

API ALMACÉN PARA REALIZAR PAGOS EN LINEA

Para el desarrollo de la API se utilizó **Spring Boot – JAVA** con una base de datos simulada en un mapa y para el testing de funcionamiento de la API se utilizó la herramienta **POSTMAN**. A continuación se muestra la solución a las 3 historias de usuario del reto con los criterios de aceptación.

2.1 Historia de Usuario

Como dueño del almacén quiero que mis clientes puedan realizar pedidos en línea.

Criterios de aceptación

- Dado un cliente con la cedula de ciudadanía 12345 y con la dirección carrera 11# 14-08, cuando el cliente selecciona los productos a comprar con un valor mayor de 70.000 pesos entonces el sistema genera la factura con el iva (Impuesto del 19%) y el valor del domicilio.
- Dado un cliente con la cedula de ciudadanía 12345 y con la dirección carrera 11# 14-08, cuando el cliente selecciona los productos a comprar con un valor mayor de 100.000 pesos entonces el sistema genera la factura con el iva (Impuesto del 19%) y el valor del domicilio en 0 pesos.

Se realizó mediante el método POST el envío de 3 datos: cedula de ciudadanía, dirección y valor del pedido en la URL <http://localhost:8080/addClientes>. Una vez se realiza el envío y almacenamiento de los datos, se retorna el registro del pedido incluyendo: id, cédula, dirección, valor, iva, domicilio, total y hora del pedido como se muestra en las figuras.

Calculo de iva y domicilio valor mayor de \$70.000

The screenshot shows the Postman interface for a POST request to `http://localhost:8080/addClientes`. The request body is a JSON object:

```
{  "cedula": "1069001253",  "direccion": "CALLE 9 # 1-45",  "valor": 70000}
```

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

```
{  "id": 1,  "cedula": "1069001253",  "direccion": "CALLE 9 # 1-45",  "valor": 70000.0,  "iva": 13300.0,  "domicilio": 10000.0,  "total": 93300.0,  "horaactual": "2021-04-14T21:06:58.6252402"}
```

Cálculo de iva y domicilio valor mayor de \$100.000

POST http://localhost:8080/addClientes

Body

```
1 {
2   "cedula": "12345",
3   "direccion": "carrera 11# 14.08",
4   "valor": 120000
5 }
```

Body

```
1 {
2   "id": 2,
3   "cedula": "12345",
4   "direccion": "carrera 11# 14.08",
5   "valor": 120000.0,
6   "iva": 22800.0,
7   "domicilio": 0.0,
8   "total": 142800.0,
9   "horaactual": "2021-04-14T21:09:44.2579478"
10 }
```

Mediante el método GET en la URL: <http://localhost:8080/clientes> se puede obtener el listado de los pedidos enviados.

GET http://localhost:8080/clientes

Body

```
5   "direccion": "CALLE 9 # 1-45",
6   "valor": 70000.0,
7   "iva": 13300.0,
8   "domicilio": 10000.0,
9   "total": 93300.0,
10  "horaactual": "2021-04-14T21:06:58.6252402"
11 },
12 {
13   "id": 2,
14   "cedula": "12345",
15   "direccion": "carrera 11# 14.08",
16   "valor": 120000.0,
17   "iva": 22800.0,
18   "domicilio": 0.0,
19   "total": 142800.0,
20   "horaactual": "2021-04-14T21:09:44.2579478"
21 }
```

2.2 Historia de Usuario

Como dueño del almacén quiero que mis clientes puedan editar el pedido si este fue creado no antes de 5 horas.

Criterios de aceptación

- Dado que un cliente quiere editar su pedido cuando no han transcurrido más de 5 horas entonces el sistema debe dejarlo editar si los productos seleccionados cuestan igual o más que los anteriores.
- Dado que un cliente quiere agregar a su pedido un nuevo producto cuando el pedido era por valor de 70 mil pesos y al agregar el nuevo producto el pedido pasa de 100 mil pesos entonces el sistema debe de restar el valor del domicilio.

Mediante el método PUT en la URL: <http://localhost:8080/updateClientes/1> (En donde el 1 es el id del pedido a editar, la actualización del pedido en los campos está restringida solo para los pedidos con valor mayor o igual al pedido inicial y que no hayan pasado 5 horas de la creación del pedido.

Valor inicial de 70000 y modificado a 80000 con cambio de cédula y dirección.

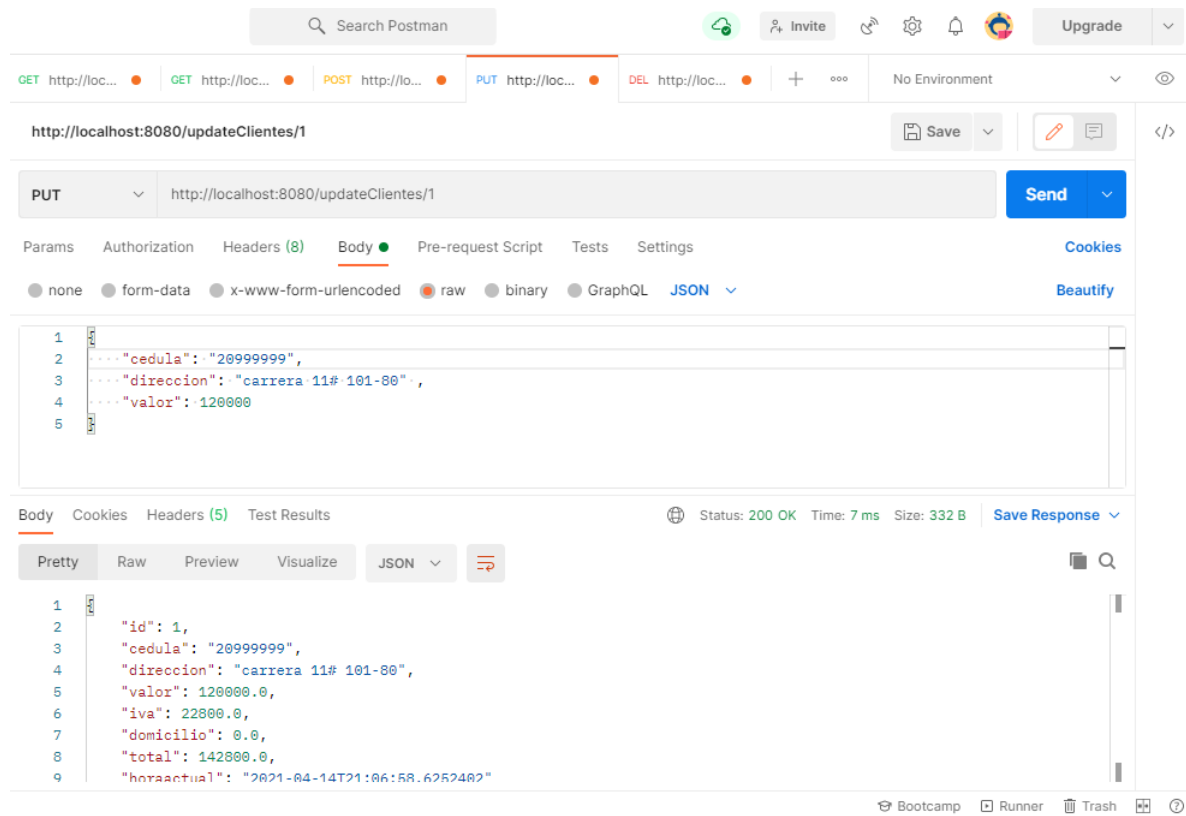
The screenshot shows a REST client interface with the following details:

- Method:** PUT
- URL:** http://localhost:8080/updateClientes/1
- Body (Request):**

```
1 {
2   "cedula": "20999999",
3   "direccion": "carrera 11# 101-80",
4   "valor": 80000
5 }
```
- Status:** 200 OK, Time: 10 ms, Size: 335 B
- Body (Response):**

```
1 {
2   "id": 1,
3   "cedula": "20999999",
4   "direccion": "carrera 11# 101-80",
5   "valor": 80000.0,
6   "iva": 15200.0,
7   "domicilio": 10000.0,
8   "total": 105200.0,
9   "horaactual": "2021-04-14T21:06:58.6252482"
```

Cuando el pedido sobrepasa los \$100.000 se resta el valor del domicilio.



2.3 Historia de Usuario

Como dueño del almacén quiero que mis clientes puedan eliminar el pedido si este fue creado antes de 12 horas.

Criterios de aceptación

- Dado que un cliente quiere eliminar su pedido cuando no han pasado más de 12 horas entonces el sistema debe dejarlo eliminar.
- Dado que un cliente quiere eliminar su pedido cuando han pasado más de 12 horas entonces el sistema debe de facturarle el 10% del valor del pedido y cancelar el pedido.

Mediante el método DELETE en la URL: <http://localhost:8080/deleteClientes/1> En donde el número 1 es el id del pedido a eliminar, con restricciones de tiempo de creado el pedido se dio solución a la historia de usuario como se muestra en las siguientes imágenes.

Eliminacion de Pedido dentro de 12 horas.

Search Postman

GET http://loc... • GET http://loc... • POST http://lo... • PUT http://loc... • **DEL http://loc...** + ... No Environment

http://localhost:8080/deleteClientes/1 Save Send

DELETE http://localhost:8080/deleteClientes/1

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

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

1

Body Cookies Headers (5) Test Results Status: 200 OK Time: 6 ms Size: 318 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "cedula": "20999999",
4   "direccion": "carrera 11# 101-80",
5   "valor": 0.0,
6   "iva": 0.0,
7   "domicilio": 0.0,
8   "total": 0.0,
9   "horaactual": "2021-04-14T21:38:23.2470847"
```

Cancelación de pedido después de las 12 horas, en donde se factura el 10% del pedido.

Search Postman

GET http://loc... • GET http://loc... • POST http://lo... • PUT http://loc... • **DEL http://loc...** + ... No Environment

http://localhost:8080/deleteClientes/2 Set as variable ... Save Send

DELETE http://localhost:8080/deleteClientes/2

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

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

1

Body Cookies Headers (5) Test Results Status: 200 OK Time: 6 ms Size: 325 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 2,
3   "cedula": "12345",
4   "direccion": "carrera 11# 14.08",
5   "valor": 12000.0,
6   "iva": 2280.0,
7   "domicilio": 0.0,
8   "total": 14280.0,
9   "horaactual": "2021-04-14T21:09:44.2579478"
```

Finalmente mediante el método GET de consulta de los pedidos se puede evidenciar si se han eliminado.

The screenshot displays the Postman application interface. At the top, there is a search bar labeled "Search Postman" and a row of utility buttons including "Invite", "Settings", "Notifications", and "Upgrade". Below this, a horizontal bar shows a list of recent requests with their methods and URLs. The main workspace is titled "http://localhost:8080/clientes" and features a "Save" button and a "Send" button. The "GET" method is selected for the request. Below the request bar, tabs for "Params", "Authorization", "Headers (6)", "Body", "Pre-request Script", "Tests", and "Settings" are visible. The "Params" tab is active, showing a table for "Query Params".

| KEY | VALUE | DESCRIPTION | ... | Bulk Edit |
|-----|-------|-------------|-----|-----------|
|-----|-------|-------------|-----|-----------|

Below the table, the "Body" tab is selected, showing the response status: "Status: 200 OK", "Time: 6 ms", and "Size: 166 B". A "Save Response" button is present. At the bottom, there are tabs for "Pretty", "Raw", "Preview", and "Visualize", along with a "JSON" dropdown and a "Send" button. The response body is currently empty.

At the bottom of the interface, there is a footer bar with icons for "Bootcamp", "Runner", "Trash", and a help icon.