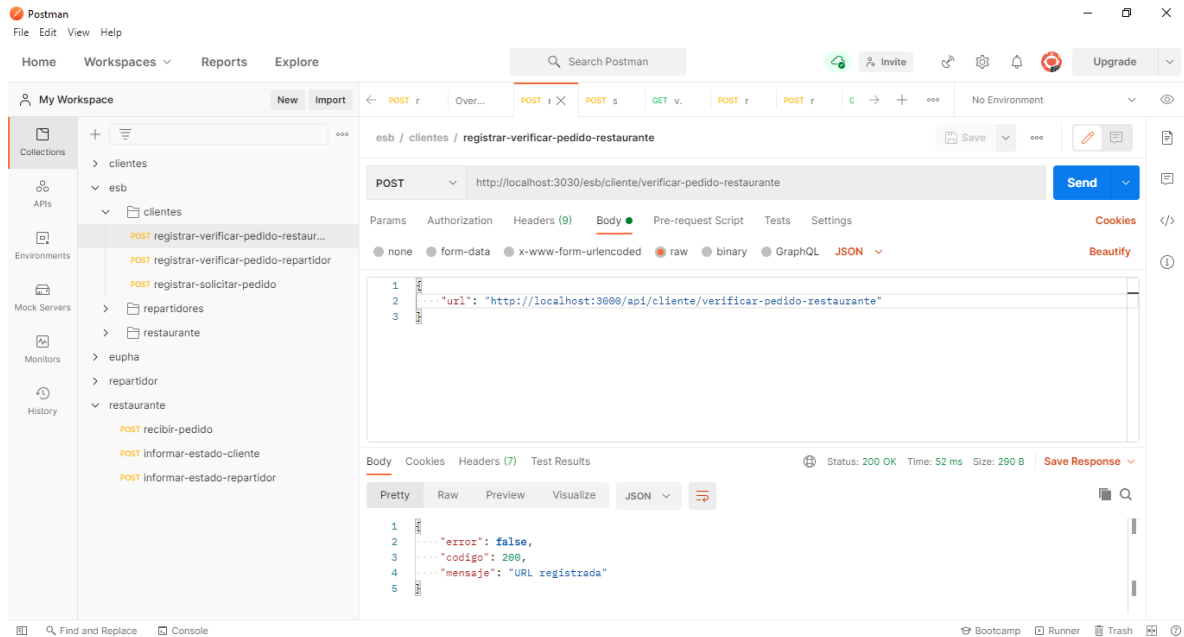
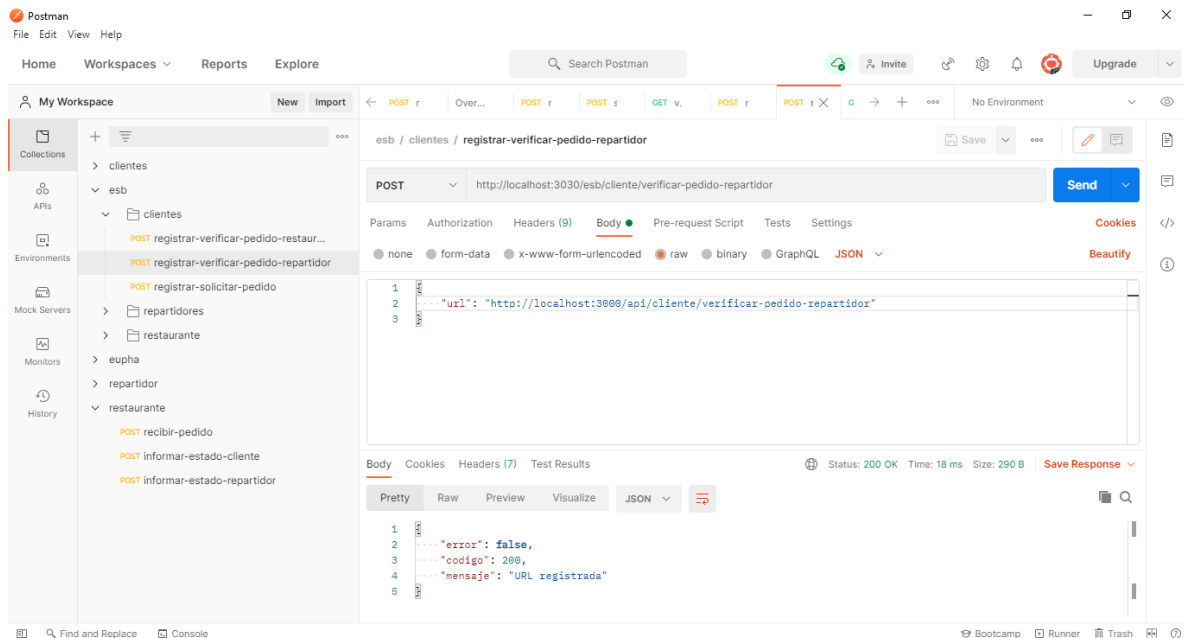


ENTERPRISE SERVICE BUS

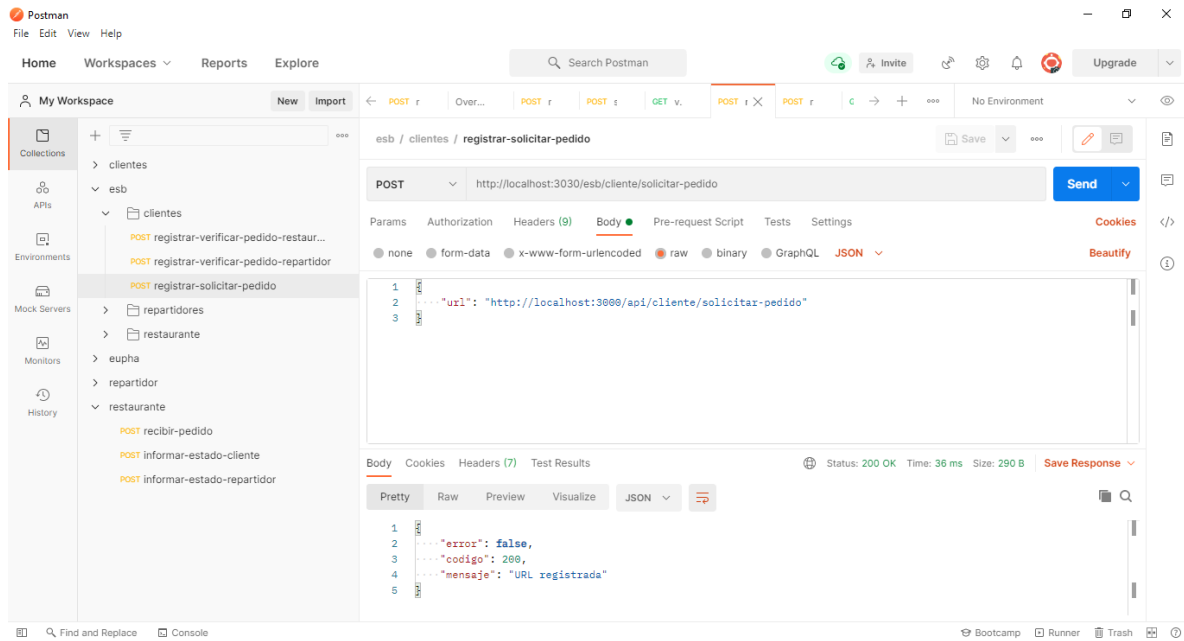
1. Registrando rutas para microservicio de clientes



Para poder verificar pedidos en el restaurante

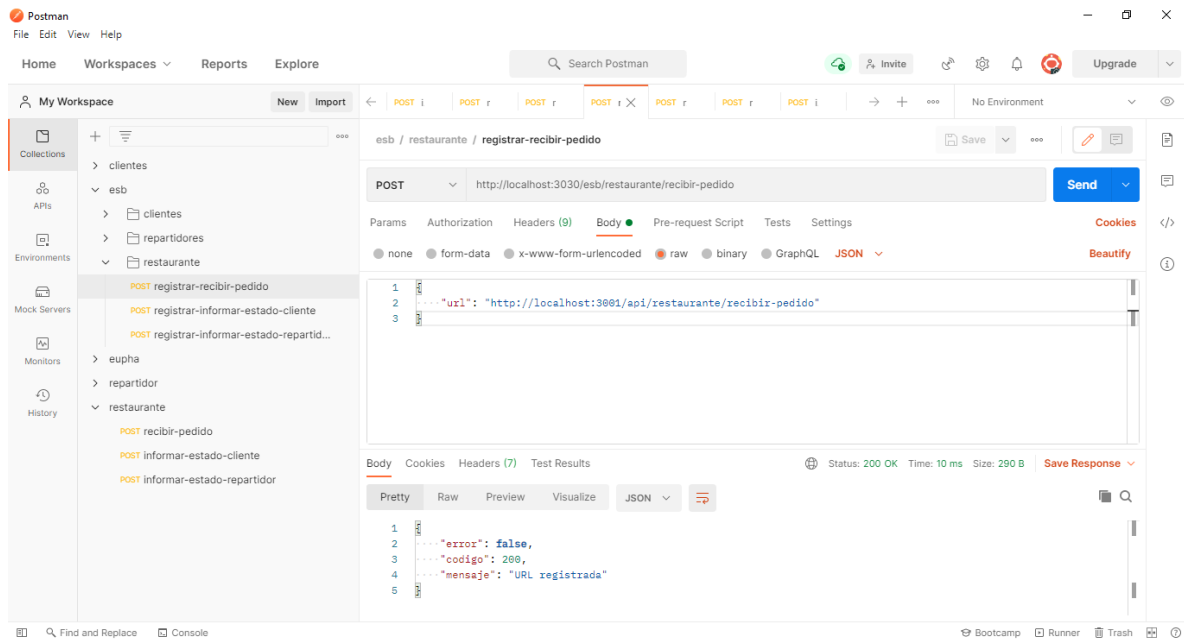


Para verificar pedidos con el repartidor

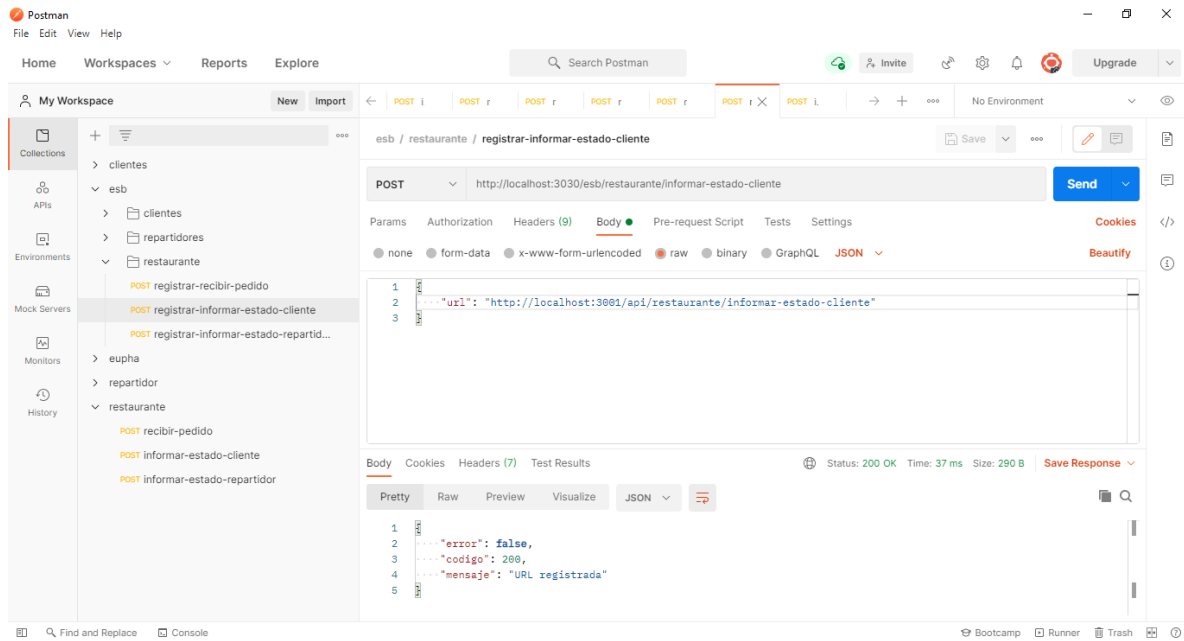


Para solicitar pedidos

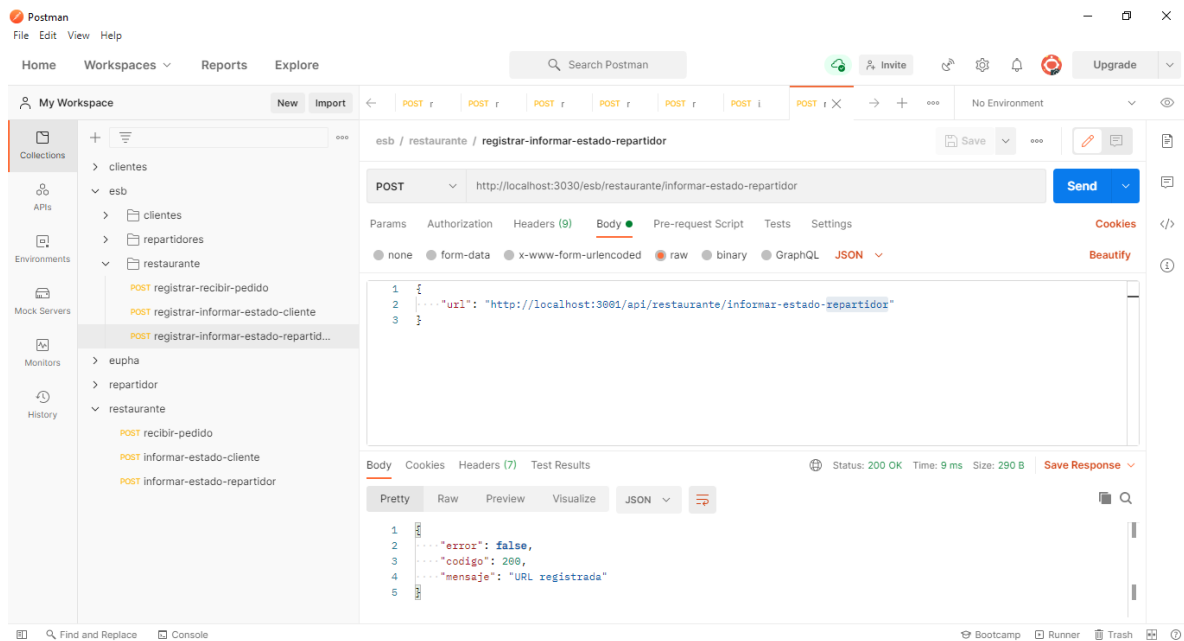
2. Registrando rutas para microservicio de restaurantes



Para recibir pedidos

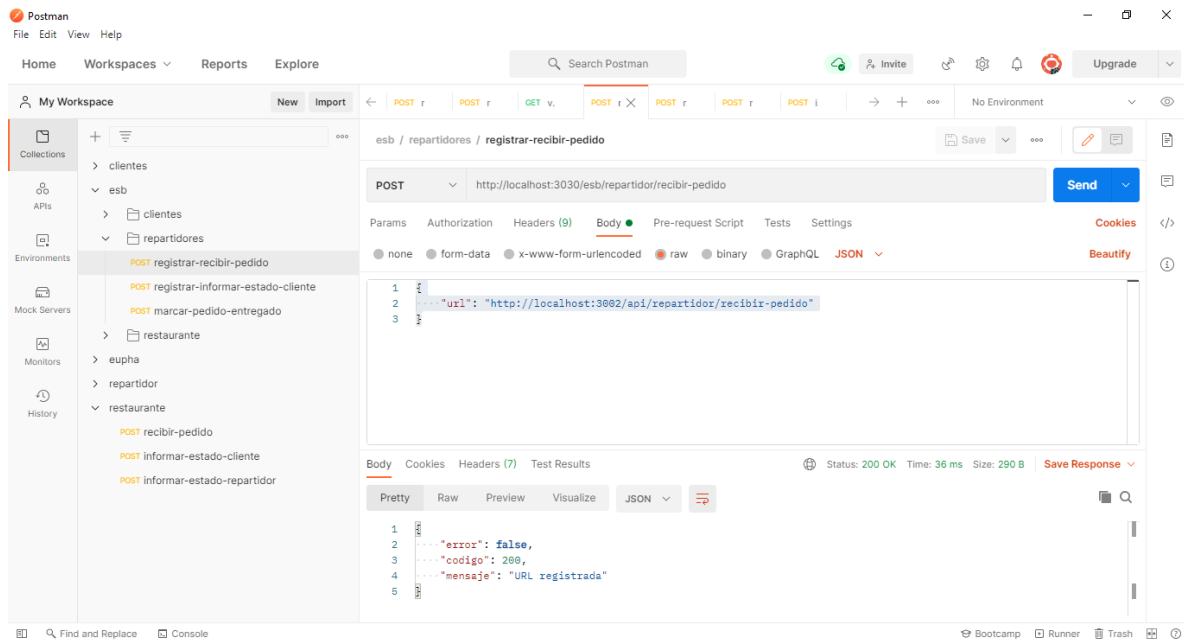


Para informar estado al cliente

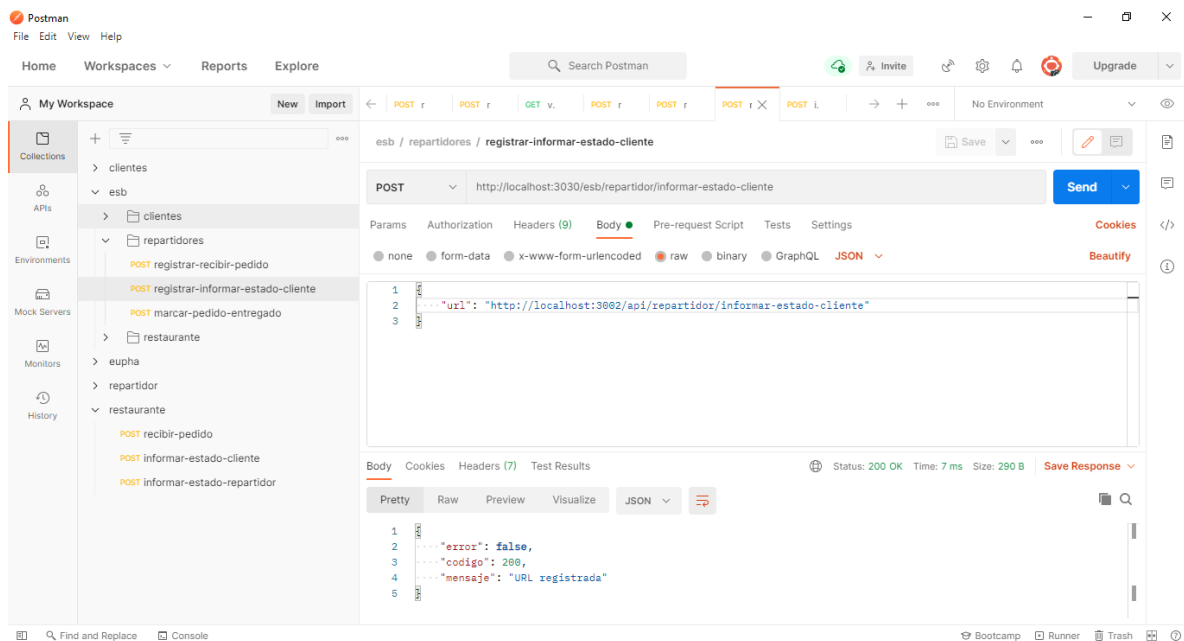


Para informar estado al repartidor

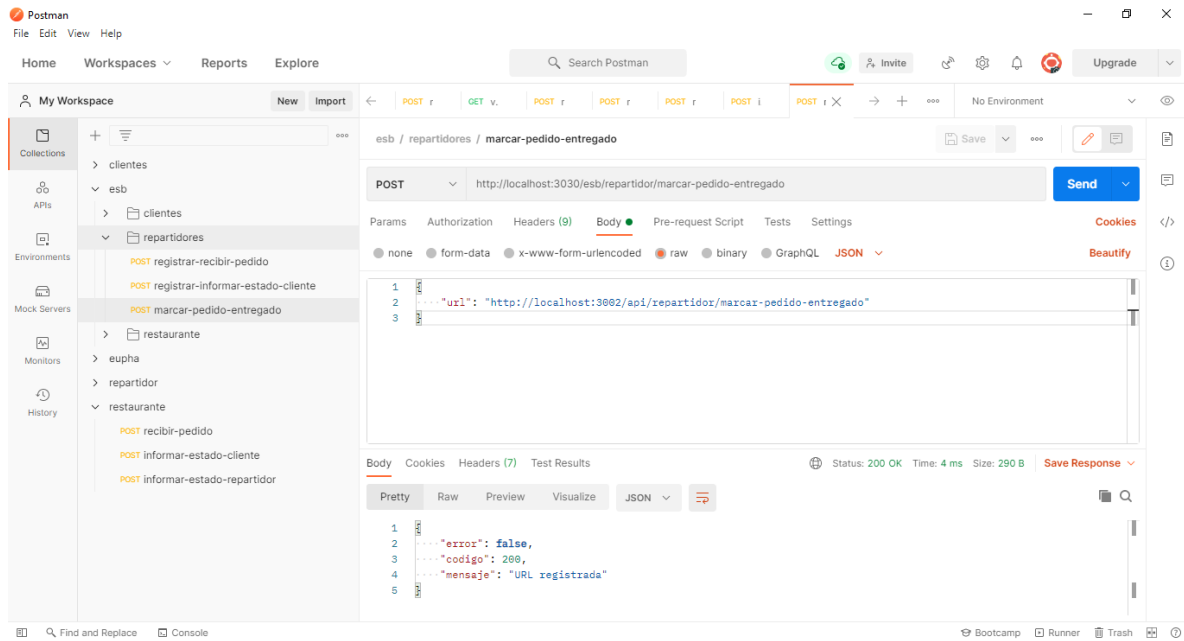
3. Registrando rutas para microservicio de repartidores



Para recibir pedidos



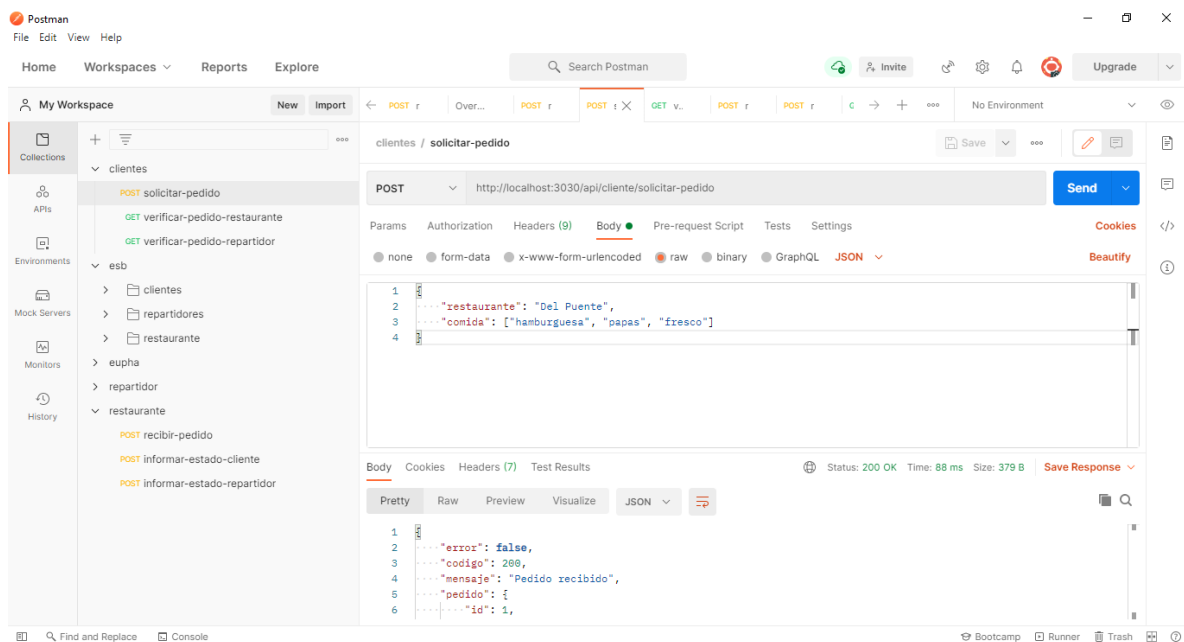
Para informar estado al cliente



Para marcar pedido como recibido

MICROSERVICIO CLIENTES

1. Solicitar pedido



2. Verificar pedido restaurante

The screenshot shows the Postman application with a workspace named 'My Workspace'. The left sidebar displays a collection of API requests under the 'clientes' folder, including 'solicitar-pedido', 'verificar-pedido-restaurante', and 'verificar-pedido-repartidor'. The main panel shows the 'verificar-pedido-restaurante' request selected, which is a GET request to the URL 'http://localhost:3030/api/cliente/verificar-pedido-restaurante?pedidold=1'. The 'Query Params' section shows a parameter 'pedidold' with a value of '1'. The 'Body' tab is selected, showing a JSON response in 'Pretty' format:

```
3 {
4   "codigo": 200,
5   "mensaje": "Pedido en proceso",
6   "pedido": {
7     "id": "1"
8   }
9 }
```

The status bar at the bottom indicates a successful response with status 200 OK, time 13 ms, and size 313 B.

3. Verificar pedido repartidor

The screenshot shows the Postman application with the same workspace. The left sidebar shows the 'verificar-pedido-repartidor' request selected, which is a GET request to the URL 'http://localhost:3030/api/cliente/verificar-pedido-repartidor?pedidold=1'. The 'Query Params' section shows a parameter 'pedidold' with a value of '1'. The 'Body' tab is selected, showing a JSON response in 'Pretty' format:

```
1 {
2   "error": false,
3   "codigo": 200,
4   "mensaje": "Pedido en proceso",
5   "pedido": {
6     "id": "1"
7   }
8 }
```

The status bar at the bottom indicates a successful response with status 200 OK, time 11 ms, and size 313 B.

MICROSERVICIO REPARTIDOR

1. Recibir pedido

The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar is expanded to show a collection named 'repartidor' under the 'esb' folder. The 'recibir-pedido' endpoint is selected. The main panel displays a POST request to 'http://localhost:3030/api/repartidor/recibir-pedido'. The request body is a JSON object:

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

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

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

. The status bar at the bottom indicates 'Status: 200 OK', 'Time: 63 ms', and 'Size: 379 B'.

2. Informar estado cliente

The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar is expanded to show a collection named 'repartidor' under the 'esb' folder. The 'informar-estado-cliente' endpoint is selected. The main panel displays a POST request to 'http://localhost:3030/api/repartidor/informar-estado-cliente'. The request body is a JSON object:

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

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

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

. The status bar at the bottom indicates 'Status: 200 OK', 'Time: 10 ms', and 'Size: 304 B'.

3. Marcar pedido entregado

The screenshot shows the Postman interface with a workspace named 'My Workspace'. The left sidebar shows a collection of folders: 'clientes', 'esb', 'repartidores', 'restaurante', 'eupha', 'repartidor', and 'restaurante'. The 'repartidor' folder is expanded, showing three POST requests: 'recibir-pedido', 'informar-estado-cliente', and 'marcar-pedido-entregado'. The 'marcar-pedido-entregado' request is selected. The request details show a POST method to 'http://localhost:3030/api/repartidor/marcar-pedido-entregado'. The body is a JSON object:

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

. The response is a 200 OK status with a response time of 10 ms and a size of 303 B. The response body is a JSON object:

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

MICROSERVICIO RESTAURANTE

1. Recibir pedido

The screenshot shows the Postman interface with a workspace named 'My Workspace'. The left sidebar shows a collection of folders: 'clientes', 'esb', 'repartidores', 'restaurante', 'eupha', 'repartidor', and 'restaurante'. The 'restaurante' folder is expanded, showing three POST requests: 'recibir-pedido', 'informar-estado-cliente', and 'informar-estado-repartidor'. The 'recibir-pedido' request is selected. The request details show a POST method to 'http://localhost:3030/api/restaurante/recibir-pedido'. The body is a JSON object:

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

. The response is a 200 OK status with a response time of 67 ms and a size of 381 B. The response body is a JSON object:

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


2. Informar estado cliente

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

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

. The response is displayed in the bottom panel, showing a status of 200 OK and a JSON body:

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

.

3. Informar estado repartidor

The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar displays a collection named 'restaurante' with a sub-collection 'informar-estado-repartidor' selected. The main panel shows a POST request to 'http://localhost:3030/api/restaurante/informar-estado-repartidor'. The request body is a JSON object:

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

. The response is displayed in the bottom panel, showing a status of 200 OK and a JSON body:

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

.