Microservicio de Gestión de Reclamos

Introducción

Se desarrollará un sistema que permita a los usuarios gestionar sus reclamos directamente desde el sitio para agilizar su resolución y así lograr mayor satisfacción de los clientes. Para ello se realizará un sistema de reclamos con el cual los cliente serán capaces de iniciar un reclamo, así un administrador luego podrá analizar su situación y ofrecerle una solución. Durante este proceso tanto el cliente cómo administrador podrán enviarse mensajes para brindar mayor información de lo sucedido y acordar su resolución.

Lista de Casos de Uso

- Iniciar Reclamo:
  - El usuario inicia un reclamo sobre una orden realizada.
- Seguir Detalle Reclamo:
  - El usuario consulta los detalles de su reclamo.
- Cerrar Reclamo:
  - El usuario da por satisfecho un reclamo y procede a cerrarlo.
- Escribir Mensaje:
  - El usuario escribe un nuevo mensaje en su reclamo para ampliar la información del mismo.
- Consultar Reclamos:
  - El administrador consulta los reclamos realizados que se encuentran activos.
- Responder Mensaje:
  - El administrador responde los mensajes que dejan los usuarios en cada reclamo.
- Finalizar Reclamo:
  - El administrado entiende que un reclamo ha sido resuelto y procede a cerrarlo.
- Caducar Reclamo:
  - Si un usuario no responde los mensajes que el administrador deje en el reclamo ni cierra el reclamo, este debe caducar a los 14 días.

Interacción con otros servicios

El sistema de reclamos deberá interactuar con el sistema de Auth para comprobar si el usuario está debidamente logueado y su identidad. También debe comunicarse con el sistema de Ordenes, ya que los reclamos se realizan sobre ordenes creadas. Por este

último motivo, también es conveniente comunicarse con el servicio de carrito para consultar el carrito de la orden y catálogo para poder traer los productos asociados a este carrito.

También se debe suscribir a los fanouts que pueda emitir el servicio de Auth mediante RabbitMQ en caso de que un token JWT de usuario sea invalidado.

De esta forma se le podrá brindar información detallada al usuario a la hora de iniciar un reclamo.

Modelo de datos

```
{
    "id" : string,
    "userId": string,
    "created": Date,
    "updated": Date,
    "status": string,
    "orderId": string,

    "cart": {
        "id": string,
        "articles": [
            {
                "id": string,
                "quantity": number,
                "validated": boolean,
                "valid": boolean
            }
        ]
    },

    “issue”: string,
    "messages": [
        {
            "id": string,
            "created": Date,
            "dateRead": Date,
            "userId": string,
            "content": string
        }
    ]
}
```

API REST

Listar Reclamos: retorna los reclamos realizados por un usuario logueado.

GET	/complaints
HEADER	Authorization = bearer {JWT} (debe ser user)
PARAMS	

200	OK:
	{
	“data”:{complaint[ ]}
	}

400	Bad Request
401	Unauthorized
500	Server Error

Listar Reclamos Administrador: retorna los reclamos activos para que el administrador pueda resolverlos.

GET	/complaints/admin
HEADER	Authorization = bearer {JWT} (debe corresponder a un admin)
PARAMS	

200	OK:
	{
	“data”:{complaint[ ]}
	}

400	Bad Request
401	Unauthorized
500	Server Error

Crear Reclamo: crea un reclamo nuevo asociado a una orden existente.

POST	/complaints
HEADER	Authorization = bearer {JWT} (debe ser user)

BODY	{
	“orderId”: {orderId},
	“issue”: {issue},
	“message”: {message}
	}

200	OK:
	{
	“data”: {complaint}
	}

400	Bad Request
401	Unauthorized
500	Server Error

Detalle Reclamo: brinda información detallada de un reclamo en particular.

GET	/complaints/:complaintId
HEADER	Authorization = bearer {JWT}

PARAMS	{
	“complaintId”: {complaintId}
	}

200	OK:
	{
	“data”: {complaint}
	}

400	Bad Request
401	Unauthorized
500	Server Error

Enviar Mensaje Reclamo: permite tanto a clientes como administradores logueados enviar mensajes sobre un reclamo.

PUT	/complaints/:complaintId/msg
HEADER	Authorization = bearer {JWT}

BODY	{
	“message”: {message};
	}

200	OK:
	{
	“data”: {complaint}
	}

400	Bad Request
401	Unauthorized
500	Server Error

Cerrar Reclamo: da por resuelto un reclamo activo. Sólo lo puede cerrar el usuario o administrador.

DELETE	/complaints/:complaintId
HEADER	Authorization = bearer {JWT}

BODY	{
	“orderId”: {orderId},
	“message”: {message};
	}

200	OK:

	{
	“message”: `Complaint ${complaintId} was deleted succesfully`
	}

400	Bad Request
401	Unauthorized
500	Server Error

Rabbit MQ

Logout de Usuarios: escucha los fanouts que emita el servicio de Auth anunciando cuando un usuario ha terminado su sesión.

FANOUT	/auth/logout

Mensaje	{
	"type": "logout",
	"message": "{tokenId}"
	}