

Problemática

Se requiere de un sistema para solventar la necesidad de orden en una tortillería. Los objetivos del sistema son, ser seguro, optimizado, con una interfaz limpia y perfectamente funcional en situaciones de estrés en horas pico de trabajo.

Reglas del negocio

La tortillería provee a terceros de sus servicios a través de dos formas. En local o a través de reparto. En cualquiera de las dos formas, sustenta la necesidad de sus clientes. La cual es proveer de tortilla, ya sea para consumo propio, como para reventa de esta.

Venta en local

La venta local, consiste en que los clientes vayan directamente al establecimiento a recoger su producto. Esto desemboca en una transacción entre el vendedor en punto de venta y el cliente.

En punto de venta, se pueden proveer los siguientes productos:

- Tortilla
- Salsas
- Mole
- Totopos
- Tostadas
- Tortilla tostada en tiras
- Masa de Maíz

Cada producto cuenta con los siguientes atributos:

- Nombre
- Gramaje

- Precio

A cada producto escrito anteriormente, se le lleva un conteo. Cada uno está dado de alta en inventario. Cada vez que un producto es vendido, este se resta del inventario.

A parte de proveer a los clientes, el vendedor en local tiene la obligación de informarle al vendedor de reparto, la cantidad total de kilos de tortilla que lleva en su salida en ruta.

Venta en reparto

El repartidor, cada mañana llega al local. Lo primero que hace es registrar su entrada y prepararse para su salida. Una de las actividades que realiza antes de salir a repartir, es mirar la lista de clientes a los cuales proveerá. Una vez trazada su ruta, lo que debe hacer es abastecerse de tortilla. La cantidad que lleva de tortilla en su vehículo, varia en cada salida a ruta. El encargado de decirle al repartidor cuantos kilos de tortilla lleva a vender, es el vendedor en local. Quien guarda un registro de lo que se lleva el repartidor en cada salida a ruta. El repartidor confirma la cantidad que recibe de kilos de tortilla, y sale a vender.

Una vez llega al primer destino, revisa la cantidad de kilos de tortilla que pudieran haber sobrado. Los lleva a su vehículo y prepara la hielera para reabastecer al cliente. Por cada kilo de tortilla que haya sobrado del día anterior, se le descontará al cliente del total que compre hoy. También se lleva un conteo de la cantidad de kilos de tortilla que se han regresado cada cliente.

Una vez reabastecido el cliente, se hace un total de costos. A ese total se le descuenta la tortilla sobrante, cada descuento es equivalente al precio total de la tortilla para ese cliente. Por ejemplo, si al cliente le sobraron 2 kilos, y cada kilo le cuesta \$12 pesos mexicanos, al cliente se le descuenta \$24 pesos mexicanos del total que compre el día de hoy. Otra cosa a tener en cuenta es que existen 2 tipos de clientes. Existen los clientes individuales, y aquellos que son empresas. Los clientes individuales tienen un crédito fijo, el cual se va agotando cada vez que el cliente decide usarlo. Si se les agota el crédito, no se les podrá abastecer más hasta

que paguen. Los clientes que son empresas solo pagan a plazo. Se decide un plazo, y en base a ese se hacen los pagos. Por ejemplo, se fija el plazo de una semana, se decide que hoy lunes se les abastecerá de forma indeterminada. Pero el siguiente lunes pagarán todo lo que hayan consumido toda la semana. En caso de no hacer el debido pago, no se les podrá abastecer hasta que se haga el pago correspondiente.

Tomando todo esto en cuenta, se hace un total a pagar. En caso de los clientes individuales, se revisa si se tiene adeudo en su crédito, en caso de tenerlo, se suma al total a pagar. Si hay tortilla sobrante, se resta del total a pagar. Y se suma toda la tortilla que se dejará en ese momento. En caso de ser una empresa, se descuentan los kilos de tortilla fríos al total que se dejará en ese momento. En ambos casos se entrega un ticket que demuestre que la transacción se llevó a cabo.

La cantidad de repartidores crecerá conforme crezca el negocio, por esa razón, se requiere de un sistema de usuarios.

Como usar

API REST

URL Base
https://tproject4is.herokuapp.com/user

Login

Para comenzar a utilizar el api del repartidor, lo primero que necesitamos es logearnos con el usuario y contraseña para que podamos recibir nuestro token.

Method	Path
POST	/login

El email y contraseña mostrados aquí están dados de alta en la base de datos y son válidos

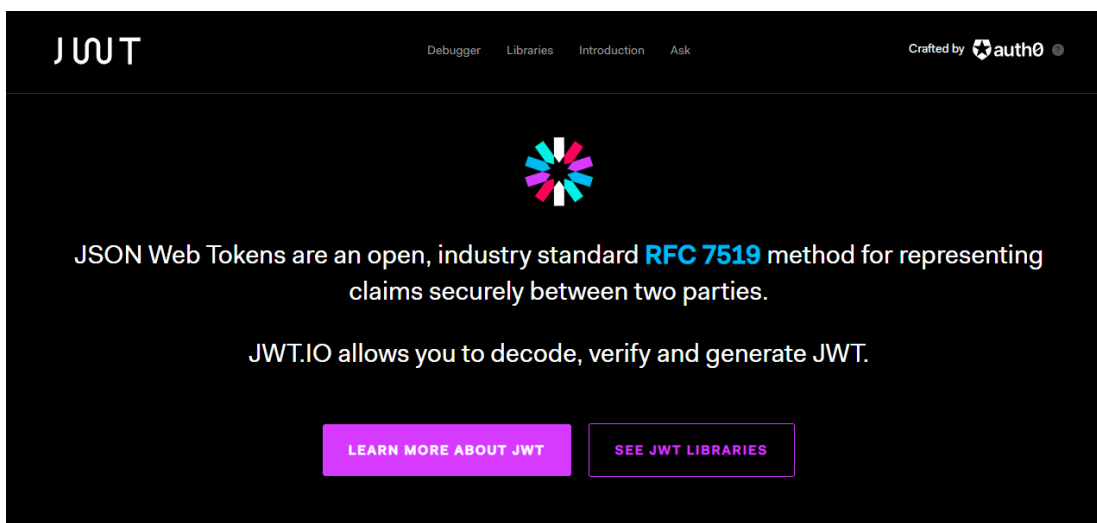
Request Params
<pre>{ "email": "prueba@gmail.com", "pass": "prueba123" }</pre>

En la respuesta se nos habrá generado un token, el cual utilizaremos a la hora de consumir el resto del api.

Response
{ "succes": true, "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCkiOiJYwNDQyMDQ5ZWQyMWZIMDNjYjQ2MDkzOSIsIm5pYW1lIljoiiY2VzYXJfcndylwidHlwZSI6ImVzZXIiLCJlbGlnbnRzIjpibeyJfaWQiOiI2MDQ0MjFjN2VkMjFmZTAzY2I0NjA5M2EiLCJuYW1lIljoiTGEgdGllbmRpdGEiLCJ0eXBIIljoiiY2xpZW50In0seyJfaWQiOiI2MDQ0MjRIMGVkMjFmZTAzY2I0NjA5M2IiLCJuYW1lIljoiiUGhhbnRvbSBTdG9yZSIsInR5cGUiOiJjbGllbnQifV0slmNvbXBhbmlldcyI6W3siX2lkIjoiiNjA0NjRiMWE4ZGVhMzMZIMTIOWVhNDNJliuibmFtZSI6IkFsZG1pbjBzYS4gZGUyY3YiLCJ0eXBIIljoiiY29tcGFueSJ9LHsiX2lkIjoiiNjA0NjVIMWZmYmZIZDE5MjU5YjE0YTZRmlwiubmFtZSI6ImdlbmVyYWwgbW90b3JzIiwidHlwZSI6ImNvbXBhbmkifV0slmV4cCI6MTY1MzQ5NTc3NX0.kKulXUuLdgR_T5MbJqf6rfVutGazZNVDjAOOffuSfSSQ" }


Intermedio

Lo que debemos hacer antes de continuar, es extraer la información que se encuentra en el token. Para esto debemos ir a [JSON Web Tokens - jwt.io](https://jwt.io) y veremos la pantalla principal de jwt.io.




Debugger

Si bajamos encontraremos lo siguiente:



JWT

DebuggerLibrariesIntroductionAsk

Crafted by

Encoded

PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c
```

Decoded

EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{  "alg": "HS256",  "typ": "JWT"}
```

PAYLOAD: DATA

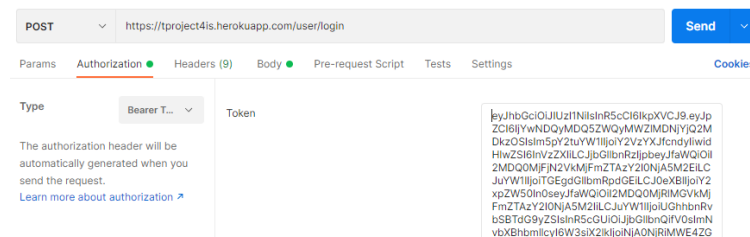
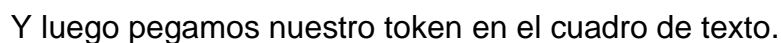
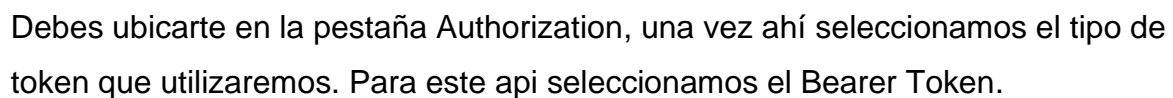
```
{  "sub": "1234567890",  "name": "John Doe",  "iat": 1516239822}
```

VERIFY SIGNATURE

```
HMASHA256(  
  base64UrlEncode(header) + "." +  
  base64UrlEncode(payload),  
  your-256-bit-secret  
)  
// secret: base64 encoded
```

[illegible]

Esto varia de cliente a cliente, pero en postman se hace de la siguiente manera.



Con esto tendríamos nuestra cabecera de autorización configurada. Recuerda que tu token tiene una vida útil de 1hr.

Cientes

Method	Path
POST	/clients

El payload del token contiene una lista de clientes, de esa lista agarramos los ids y los enviamos para obtener los datos de cada cliente.

Request Params
<pre>{ "clients": ["604421c7ed21fe03cb46093a", "604424e0ed21fe03cb46093b"] }</pre>

Retorna la información de los clientes obtenidos

Response
<pre>{ "sucess": true, "clients": [] }</pre>

Companies

Method	Path
POST	/companies

Ids de compañías

Request Params
<pre>{ "companies": ["60464b1a8dea36e1249ea43c", "60465e1ffbfed19259b14a4f"] }</pre>

información de compañías

Response
<pre>{ "sucess": true, "companies": [] }</pre>

Venta

Method	Path
POST	/venta

El cliente es quien debe realizar los cálculos.

Request Params

```
{
  "cliente_id": "604421c7ed21fe03cb46093a",
  "vendedor_id": "60442049ed21fe03cb460939",
  "precio_tortilla": 12,
  "precio_totopos": 24,
  "venta": {
    "tortilla": 5,
    "totopos": 2
  },
  "merma": {
    "tortilla": 2,
    "totopos": 0
  },
  "fecha": 20220525,
  "total": 84
}
```

Response

```
{
  "sucess": true,
  "msg": "Sucess operation",
  "something": "628ec69555578fe26cdd4d0e"
}
```

Agendar

Method	Path
POST	/shedule

Request Params

```
{
  "cliente": {
    "_id": {
      "$oid": "60465e1ffbfed19259b14a4f"
    },
    "tienda": "general motors",
    "type": "company"
  },
  "tortilla": 3,
  "totopos": 2,
  "fecha_entrega": "2021-05-05T19:01:54.000Z",
  "vendedor": {
    "_id": {
      "$oid": "60442049ed21fe03cb460939"
    },
    "name": "cesar_rwr"
  }
}
```

Response

```
{
  "sucess": true,
  "msg": "Sucess operation",
  "something": "628ecc6855578fe26cdd4d0f"
}
```

Obtener pedidos agendados

Method	Path
GET	/agendas

No request params

Regresa los pedidos agendados

Response

```
{
  "sucess": true,
  "agenda": [
    {},
    {}
  ]
}
```

Obtener pedido agendado

El :id en la query debe sustituirse por un id de un pedido agendado. Se debió haber obtenido la lista de agendas primero para obtener un id.

Method	Path
GET	/agenda/schedule/:id

Response
<pre>{ "sucess": true, "schedule": {} }</pre>

Obtener cliente

Se debe contar con un id de empleado

Method	Path
GET	/agenda/client/:id

Response
<pre>{ "sucess": true, "client": {} }</pre>

Obtener cliente

Se debe contar con un id de compañía

Method	Path
GET	/agenda/company/:id

Response
<pre>{ "sucess": true, "company": {} }</pre>