

# USO DE SOLUCIONES CLOUD PARA LA AUTOMATIZACIÓN DE PROCESOS

08/03/2020

Álex Sánchez Ortega

Informe Inicial

## Contenido

---

1.	Introducción.....	3
2.	Objetivo .....	4
2.1	Plataforma Web.....	5
2.2	Desarrollo API .....	5
2.3	Automatización de procesos con Logic Apps.....	6
2.4	Monitorización de errores .....	10
3.	Metodología .....	11
4.	Planificación.....	11
4.1	Inicio del Proyecto .....	12
4.2	Captación de requisitos .....	12
4.3	Preparación del entorno y diseño .....	12
4.4	Desarrollo y Test.....	13
4.5	Test y rendimiento.....	13
5.	Bibliografía .....	14

## 1. Introducción

---

La gestión de registros es una de las funciones imprescindibles en cualquier empresa u organización. A parte de ser obligatoria, da la posibilidad de tener un control exhaustivo de cualquier movimiento y acción dentro de la entidad. Este proceso aborda tres fases del ciclo de vida de los registros: la creación o recepción; el mantenimiento, el almacenamiento, la recuperación o el uso general; y la disposición. Sabiendo que en los registros reside todo lo que sucede en la empresa y son de gran importancia de cara a las entidades legislativas, los clientes y la gestión de la propia empresa, hay que asegurar un almacenamiento seguro de éstos.

Hoy en día, una empresa puede almacenar cientos de miles, o incluso millones, de registros y, más allá de las preocupaciones legales, debe haber un propósito adicional para su almacenamiento. Es decir, debe haber un enfoque que beneficie a la propia empresa y/o al cliente de alguna manera.

Partiendo de este punto, el proyecto está relacionado con dar una disposición a cualquier tipo de dato de los registros almacenados por una empresa que trabaja en la cadena de suministros. Desde el etiquetado de productos, la gestión de envíos, llegadas, facturas de pedidos, etc.

Antes de nada, para trabajar con la gestión de registros y dar un acceso a estos se debe concretar de donde salen éstos y cómo van a ser registrados.

Por otro lado, para tener un registro de estos datos, por parte de los usuarios, debe haber una herramienta que dé la posibilidad de registrarlos. En una cadena de suministros son varios los métodos para ejercer esta función, como por ejemplo, mediante registros físicos o, como es más común, con dispositivos móviles o plataformas web. Para éstas dos últimas, el sistema de almacenaje suele ser común, es decir, en bases de datos accedidas desde ambos dispositivos.

Para acabar, hay que señalar que estos sistemas, capaces de registrar cuantiosas cantidades de datos, deben tener una monitorización y un control del flujo de las acciones. Esto es lo que permitirá dar fiabilidad y consistencia a los datos registrados por los usuarios y evitar cualquier tipo de error que provoque su pérdida, duplicación, etc. Según el impacto de la entidad, un error inesperado puede significar millones de datos erróneos o perdidos, en cuestión de minutos.

## 2. Objetivo

---

El propósito del proyecto se divide en dos partes.

En primer lugar, crear una plataforma web que pueda mostrar los registros de datos almacenados a lo largo de una cadena de suministros. Esto es necesario para, posteriormente, crear flujos de trabajo automatizados que puedan tratar los datos y dar una disposición de ellos a los usuarios.

En segundo lugar, hacer un rastreo de los errores posibles que puede lanzar la plataforma web desarrollada y tener una monitorización exhaustiva de ellos.

El primer punto está enfocado en dar una funcionalidad a los usuarios finales para poder gestionar los datos de sus registros de forma eficiente y personalizada.

El segundo punto busca mejorar la eficiencia de la plataforma y darle una mejor fiabilidad.

## 2.1 Plataforma Web

Como se ha comentado anteriormente, hay varios medios para gestionar los registros y datos que se generan a lo largo de una cadena de suministros pero, al final, la forma más eficiente de hacerlo es mediante un aplicación o software que permita personalizar la disposición de éstos.

En este caso, se opta por la realización de una plataforma web, por su accesibilidad global, simultaneidad de usuarios y la disponibilidad de la información en un formato más visual, a diferencia de un dispositivo móvil. Por otro lado, éste último debe ser más eficiente a la hora de registrar los *datos in situ*, por ejemplo, a la hora de entregar un pedido.

A la hora de hacer el desarrollo de ésta, se ha optado por .NET principalmente por su soporte para múltiples lenguajes, el fácil desarrollo basado en componentes y la compilación *just-in-time*, que permite generar el código máquina propio de la plataforma, aumentando el rendimiento de la aplicación al ser específico para cada plataforma. Éste último aspecto es muy importante, ya que necesitamos una plataforma eficiente y óptima, capaz de acceder a muchos datos lo más rápido posible.

Como aliciente, cabe señalar, como se comenta más adelante, las soluciones de Microsoft Azure son compatibles con .NET.

## 2.2 Desarrollo API

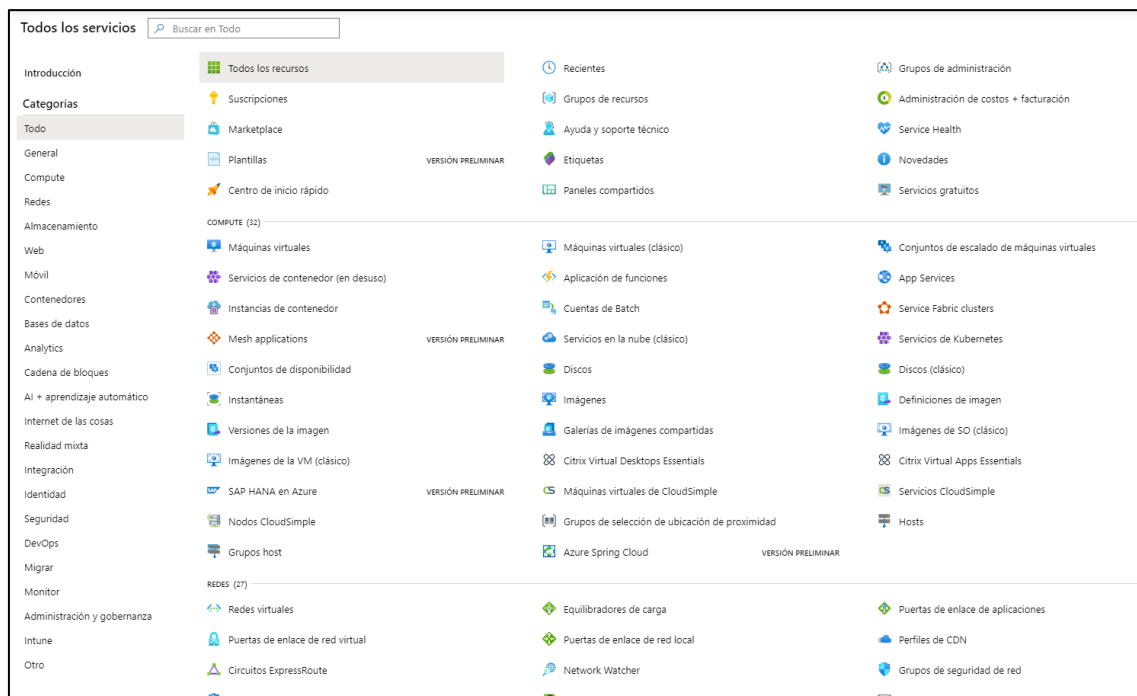
Para que sea posible la automatización de los procesos de la plataforma, se necesita una API personalizada que permita, abstraer funciones y ser ejecutadas desde llamadas externas.

Esta biblioteca de subrutinas debe poder realizar las funciones necesarias para que la plataforma sea eficiente y, a grandes rasgos, útil.

## 2.3 Automatización de procesos con Logic Apps

Como se ha mencionado anteriormente, en el proyecto se utilizarán recursos de Microsoft Azure. Éste es un conjunto de servicios de informática en la nube que se utilizan en una extensa cantidad de empresas y entidades. Ofrece mucha flexibilidad y globalización.

Entre sus recursos disponibles encontramos la posibilidad de crear máquinas virtuales, sistemas de almacenamiento, monitorización, migraciones, etc.



*Figura 1. Microsoft Azure Services*

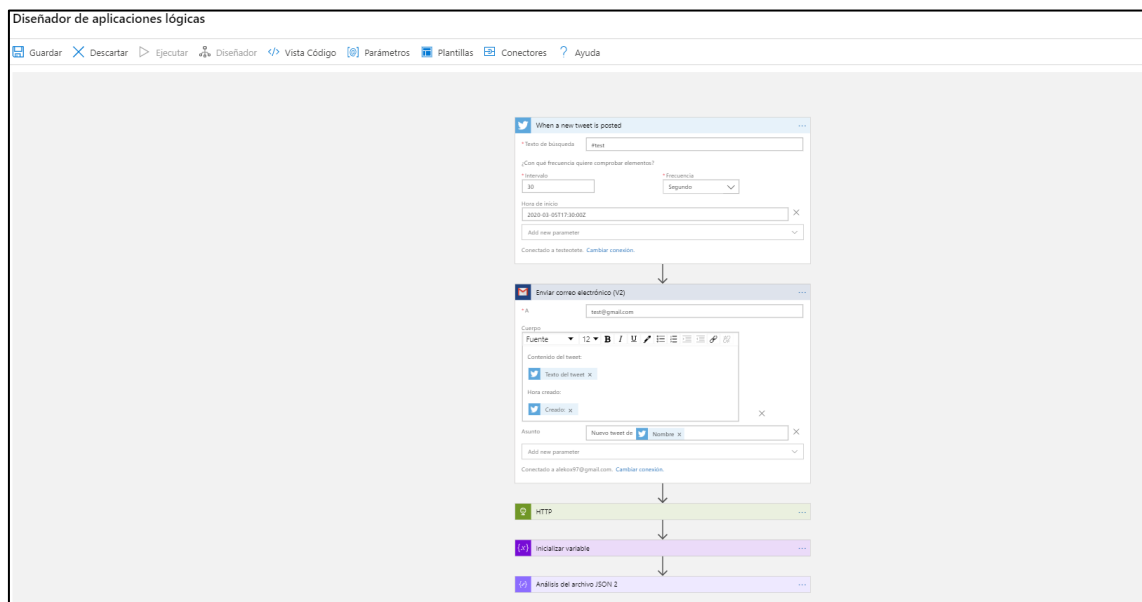
Cabe señalar que Microsoft Azure es de pago y que para ello se debe tener una suscripción y hacer una contratación de sus servicios, normalmente de pago por uso.

En el proyecto, principalmente, se trabajará con el servicio cloud *Logic Apps*.

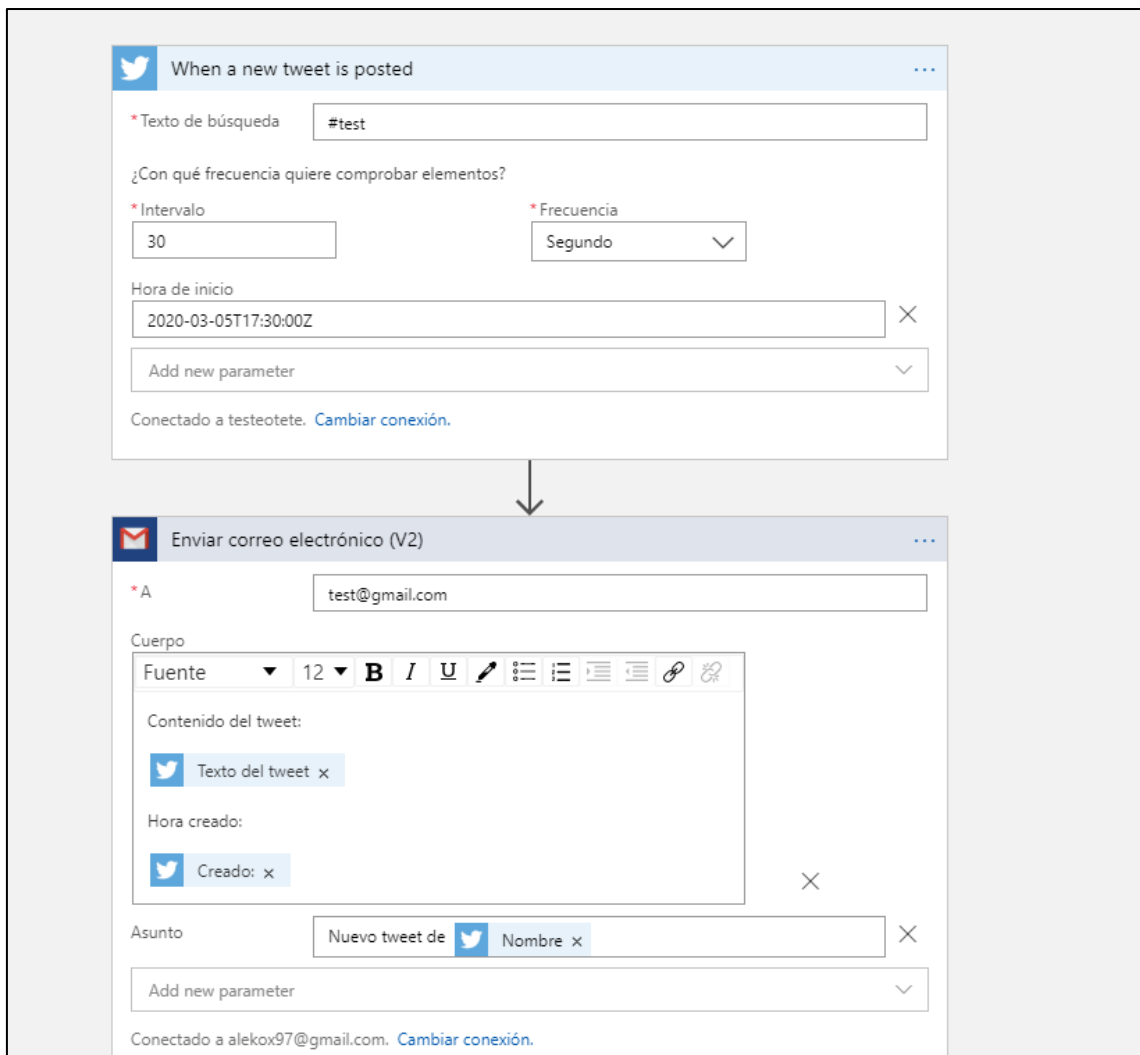
Éste permite automatizar y orquestar flujos de trabajo y procesos de una manera visual, organizada y simplificada.

Entre sus principales aplicaciones está la de llamar a APIs y realizar procesos mediante un desencadenador.

Por ejemplo, en el caso siguiente se aprecia un flujo de trabajo de ejemplo realizado para probar este tipo de servicio.



*Figura 2. Interfaz Logic Apps*



*Figura 3. Flujo de trabajo de ejemplo*

En la Figura 3, podemos apreciar dos cajas, donde cada una de ellas expresa una acción.

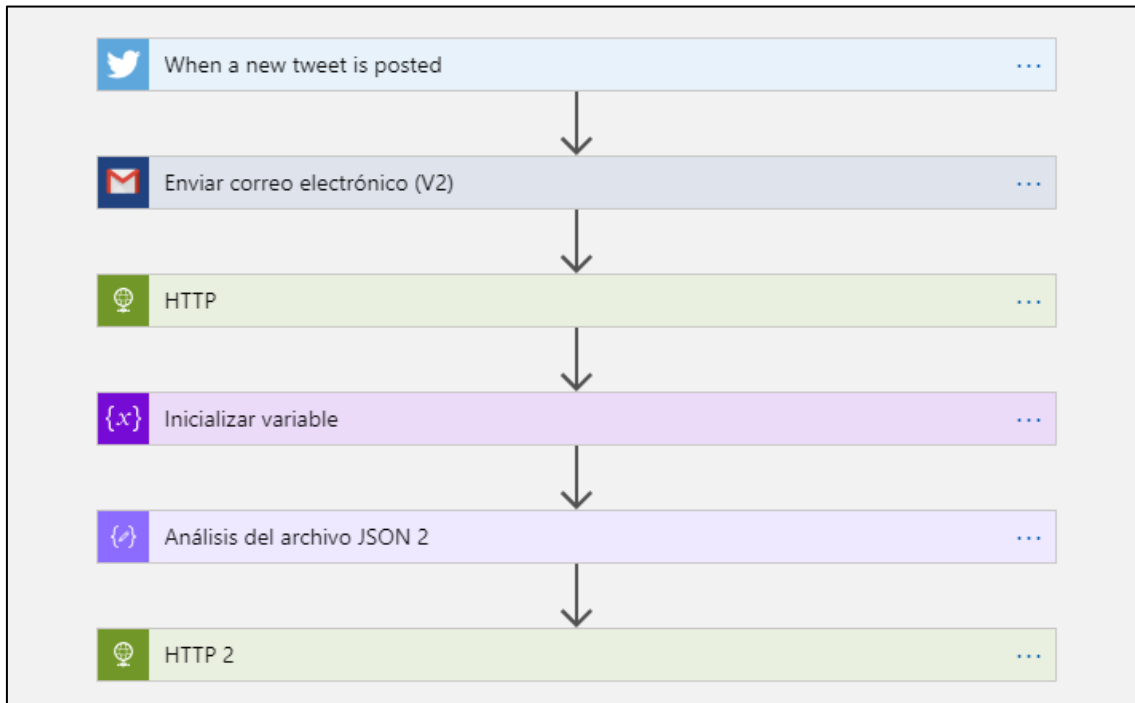
La primera de ellas es el desencadenador de la aplicación lógica, es decir, el **qué tiene que pasar para que se active**.

La segunda, es la acción que encadena el desencadenador.

En el ejemplo, se puede apreciar que (1) si se publica un tweet que contenga la etiqueta *#test* se activará la Logic App. Seguidamente, (2) se recuperarán datos del tweet desencadenante y se enviará un correo electrónico personalizado (segunda caja).



Como se puede ver es un flujo de trabajo muy simple y que puede ocupar dos minutos de nuestro tiempo.



*Figura 4. Flujo de trabajo ampliado de ejemplo*

En la Figura 4, se aprecia un flujo de trabajo más largo, en el que se añaden 4 cajas. Éstas son responsables de acceder a la API de Google, hacer una búsqueda del contenido del tweet en Google, formatear la respuesta de la API y enviarlo a otra API.

Después de exponer todo lo anterior, uno de los objetivos principales del proyecto es, mediante Logic Apps, recuperar datos de los registros de la plataforma web y ofrecerlos a los usuarios en un formato personalizado, usando plantillas en formatos de texto. Dar a los usuarios la posibilidad de recibir un resumen organizado de sus datos, a modo de suscripción Webhook.

## 2.4 Monitorización de errores

Por último tema a tratar, como se ha comentado en la introducción, un error en producción, dentro de un ámbito con gran impacto y con muchos usuarios, puede suponer la pérdida de miles de datos y tiempos de retraso irre recuperables. Por eso es necesario un control de errores y una monitorización de estos.

El propósito de este punto es dar la posibilidad al equipo de desarrollo de asegurar un mantenimiento eficiente y poder ofrecer, mediante Logic Apps, un tipo de suscripción de notificaciones para cualquier tipo de error desencadenante de las acciones de la plataforma web.

Por ejemplo, en los días de subida a producción, tener un control más exhaustivo de cualquier tipo de error que puede surgir y así poder actuar cuanto antes para solucionar el problema.

### 3. Metodología

---

Para el desarrollo del proyecto se empleará una metodología de trabajo *Waterfall*. Es decir, el desarrollo será de manera secuencial, por fases, comenzando por un análisis, diseño y terminando con el testeo y puesta en producción.

Las ventajas que supone este tipo de metodología son (1) la capacidad de medir de forma más fácil el progreso del proyecto, (2) la planificación es más sencilla, (3) solo hay una persona implicada en el desarrollo y (4) se deben desarrollar varios componentes de software de manera paralela.

Las posibles desventajas de usar esta metodología están relacionadas con las primeras fases, de captación de requisitos y diseño.

### 4. Planificación

---

La fecha de finalización del proyecto ha sido marcada para finales de mayo.

Es decir, para entonces debe haber una solución funcional y testeada del conjunto del proyecto.

Para cumplir con los términos marcados, el proyecto se ha dividido en dos partes:

- Fase de desarrollo de Plataforma Web y API
- Fase de automatización de procesos y monitorización con Logic Apps.

La primera de ellas ha sido marcada con un deadline **el 19 de abril**.

La segunda parte debe estar finalizada un mes después, **el 20 de mayo**.

De esta manera se dispondrá de dos semanas, aproximadamente, para hacer las pruebas finales de rendimiento.

Las últimas semanas antes de la entrega final servirán para finalizar el informe final, realizar el artículo y preparar la exposición.

#### 4.1 Inicio del Proyecto

La primera fase del proyecto es la responsable de dar una idea concisa del trabajo a realizar, valorar las capacidades y límites del proyecto y dar una visión concreta del rumbo a seguir.

**Fecha final: 08-03-2020**

#### 4.2 Captación de requisitos

Para captar los requisitos, se tendrá que realizar una investigación para concretar y responder varias preguntas como por ejemplo: qué necesita la plataforma web, cómo son las plataformas web similares, qué tipo de API se necesitará, qué tipo y cuántos recursos habrá que contratar en Azure, qué modelo de programación se seguirá, etc.

**Fecha final: 13-03-2020**

#### 4.3 Preparación del entorno y diseño

Una vez respondidas casi todas las preguntas anteriores o todas las imprescindibles, hay que decidir cómo será la plataforma web, qué tipo de diseño tendrá, ...

**Fecha final: 21-03-2020**

#### 4.4 Desarrollo y Test

Se desarrollará mediante TDD, es decir en todo momento se deberá probar que lo que se desarrolla cumplirá con las especificaciones expuestas.

Para ello, en primer lugar habrá que crear el entorno web, y, paralelamente, crear la API responsable de realizar las acciones.

Por último, se abordará el tema principal del proyecto, las Logic Apps.

**Fecha final: 20-05-2020**

#### 4.5 Test y rendimiento

Para finalizar el desarrollo, habrá que realizar las pruebas de rendimiento finales para saber si la plataforma web es eficiente, en tiempo y consumo.

Puesto que la plataforma es una simulación de un caso real en el que deberemos de acceder a miles de datos, habrá que tener especial cuidado con las consultas a las bases de datos y controlar los tiempos.

**Fecha final: 30-05-2020**

## 5. Bibliografía

---

- Portal de Microsoft Azure [1]  
<https://portal.azure.com/#home>
- Información acerca de Azure [2]  
<https://azure.microsoft.com/es-es/services/logic-apps/>
- Documentación Logic Apps [3]  
<https://docs.microsoft.com/es-es/azure/logic-apps/logic-apps-overview>
- Metodología Waterfall vs Agile [4]  
<https://thedigitalprojectmanager.com/es/agile-frente-a-waterfall/>  
<https://www2.deloitte.com/es/es/pages/technology/articles/waterfall-vs-agile.html>  
<https://www.esan.edu.pe/apuntes-empresariales/2017/05/proyectos-en-metodologia-waterfall-o-agile-hacia-donde-vamos/>
- Información de Metodología Waterfall [5]  
<http://design-toolkit.recursos.uoc.edu/es/waterfall/>  
[Recursos de la asignatura de Gestión de Proyectos y Ingeniería del Software]
- Gestión de recursos [6]  
<https://www.wrike.com/es/blog/que-es-la-gestion-de-recursos-y-por-que-es-importante/>  
<https://www.ceupe.com/blog/la-gestion-de-los-recursos.html>