



**PLUGIN SALESFORCE COMMERCE  
CLOUD  
MANUAL DE INSTALACIÓN**

Tabla de contenido

1. Resumen ..... 3

2. Aspectos Técnicos ..... 4

    2.1 *Compatibilidad* ..... 4

    2.2 *Documentación de GOcuotas* ..... 4

3. Diagramas de flujos..... 5

    3.1 *Flujo de pago* ..... 5

    3.2 *Flujo de reembolso* ..... 6

    3.3 *Flujo de contingencia*..... 7

4. Proceso de instalación ..... 9

Historial de versiones

Autor	Fecha	Versión
Cristhian Chacalcaje	2024/03/12	1.0.0

## **1. Resumen**

El presente documento detalla el proceso de instalación del plugin de integración entre Salesforce Commerce Cloud y la pasarela de pagos GOcuotas. El plugin y el proceso de integración fue creado y es mantenido por GOcuotas.

## **2. Aspectos Técnicos**

A continuación, se detallan los aspectos técnicos de la integración.

### ***2.1 Compatibilidad***

El cartucho es compatible con Salesforce Commerce Cloud 22.7 y con el SFRA 6.1.0.

### ***2.2 Documentación de GOcuotas***

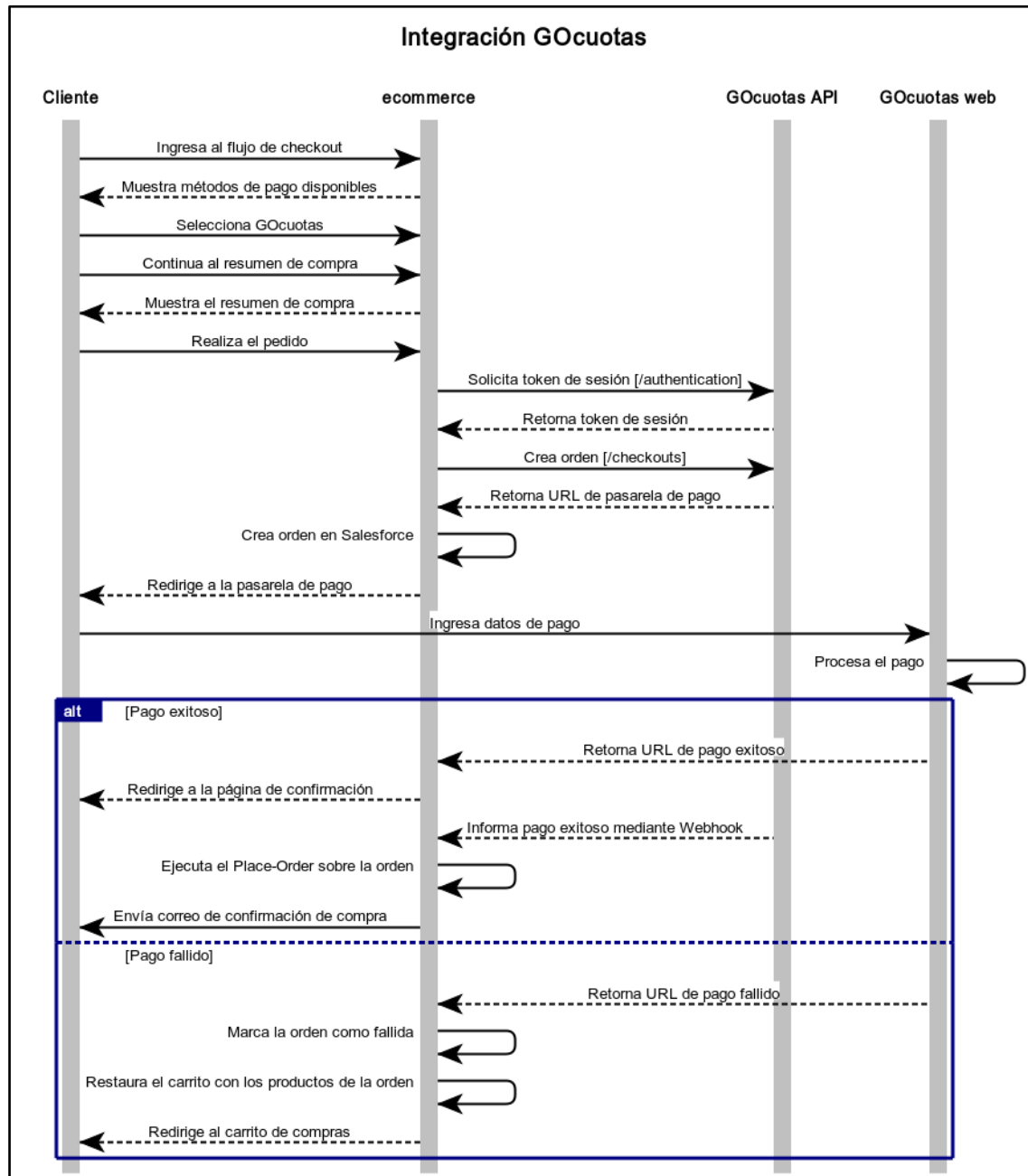
El método de pago integrado es GOcuotas redirect v1. <https://gocutas-api.stoplight.io/docs/gocutas/dae8814842a1e-api-redirect-v1>

### 3. Diagramas de flujos

A continuación, se detallan los flujos soportados por el cartucho.

#### 3.1 Flujo de pago

En el siguiente diagrama se describe el flujo de pagos con GOcuotas.



1. El cliente ingresa al flujo de Checkout del ecommerce
2. El ecommerce muestra los métodos de pago disponibles
3. El cliente selecciona GOcuotas
4. El cliente continúa a la página de resumen de compra
5. El ecommerce muestra el resumen de compra
6. El cliente realiza el pedido
7. El ecommerce solicita un token de sesión a GOcuotas API

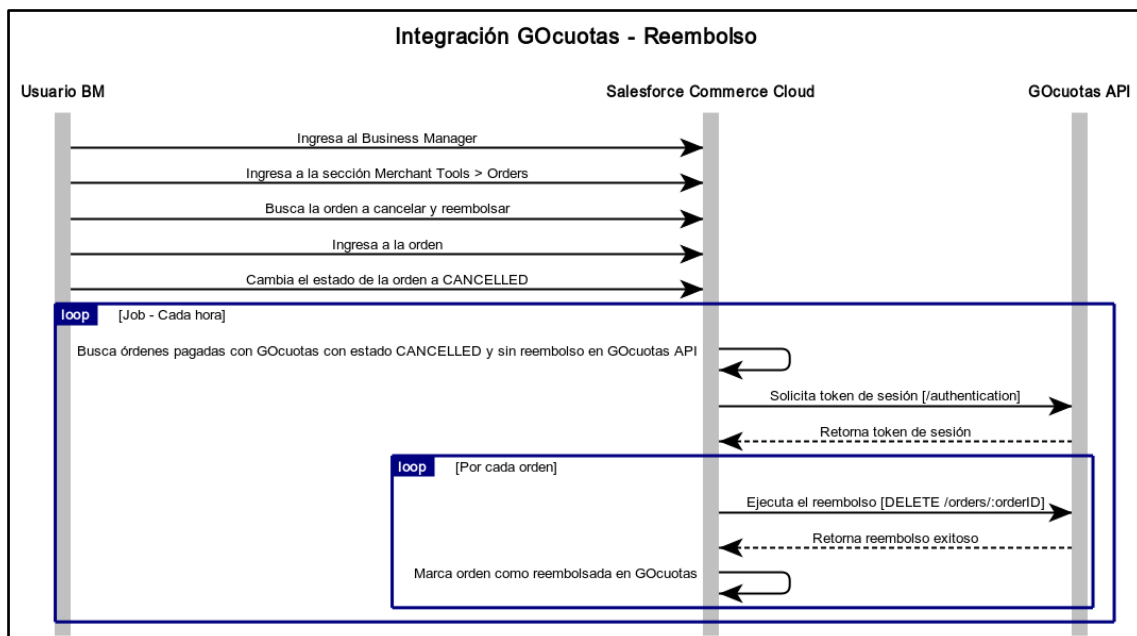
8. GOcuotas API retorna un token de sesión
9. El ecommerce crea una orden en GOcuotas API
10. GOcuotas API retorna la URL de la pasarela de pago
11. El ecommerce crea la orden en Salesforce
12. El ecommerce redirige al cliente a la pasarela de pago de GOcuotas
13. El cliente ingresa sus datos de pago en GOcuotas web
14. GOcuotas web procesa el pago
15. GOcuotas web retorna a la URL de pago exitoso
16. El ecommerce redirige al cliente a la página de confirmación
17. GOcuotas API informa al ecommerce el pago exitoso mediante un Webhook
18. El ecommerce ejecuta el Place-Order
  - a. Se marca la orden como pagada
  - b. Se marca la orden como confirmada
19. El ecommerce envía al cliente el correo de confirmación de compra

### Flujos alternativos

1. En el punto **15**, si GOcuotas web retorna a la URL de pago fallido
2. El ecommerce marca la orden como fallida
3. El ecommerce restaura el carrito de compras con los productos de la orden fallida
4. El ecommerce redirige al cliente al carrito de compras

### 3.2 Flujo de reembolso

En el siguiente diagrama se describe el flujo de reembolso con GOcuotas.

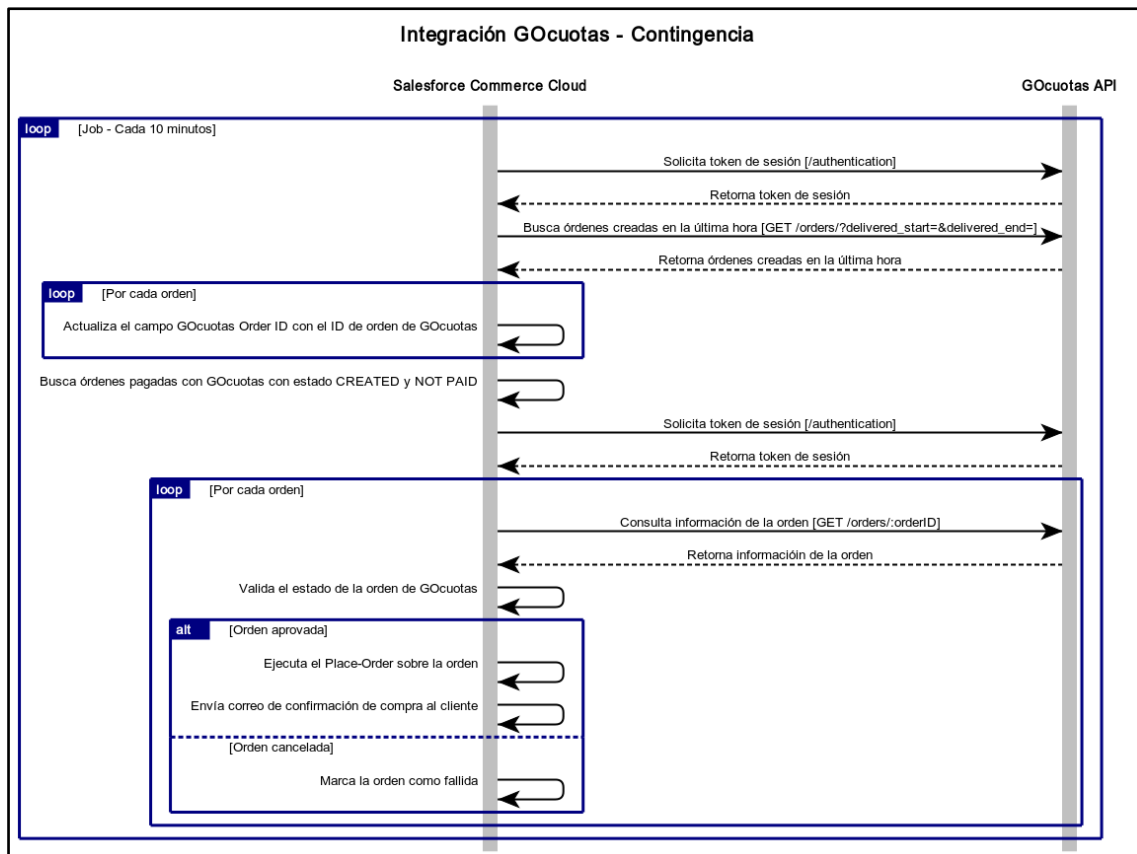


1. El usuario de Business Manager ingresa al Business Manager de Salesforce Commerce Cloud
2. El usuario ingresa a la sección **Merchant Tools > Orders**
3. El usuario busca la orden que desea cancelar y reembolsar
4. El usuario ingresa a la orden

5. El usuario cambia el estado de la orden a **CANCELLED**
- JOB - CADA HORA:**
6. Salesforce Commerce Cloud busca las órdenes pagadas con GOcuotas con estado CANCELLED y que no estén marcadas como reembolsadas en GOcuotas
7. Salesforce Commerce Cloud solicita un token de sesión a GOcuotas API
8. GOcuotas API retorna un token de sesión
- POR CADA ORDEN:**
9. Salesforce Commerce Cloud ejecuta el servicio de reembolso [DELETE] /orders/:orderId
10. GOcuotas API retorna estado de reembolso exitoso
11. Salesforce Commerce Cloud marca la orden como reembolsada en GOcuotas

### 3.3 Flujo de contingencia

El flujo de contingencia se da cuando una orden no ha sido actualizado mediante el webhook de GOcuotas. En el siguiente diagrama se describe el flujo de contingencia.



#### **JOB – CADA 10 MINUTOS**

1. Salesforce Commerce Cloud solicita un token de sesión a GOcuotas API
2. GOcuotas API retorna un token de sesión
3. Salesforce Commerce Cloud busca las órdenes creadas en la última hora en GOcuotas
4. GOcuotas retorna las órdenes creadas en la última hora

#### **POR CADA ORDEN:**

5. Salesforce Commerce Cloud actualiza el campo GOcuotas Order ID con el ID de la orden de GOcuotas
6. Salesforce Commerce Cloud busca las órdenes pagadas con GOcuotas con estado CREATED y NOT PAID
7. Salesforce Commerce Cloud solicita un token de sesión a GOcuotas API
8. GOcuotas API retorna un token de sesión

**POR CADA ORDEN:**

9. Salesforce Commerce Cloud consulta la información de la orden en GOcuotas API
10. GOcuotas API retorna la información de la orden
11. Salesforce Commerce Cloud valida el estado de la orden de GOcuotas
12. El ecommerce ejecuta el Place-Order
  - a. Se marca la orden como pagada
  - b. Se marca la orden como confirmada
13. El ecommerce envía al cliente el correo de confirmación de compra

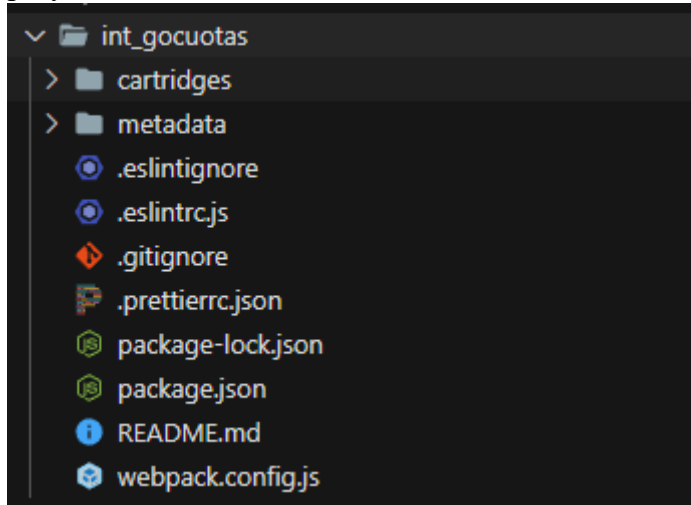
**Flujos alternativos**

1. En el punto **11**, si la información de la orden GOcuotas indica que la orden está cancelada
2. El ecommerce marca la orden como fallida

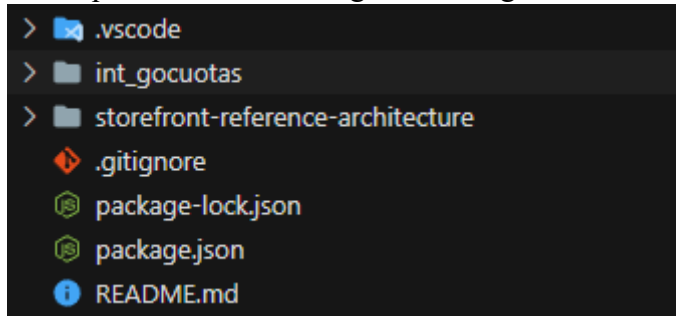


#### 4. Proceso de instalación

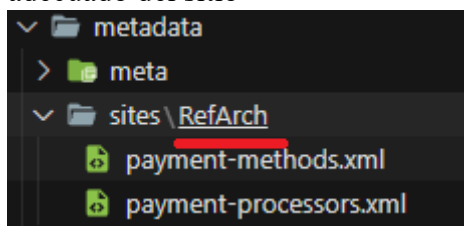
1. Copiar el contenido del proyecto **int\_gocuotas** en el código fuente de su proyecto.



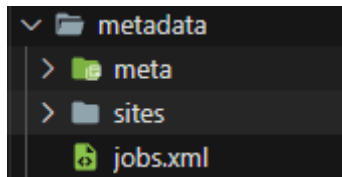
Debe quedar similar a la siguiente imagen:



2. Mediante consola, ingresar a la carpeta **int\_gocuotas**
3. Ejecutar el comando **npm install** para instalar las dependencias
4. Ejecutar el comando **npm build** para compilar los archivos js y css
5. Cambiar el nombre de la carpeta metadata/sites/**RefArch** con el nombre adecuado del sitio



6. En el archivo **jobs.xml**, reemplazar la palabra **RefArch** con el nombre adecuado del sitio



```
jobs.xml x
int_gocuotas > metadata > jobs.xml
You, 5 days ago | 1 author (You)
1 <?xml version="1.0" encoding="UTF-8"?>
2 <jobs xmlns="http://www.demandware.com/xml/impex/jobs/2015-07-01">
3   <job job-id="RefArch.G0cuotas.RefundOrders">
4     <description/>
5     <parameters/>
6     <flow>
7       <context site-id="RefArch"/>
8       <step step-id="custom.Gocuotas.RefundOrders" type="custom.Gocuotas.RefundOrders" enforce-restart="false">
```

7. Comprimir la carpeta **metadata**

Nombre	Fecha de modificación	Tipo	Tamaño
cartridges	9/01/2024 20:18	Carpeta de archivos	
metadata	7/03/2024 09:21	Carpeta de archivos	
node_modules	11/03/2024 15:51	Carpeta de archivos	
.eslintignore	5/03/2024 15:45	Archivo ESLINTIG...	1 KB
.eslintrc.js	5/03/2024 17:56	Archivo de origen ...	2 KB
.gitignore	5/03/2024 15:46	Archivo de origen ...	1 KB
.prettierrc.json	9/01/2024 20:04	Archivo de origen ...	1 KB
metadata.zip	12/03/2024 14:39	Archivo WinRAR Z...	4 KB
package.json	5/03/2024 15:47	Archivo de origen ...	2 KB
package-lock.json	11/03/2024 15:48	Archivo de origen ...	443 KB
README.md	5/03/2024 16:04	Archivo de origen ...	1 KB
webpack.config.js	29/02/2024 16:58	Archivo de origen ...	3 KB

8. Ingresar al **Business Manager**

9. Ingresar a la sección **Administration > Sites > Manage Sites > RefArch – Settings**

10. Configurar el cartridge path con el plugin de GOcuotas **int\_gocuotas**. Debe quedar de la siguiente manera

RefArch

Merchant Tools

Administration

Storefront

Toolkit

Administration > Sites > Manage Sites > RefArch - Settings

GeneralSettingsCacheSite StatusPage Meta Tag Rules

RefArch - Settings

Click Apply to save the details. Click Reset to revert to the last saved state.

Instance Type: Sandbox/Development

Deprecated. The preferred way of configuring HTTP and HTTPS hostnames is by using new features of the site aliases configuration ("SEO > Aliases Configuration"). The HTTP/HTTPS hostname values set in this section will be used if no hostnames .

HTTP Hostname:

HTTPS Hostname:

Instance Type: All

Cartridges: int\_gocuotas:app\_storefront\_base

Effective Cartridge Path: int\_gocuotas:app\_storefront\_base:plugin\_apple\_pay:plugin\_facebook:plugin\_payments:plugin\_pinterest\_commerce:plugin\_web\_payments:bc\_content:core

Session Timeout: 30 Minutes

11. Ingresar a la sección **Administration > Site Development > Site Import & Export**

12. Seleccionar y cargar el archivo **metadata.zip**

### Import

Upload Archive:

☒ Local ☐ Remote

metadata.zip

13. Seleccionar el archivo **instance/metadata.zip** y hacer click en **Import**

### Import

Upload Archive:

☒ Local ☐ Remote

metadata.zip

<input checked="" type="radio"/>	<a href="#">instance/metadata.zip</a>	local	3.80 KB	3/11/24 11:48:58 am
<input type="radio"/>	SiteGenesis Demo Site			
<input type="radio"/>	Storefront Reference Architecture Demo Sites			

14. Ingresar a la sección **Merchant Tools > Site Preferences > Custom Preferences**

15. Ingresar al grupo **GOcuotas Credentials**

16. Configurar las credenciales de **GOcuotas**

Name	Value
GOcuotas Email	
(gocuotasEmail) (String)	
GOcuotas Password	
(gocuotasPassword) (String)	

17. Ingresar a la sección **Merchant Tools > Site Preferences > Order**

18. Validar que el parámetro **Limit Storefront Order Access** esté en **No**

### Order Access Settings

Limit Storefront Order Access: