

Integración OpenKM y ProcessMaker

Autor: Diego Valladares

Fecha: 05 de marzo del 2016

Tabla de contenido

1. Resumen	4
2. Alcances	4
3. Requisitos técnicos.....	5
3.1 Servidor	5
3.2 Programación.....	5
3.3 Puertos	5
4. Ejecutables	6
4.1 Imagen Firma.....	6
5. Tareas programadas	7
5.1 Crear tareas programas.....	7
6. Encolamiento - RabbitMQ	9
6.1 Consultar la Cola	10
7. Servicios	11
7.1 OpenKMPDF	11
7.2 FtpPDF.....	12
7.3 GenerarFirma.....	13
8. Excepciones	14
8.1 Error de autenticación FTP	14
8.2 Error de autenticación OpenKM	14
8.3 Error lectura de archivos.....	15
8.4 Servidor Queue Deshabilitado.....	15
8.5 Las colas no son consumidas por las aplicaciones	15
9. Ejemplos	16
10. Código Fuente	16
10.1 Fuente – Servicios	17
10.2 Fuente – Cola FTP	17
10.3 Fuente – Cola OpenKM	18
10.4 Fuente – Cola Firma	18

Tabla de figuras

Figura 1. Ejecutables	6
Figura 2. Ejemplo de tareas en ejecución.....	7
Figura 3. Crear tareas programadas	7
Figura 4. Pestaña de Actions	8
Figura 5. Pestaña Conditions	9
Figura 6. Panel de administración - RabbitMQ	10
Figura 7. Ejemplo PHP – Servicio OpenKMPDF	11
Figura 8. Ejemplo PHP - Servicio FtpPDF	12
Figura 9. Error de autenticación FTP	14
Figura 10. Error de autenticación a OpenKM	14
Figura 11. Error lectura de archivos	15
Figura 12. Ejemplo, Consumir servicios desde PHP	16
Figura 13. Códigos ejecutables	16
Figura 14. Código Servicios	16
Figura 15. Parámetros Cola - Servicios	17
Figura 16. Ubicación ruta imagen	19

1. Resumen

El presente documento tiene como objetivo, configurar, implementar y consumir los servicios de integración para ser usado desde ProcessMaker.

2. Alcances

- Generar un servicio que descargue documentos desde un servidor FTP, realizar una unión de estos archivos y subir el resultado a un servidor de OpenKM.
- Generar un servicio que descargue documentos desde un servidor OpenKM, realizar una unión de estos archivos y subir el resultado a un servidor OpenKM.
- Generar un servicio para subir un archivo PDF a un servidor OpenKM y generar una firma a partir de un nombre personalizado.

3. Requisitos técnicos

3.1 Servidor

A continuación se describirán los requerimientos mínimos de hardware.

Hardware	Requerimientos mínimo
Procesador	1 procesador – Intel/amd
Ram	512
Tarjeta de Red	Habilitado
Tipo de Sistema	32x

A continuación se describirán los requerimientos mínimos de software

Software	Requerimientos mínimo
Sistema Operativo	Windows Server 2012
Framework .NET	4.5
Visual Studio	2013
RabbitMQ	3.6.1
IIS	Habilitado

3.2 Programación

A continuación se describirán los requerimientos las librerías para ejecutar la solución.

Librería	Versión
RabbitMQClient	3.6.1
OKMWebservice	1.0.0.0
itextsharp	5.5.8.0
Newtonsoft.Json	8.0.0.0

3.3 Puertos

A continuación se describirán los puertos necesarios para consumir los servicios y consultar la cola

Puerto	
80	Habilitado
5672	Habilitado
15672	Opcional

4. Ejecutables

Se recomienda que los ejecutables se encuentren en la misma carpeta para utilizar las mismas librerías.

Actualmente los ejecutables se encuentran en:

C:\Users\Administrator\Desktop\ConsumirColaPDF

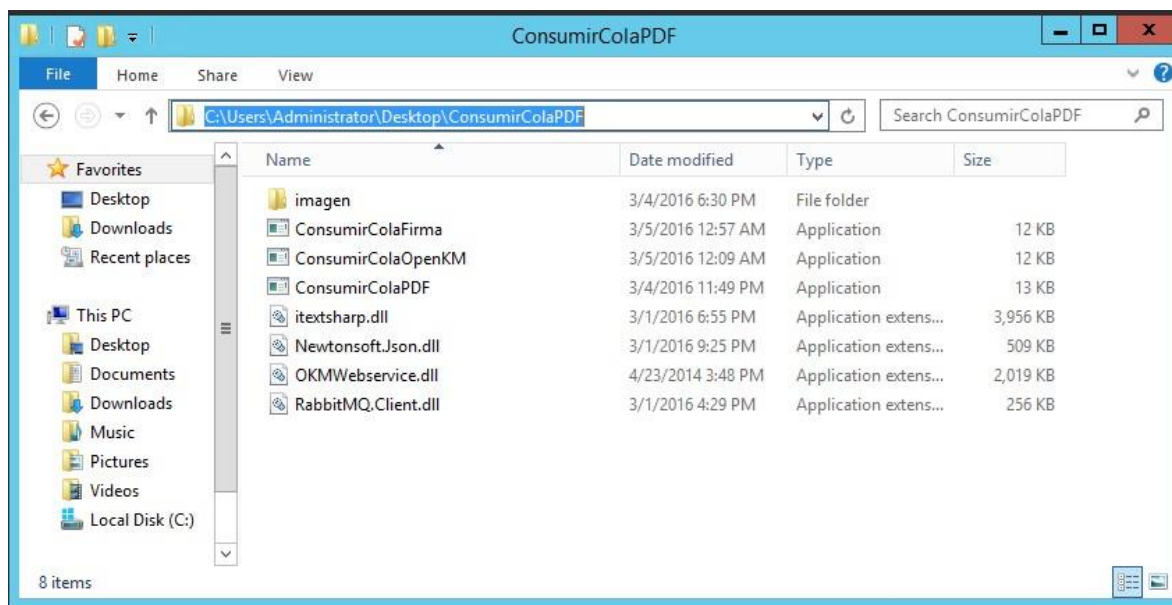


Figura 1. Ejecutables

4.1 Imagen Firma

En la carpeta imagen se encuentra la foto que será utilizada para realizar la firma, en caso de querer modificar la imagen, solo deben reemplazar la por defecto.

Requisitos cambio de firma

- **Extension:** jpg
- **Nombre:** firma
- **Tamaño máximo:** 2mb

5. Tareas programadas

Para consumir las diferentes colas se debe ejecutar tres tareas programadas, de modo que queden siempre a la escucha de nuevas peticiones.

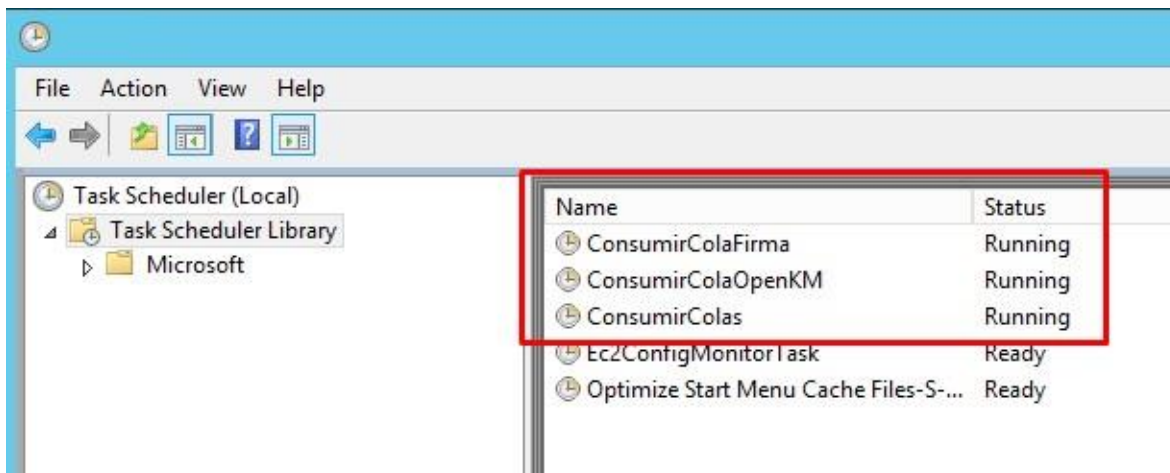


Figura 2. Ejemplo de tareas en ejecución

5.1 Crear tareas programadas

Para crear una nueva tarea programada se debe ejecutar la aplicación "Task Scheduler" e ir a la opción "Create Basic Task"

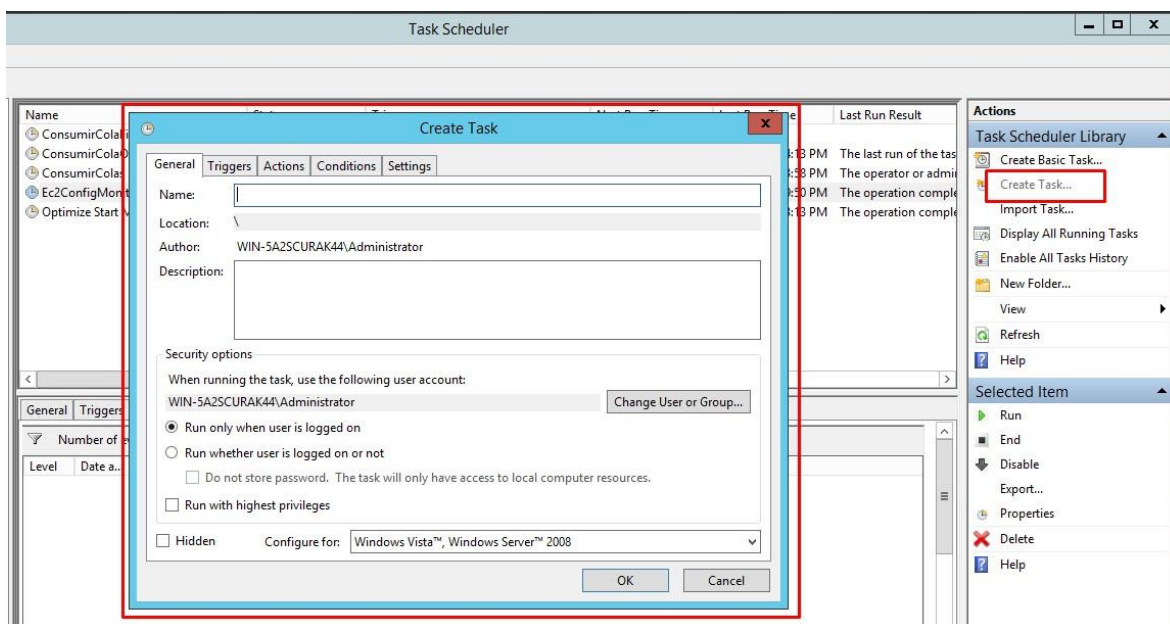


Figura 3. Crear tareas programadas

Parámetros de configuración.

- Pestaña General (Figura 3)
 - Name: Nombre de la ejecución
 - Habilitar Run with Highest Privileges
- Pestaña Actions (Figura 4)
 - Agregar una nueva ejecución (botón new)
 - Action: Start a Program
 - Program/Script: buscar la ruta de la aplicación
- Pestaña Conditions (Figura 5)
 - Start the task only if the computers is idle for (habilitado)
 - Stop if the computer ceases to be idle (habilitado)
 - Start the task only if the computer is on AC poter (habilitado)
 - Wake the computer to run this task (habilitado)
 - Start only if the following network connection is available: Any connection

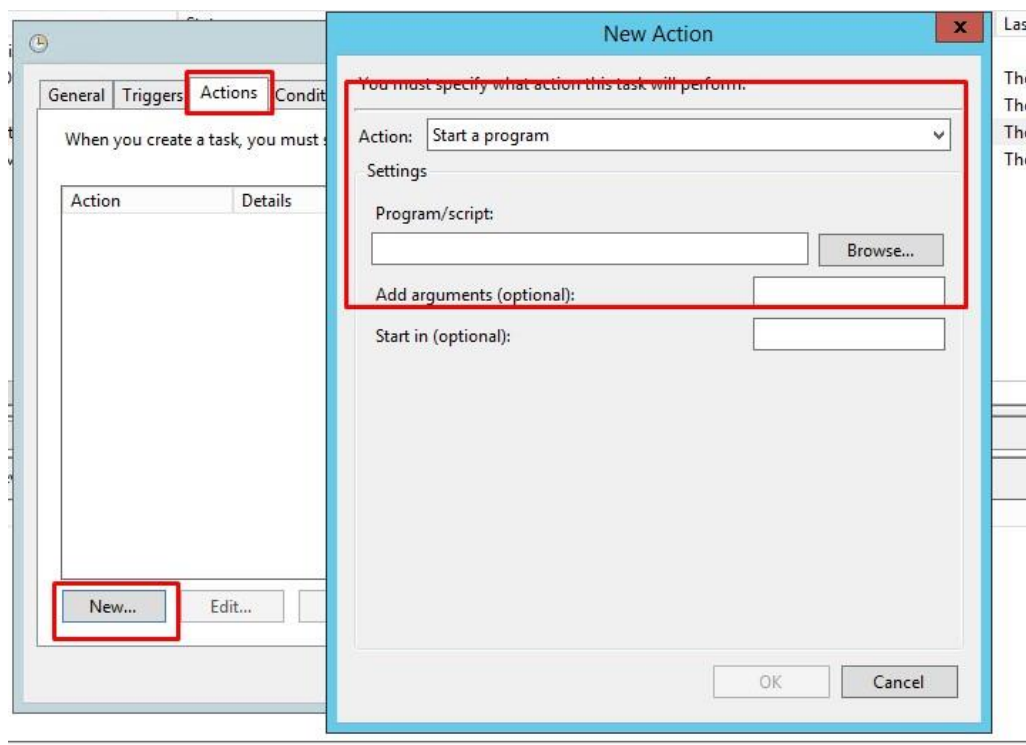


Figura 4. Pestaña de Actions

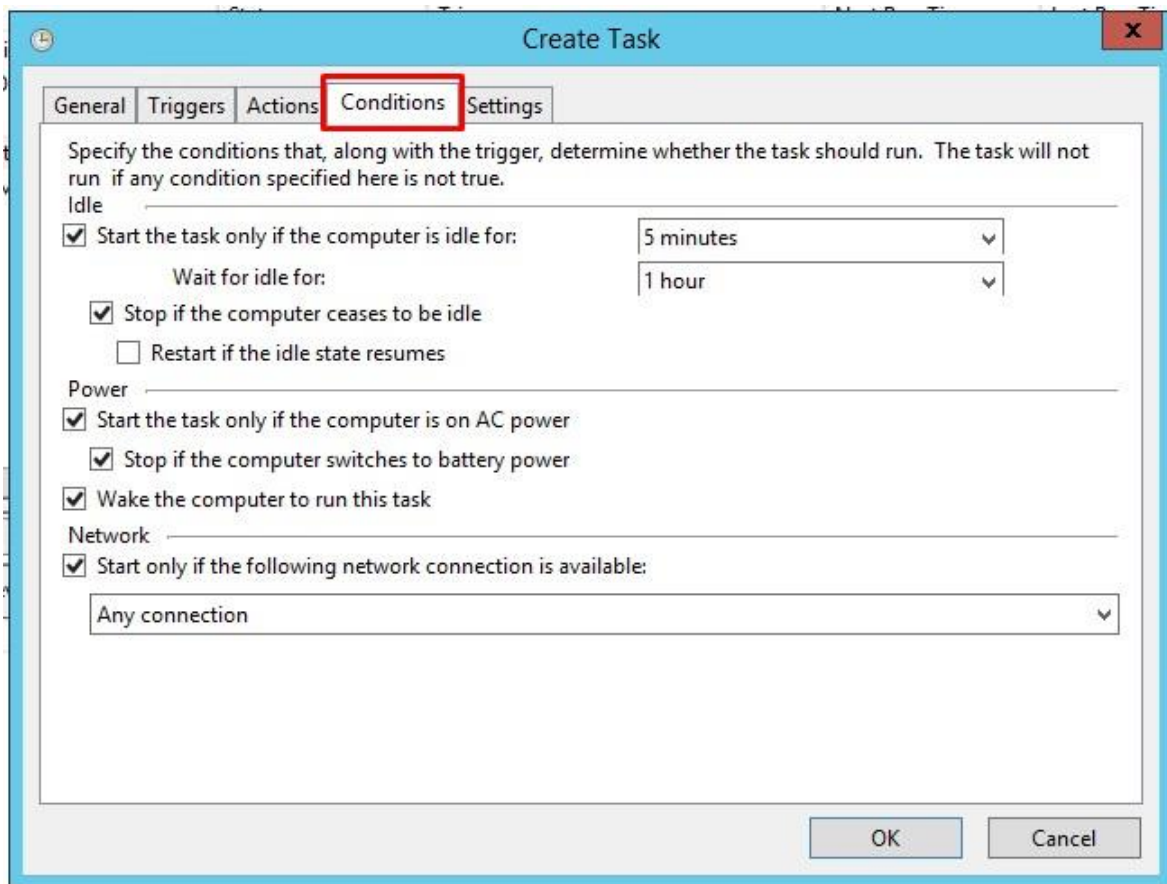
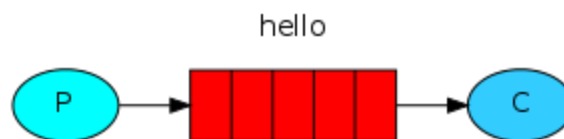


Figura 5. Pestaña Conditions

6. Encolamiento - RabbitMQ

Se utilizó RabbitMQ para crear las colas y consumir el resultado a medida que las peticiones van llegando.



Ejemplo utilizado en C# para enviar y consumir las colas.

<http://www.rabbitmq.com/tutorials/tutorial-one-dotnet.html>

Ejemplo para instalar RabbitMQ en Windows

<https://www.rabbitmq.com/install-windows.html>

6.1 Consultar la Cola

Se instaló un módulo de administración para visualizar y administrar las peticiones entrantes en la cola.

- URL: <http://localhost:15672>
- Usuario: guest
- Clave: lavidaloca123

Advertencia: si se desea cambiar el usuario y la clave, también se debe actualizar los datos en los ejecutables (Código Fuente).

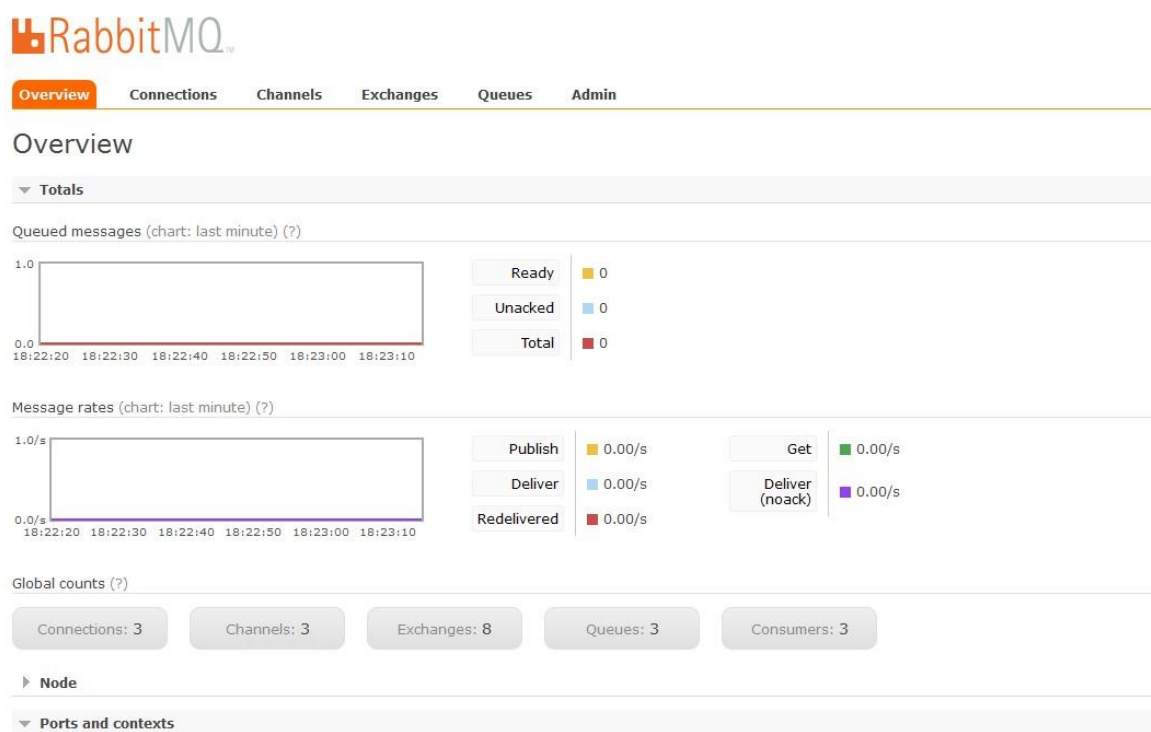


Figura 6. Panel de administración - RabbitMQ

7. Servicios

Todos los servicios retornan un booleano si la ejecución se realizó correctamente, si el retorno es falso, posiblemente los parámetros o el servidor presente problemas. Ver el punto de [Excepciones](#)

7.1 OpenKMPDF

Generar PDF en base a diferentes archivos almacenados en el servidor de OpenKM y guardarlo en el mismo servidor u otro.

Parámetros: String JSON

Llave	Tipo de dato	Descripción
Input	Array	Ruta de archivos OpenKM
Output	String	Ruta en donde se guardará el archivo generado en OpenKM
openkm_input [No Obligatorio]	Array	User: usuario OpenKM Password: Clave OpenKM Url: Host OpenKM
openkm_output [No Obligatorio]	Array	User: usuario OpenKM Password: Clave OpenKM Url: Host OpenKM

Ejemplo PHP

```
$json = array(
    'input' => array('/okm:root/archivo1.pdf','/okm:root/archivo2.pdf'),
    'output' => '/okm:root/archivo3.pdf',
    'openkm_input' => array(
        'user' => 'user7',
        'password' => 'pass7',
        'url' => 'http://demo.openkm.com/OpenKM'
    ),
    'openkm_output' => array(
        'user' => 'user7',
        'password' => 'pass7',
        'url' => 'http://demo.openkm.com/OpenKM'
    )
);
$params = array('json' => json_encode($json));
```

Figura 7. Ejemplo PHP – Servicio OpenKMPDF

7.2 FtpPDF

Generar PDF en base a diferentes archivos almacenados en un servidor FTP, el PDF generado será almacenado en un servidor de OpenKM.

Parámetros: String json

Llave	Tipo de Dato	Descripción
Input	Array	Ruta de archivos FTP
Output	String	Ruta en donde se guardará el archivo generado en OpenKM
Ftp [No Obligatorio]	Array	User: usuario OpenKM Password: Clave OpenKM IP: IP Servidor FTP
Openkm [No Obligatorio]	Array	User: usuario OpenKM Password: Clave OpenKM Url: Host OpenKM

Ejemplo PHP

```
$json = array(
    'input' => array('/pdf/pdf1.pdf', '/pdf/pdf2.pdf'),
    'output' => '/okm:root/archivo2.pdf',
    'ftp' => array(
        'user' => 'dvalladares',
        'password' => 'dvalladares',
        'ip' => '54.94.172.182'
    ),
    'openkm' => array(
        'user' => 'user7',
        'password' => 'pass7',
        'url' => 'http://demo.openkm.com/OpenKM'
    )
);
```

Figura 8. Ejemplo PHP - Servicio FtpPDF

7.3 GenerarFirma

Generar PDF con firma en base a un archivo almacenado en un servidor de OpenKM, el PDF generado será almacenado en el mismo u otro servidor de OpenKM.

Parámetros: String Json

Llave	Tipo de Datos	Descripción
Input	Array	Ruta de archivos FTP
Output	String	Ruta en donde se guardará el archivo generado en OpenKM
Firmado_por	String	Nombre de aprobador
openkm_input [No Obligatorio]	Array	User: usuario OpenKM Password: Clave OpenKM IP: IP Servidor FTP
openkm_output [No Obligatorio]	Array	User: usuario OpenKM Password: Clave OpenKM Url: Host OpenKM

Ejemplo PHP

```
$json = array(
    'input' => '/okm:root/archivo3.pdf',
    'output' => '/okm:root/archivo_firma.pdf',
    'openkm_input' => array(
        'user' => 'user7',
        'password' => 'pass7',
        'url' => 'http://demo.openkm.com/OpenKM'
    ),
    'openkm_output' => array(
        'user' => 'user7',
        'password' => 'pass7',
        'url' => 'http://demo.openkm.com/OpenKM'
    ),
    'firmado_por' => 'Diego Valladares',
);
```

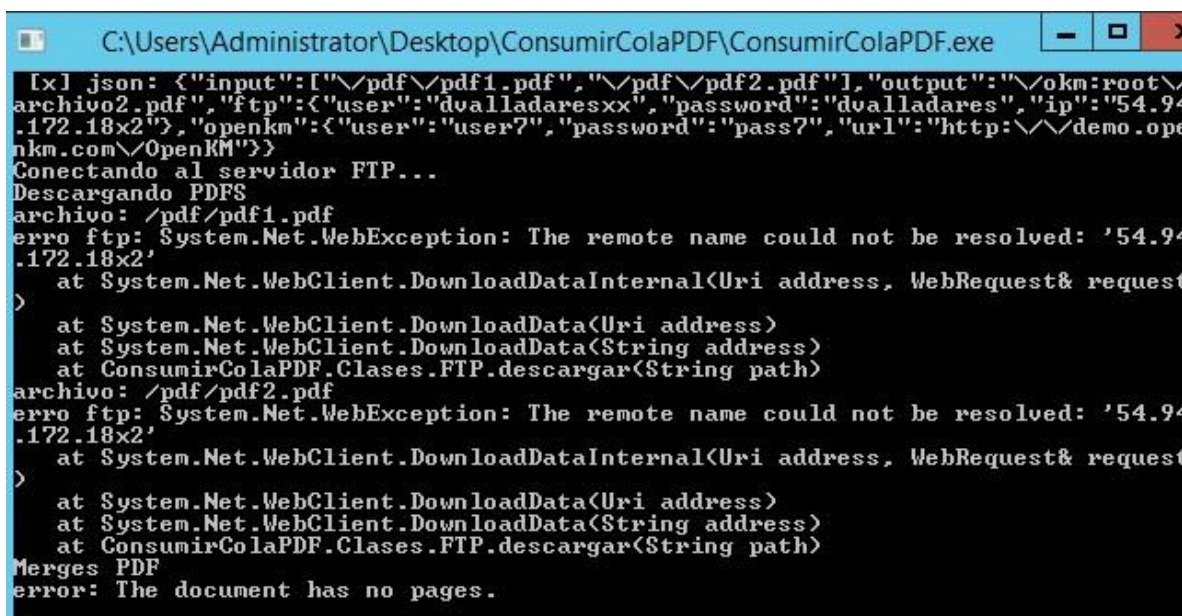
8. Excepciones

Para realizar pruebas de las aplicaciones se recomienda detener la ejecución de las tareas programas y ejecutar las aplicaciones independientemente.

A continuación se describirán los posibles errores que puede afectar el funcionamiento de la integración.

8.1 Error de autenticación FTP

La siguiente imagen muestra el error de autenticación FTP

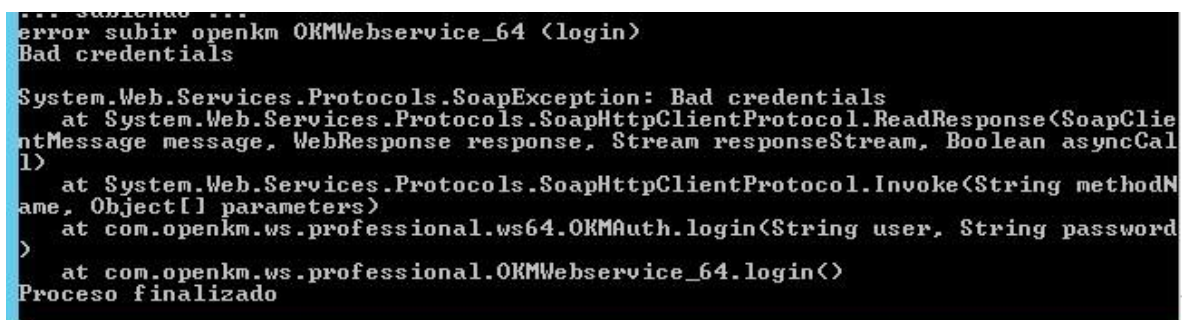


```
C:\Users\Administrator\Desktop\ConsumirColaPDF\ConsumirColaPDF.exe
[1] json: { "input": [ "\pdf\pdf1.pdf", "\pdf\pdf2.pdf" ], "output": "\okm:root\
archivo2.pdf", "ftp": { "user": "dvalladaresxx", "password": "dvalladares", "ip": "54.94
.172.18x2"}, "openkm": { "user": "user?", "password": "pass?", "url": "http://demo.ope
nkm.com/OpenKM" } }
Conectando al servidor FTP...
Descargando PDFs
archivo: /pdf/pdf1.pdf
erro ftp: System.Net.WebException: The remote name could not be resolved: '54.94
.172.18x2'
   at System.Net.WebClient.DownloadDataInternal(Uri address, WebRequest& request
)
   at System.Net.WebClient.DownloadData(Uri address)
   at System.Net.WebClient.DownloadData(String address)
   at ConsumirColaPDF.Clases.FTP.descargar(String path)
archivo: /pdf/pdf2.pdf
erro ftp: System.Net.WebException: The remote name could not be resolved: '54.94
.172.18x2'
   at System.Net.WebClient.DownloadDataInternal(Uri address, WebRequest& request
)
   at System.Net.WebClient.DownloadData(Uri address)
   at System.Net.WebClient.DownloadData(String address)
   at ConsumirColaPDF.Clases.FTP.descargar(String path)
Merges PDF
error: The document has no pages.
```

Figura 9. Error de autenticación FTP

8.2 Error de autenticación OpenKM

La siguiente imagen muestra el error de autenticación a OpenKM



```
... subiendo ...
error subir openkm OKMWebservice_64 <login>
Bad credentials

System.Web.Services.Protocols.SoapException: Bad credentials
   at System.Web.Services.Protocols.SoapHttpClientProtocol.ReadResponse(SoapClie
ntMessage message, WebResponse response, Stream responseStream, Boolean asyncCal
l)
   at System.Web.Services.Protocols.SoapHttpClientProtocol.Invoke(String methodN
ame, Object[] parameters)
   at com.openkm.ws.professional.ws64.OKMAuth.login(String user, String password
)
   at com.openkm.ws.professional.OKMWebservice_64.login()
Proceso finalizado
```

Figura 10. Error de autenticación a OpenKM

8.3 Error lectura de archivos

La siguiente imagen muestra el error en caso de no existir archivos de lectura.

```
erro ftp: System.Net.WebException: The remote server returned an error: (550) Fi
le unavailable (e.g., file not found, no access).
   at System.Net.WebClient.DownloadDataInternal(Uri address, WebRequest& request
)
   at System.Net.WebClient.DownloadData(Uri address)
   at System.Net.WebClient.DownloadData(String address)
   at ConsumirColaPDF.Clases.FTP.descargar(String path)
archivo: /pdf/pdf2.pdf22
erro ftp: System.Net.WebException: The remote server returned an error: (550) Fi
le unavailable (e.g., file not found, no access).
   at System.Net.WebClient.DownloadDataInternal(Uri address, WebRequest& request
)
   at System.Net.WebClient.DownloadData(Uri address)
   at System.Net.WebClient.DownloadData(String address)
   at ConsumirColaPDF.Clases.FTP.descargar(String path)
```

Figura 11. Error lectura de archivos

8.4 Servidor Queue Deshabilitado

Las aplicaciones se caerán por “**Exception Rabbitmqclient**”, ya que no se encuentra habilitado el servidor de colas.

8.5 Las colas no son consumidas por las aplicaciones

En caso de tener información en las colas y las aplicaciones no consumen dicha información, se debe verificar que las tareas programadas que se encuentren en ejecución (running).

9. Ejemplos

En la carpeta de códigos/ejemplos, se encuentran tres archivos para consumir los servicios independientemente desde PHP.

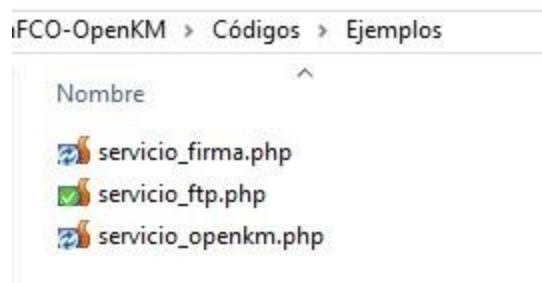


Figura 12. Ejemplo, Consumir servicios desde PHP

10. Código Fuente

En la carpeta de códigos/Ejecutables, se encuentran las tres carpeta de los códigos fuentes para modificar a gusto.

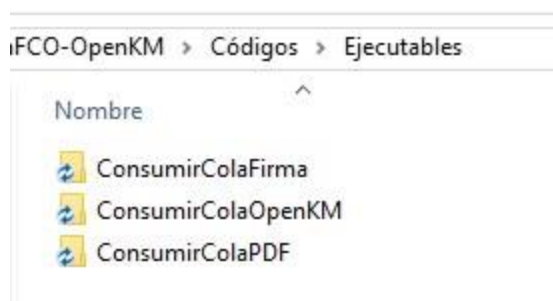


Figura 13. Códigos ejecutables

En la carpeta de códigos/Servicios, se encuentra el código fuente para modificar los servicios a gusto.

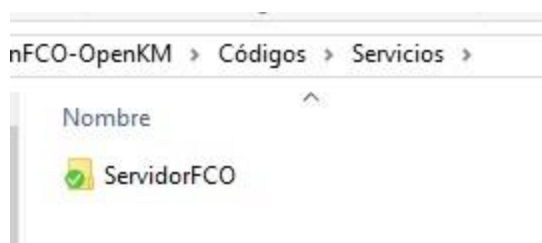


Figura 14. Código Servicios

10.1 Fuente – Servicios

En el archivo PDF.svc.cs se encuentra los parámetros de configuración para enviar las peticiones a cada cola.

```
try
{
    var factory = new ConnectionFactory() { HostName = "localhost", UserName = "guest", Password = "lavidaloca123" };
    using (var connection = factory.CreateConnection())
    using (var channel = connection.CreateModel())
    {
        channel.QueueDeclare(queue: "pdf",
                             durable: false,
                             exclusive: false,
                             autoDelete: false,
                             arguments: null);

        var body = Encoding.UTF8.GetBytes(json);
        channel.BasicPublish(exchange: "",
                             routingKey: "pdf",
                             basicProperties: null,
                             body: body);

        return true;
    }
}
catch (Exception ex)
```

Figura 15. Parámetros Cola - Servicios

10.2 Fuente – Cola FTP

Parámetros de conexión

En el archivo Program.cs se encuentran los parámetros de conexión para consumir la cola.

- **Hostname** = localhost
- **Username** = guest
- **Password** = lavidaloca123

Parámetros de cola

En el archivo Program.cs se encuentran los parámetros de conexión del servidor FTP y OpenKM.

Cada instancia pide los parámetros de conexión por defecto, también se puede enviar los datos de conexión a través del parámetro JSON. (Ver parámetros de **Servicios**)

```
OpenKM open = new OpenKM("http://demo.openkm.com/OpenKM", "user7", "pass7");
// (Se obtiene la conexión al servidor FTP)
FTP con = new FTP("54.94.172.182", "dvalladares", "dvalladares");
```

10.3 Fuente – Cola OpenKM

Parámetros de conexión

En el archivo Program.cs se encuentran los parámetros de conexión para consumir la cola.

- **Hostname** = localhost
- **Username** = guest
- **Password** = lavidaloca123

Parámetros de cola

En el archivo Program.cs se encuentran los parámetros de conexión a los servidores de OpenKM.

Cada instancia pide los parámetros de conexión por defecto, también se puede enviar los datos de conexión a través del parámetro JSON. (Ver parámetros de **Servicios**)

```
OpenKM open = new OpenKM("http://demo.openkm.com/OpenKM", "user7", "pass7");
```

10.4 Fuente – Cola Firma

Parámetros de conexión

En el archivo Program.cs se encuentran los parámetros de conexión para consumir la cola.

- **Hostname** = localhost
- **Username** = guest
- **Password** = lavidaloca123

Parámetros de cola

En el archivo Program.cs se encuentran los parámetros de conexión a los servidores de OpenKM.

Cada instancia pide los parámetros de conexión por defecto, también se puede enviar los datos de conexión a través del parámetro JSON. (Ver parámetros de **Servicios**)

```
OpenKM open = new OpenKM("http://demo.openkm.com/OpenKM", "user7", "pass7");
```

Parámetro firma imagen

En el archivo /clases/PDF.cs en el método add_firma, se encuentra la ruta para cambiar de ubicación la carpeta de la imagen.

Ubicación Actual:

C:\\Users\\Administrator\\Desktop\\ConsumirColaPDF\\imagen\\firma.jpg

```
public byte[] add_firma(byte[] documento, string aprobado_por)
{
    /* FIRMA */
    Document doc = new Document();
    MemoryStream fs = new MemoryStream();
    PdfWriter writer = PdfWriter.GetInstance(doc, fs);
    doc.Open();

    Paragraph paragraph = new Paragraph("Aprobado por: " + aprobado_por);
    string imageURL = "C:\\Users\\Administrator\\Desktop\\ConsumirColaPDF\\imagen\\firma.jpg";
    iTextSharp.text.Image jpg = iTextSharp.text.Image.GetInstance(imageURL);
    jpg.ScaleToFit(140f, 120f);
    jpg.SpacingBefore = 10f;
}
```

Figura 16. Ubicación ruta imagen