      "hamburguesa",      "papas",      "algo mas"    ]  }}
```

. The status bar at the bottom indicates 'Status: 200 OK', 'Time: 6 ms', and 'Size: 381 B'.

## 2. Informar estado cliente

The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar is the same as in the first screenshot. The main panel displays a POST request to 'http://localhost:3001/api/restaurante/informar-estado-cliente'. The request body is a JSON object: 

```
{  "pedidoId": 1}
```

. The response is a 200 OK status with a JSON body: 

```
{  "error": false,  "codigo": 200,  "mensaje": "EN PROCESO",  "pedido": {    "id": 1  }}
```

. The status bar at the bottom indicates 'Status: 200 OK', 'Time: 8 ms', and 'Size: 304 B'.

### 3. Informar estado repartidor

The screenshot shows the Postman application interface. On the left sidebar, under 'My Workspace', there is a collection named 'restaurante' which contains a folder 'repartidor'. Inside 'repartidor', the request 'POST Informar-estado-repartidor' is selected. The main panel displays the details of this request:

- Method:** POST
- URL:** http://localhost:3001/api/restaurante/informar-estado-repartidor
- Body:** The request body is in JSON format, containing: 

```
{  "pedidoId": 1}
```
- Response:** The response status is 200 OK. The response body is in JSON format, containing: 

```
{  "error": false,  "codigo": 200,  "mensaje": "EN PROCESO",  "pedido": {    "id": 1  }}
```

The bottom status bar indicates the response was received from 'Bootcamp' using the 'Runner'.