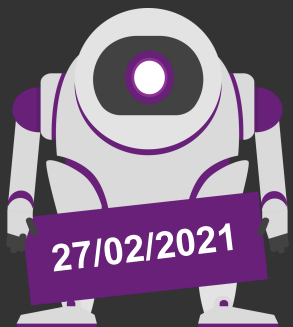


2021

# Netcoreconf

Workflow y automatizaciones  
con Logic Apps Preview



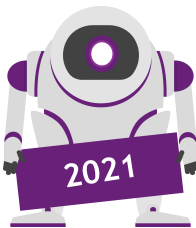
**Raúl Fernández de Córdoba**

*Cloud Solutions Architect*

@rfcm83

<https://setvalue.net>

# SPONSORS





@rfcm83



linkedin.com/in/rfcm83



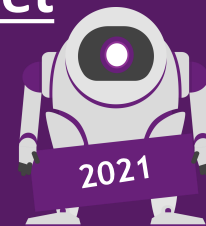
www.setvalue.net



RFernandez@kabel.es

Mi nombre es Raúl Fernández de Córdoba, trabajo en Kabel como Cloud Solutions Architect. En mi día a día, trato de asesorar y definir las mejores estrategias que nos permitan resolver los retos que encontramos en los proyectos.

Soy un apasionado del desarrollo Cloud, en mi tiempo libre me gusta disfrutar de mi familia, mis amigos, mi moto y de un buen partido de fútbol. Cuando puedo, intento escribir en <https://setvalue.net>

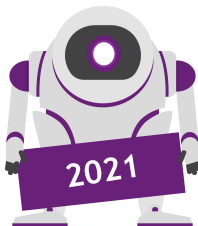


#netcoreconf

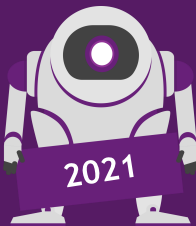
# Agenda

---

- Azure Logic Apps
  - Overview
  - Modelo de facturación
  - Desarrollo
  - Herramientas
  - Ejemplo
- Azure Logic Apps Preview
  - Overview
  - Novedades
  - Stateful y Stateless
  - Modelo de facturación
  - Limitaciones
    - Servicio
    - Visual Studio Code
  - Requisitos
- Demo

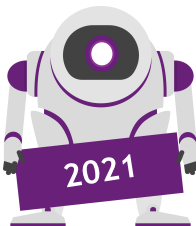


# Azure Logic Apps (Classic)



# Azure Logic Apps: Overview

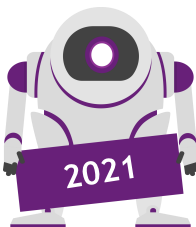
- Se trata de un servicio cloud que simplifica el diseño y la construcción de soluciones escalables, que permiten automatizar, orquestar o programar tareas y workflows, integrando aplicaciones, datos, sistemas o servicios entre empresas
- Un workflow es un conjunto de acciones, que combinadas, forman un flujo operacional que permite realizar transformaciones o movimientos de datos, generar alertas o comunicar sistemas y aplicaciones
- Cada workflow tiene mínimo un trigger que desencadena la ejecución del flujo. Estos pueden ser recurrencias periódicas, producidos por un evento o por una simple llamada http



# Azure Logic Apps: Modelo de facturación

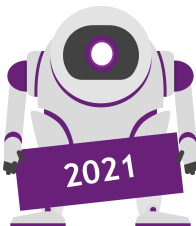
Su precio viene determinado por el número y tipo de acciones que se utilizan en cada ejecución

	PRICE PER EXECUTION
Actions	0,000022 €
Standard Connector	0,000106 €
Enterprise Connector	0,000844 €



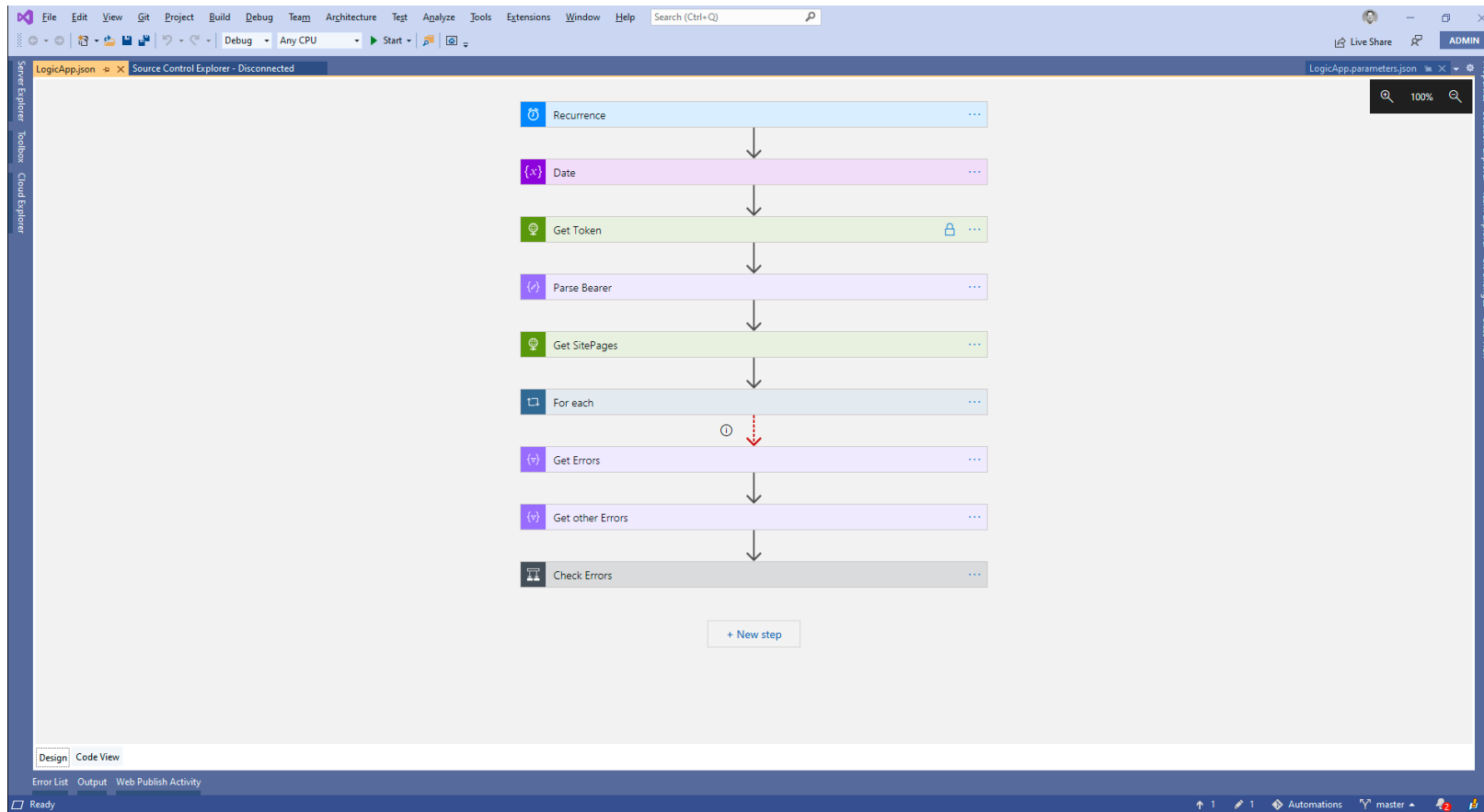
# Azure Logic Apps: Desarrollo

- La definición del workflow es una plantilla ARM que podemos editar desde Visual Studio 2019, Visual Studio Code o desde Portal Azure
- Visual Studio 2019 y Portal Azure permiten la edición de las Logic Apps de forma gráfica y modificando el JSON del ARM
- Visual Studio Code sólo permite visualizar en modo gráfico, para modificar la Logic App se debe editar el ARM
- Cada nueva publicación de cambios en el diseño del workflow, el servicio Azure Logic Apps almacena una versión de la definición, permitiendo al desarrollador promocionar un estado anterior





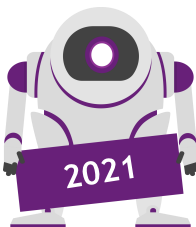
# Azure Logic Apps: Herramientas



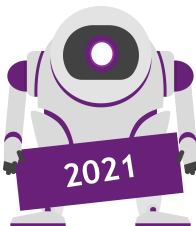
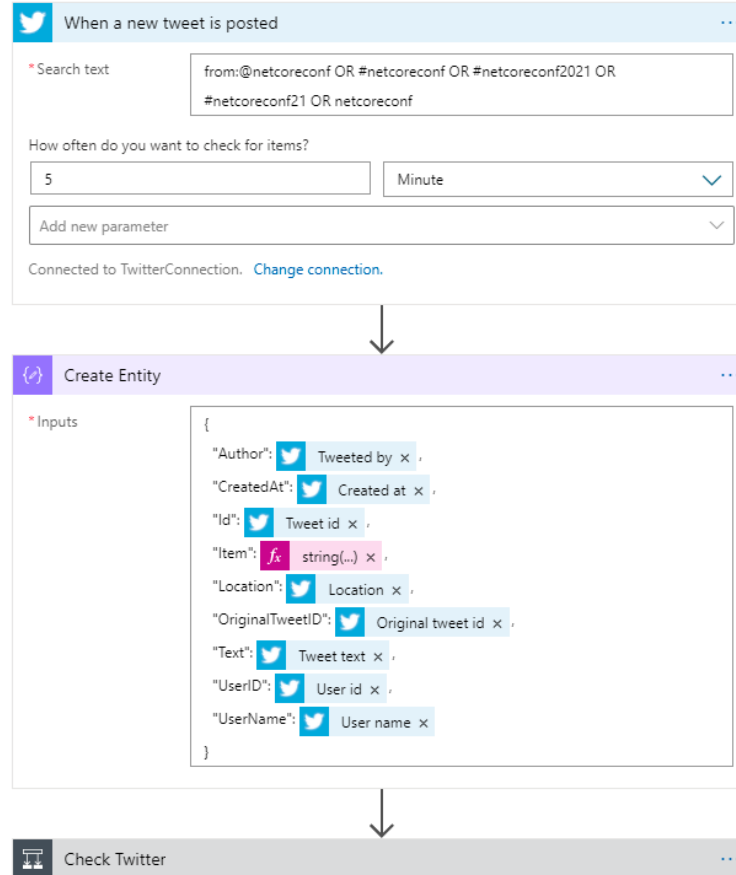
[Azure Logic Apps Tools for Visual Studio 2019](#)

[Azure Logic Apps for Visual Studio Code](#)

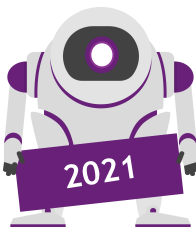
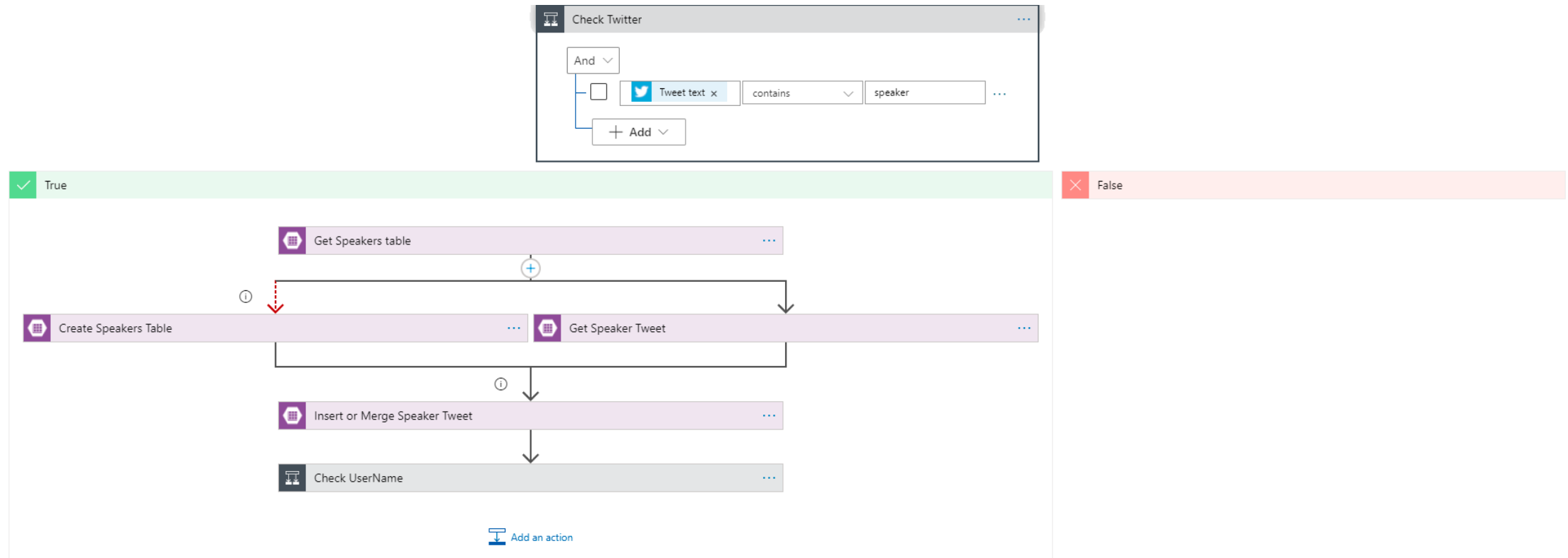
[Logic apps - Microsoft Azure](#)



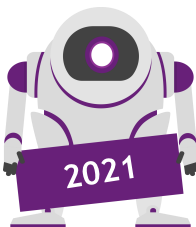
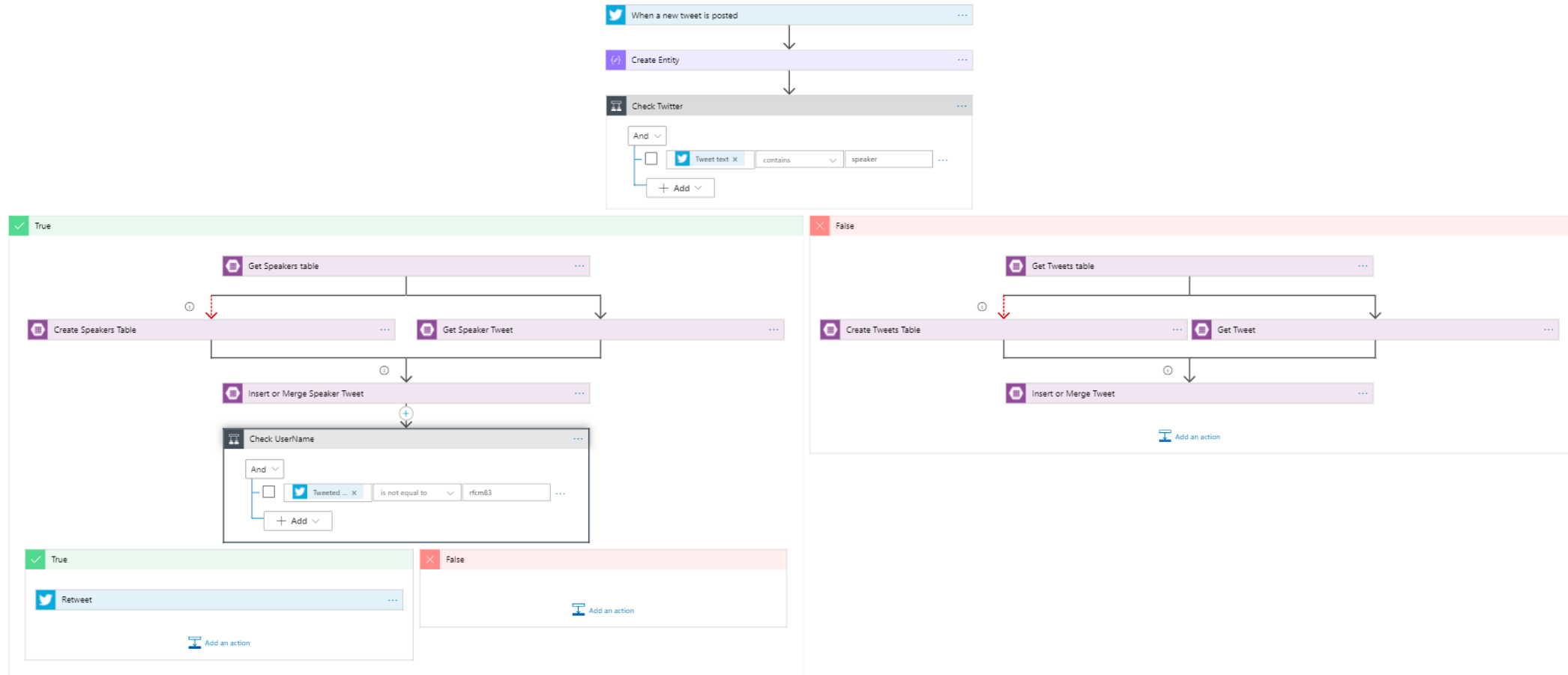
# Azure Logic Apps: Ejemplo



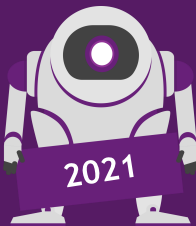
# Azure Logic Apps: Ejemplo



# Azure Logic Apps: Ejemplo



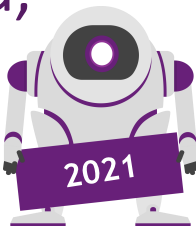
# New Azure Logic Apps (Preview)



#netcoreconf

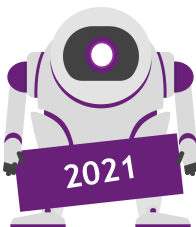
# Azure Logic Apps (Preview): Overview

- Este nuevo servicio, la novedad fundamental que nos trae, es el nuevo *runtime* que le proporciona portabilidad, mejor rendimiento y flexibilidad para implementar y ejecutar workflows no sólo en Azure, también en contenedores Docker
- Se agrega un nuevo concepto de automatizaciones con y sin estado
- Nuevo modelo de facturación basado en App Service Plan
- Actualmente, existen muchas incertidumbres en la compatibilidad, capacidades y el precio del servicio



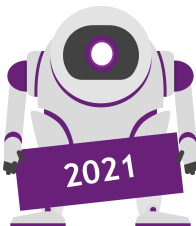
# Azure Logic Apps (Preview): Novedades

- Por defecto, cada Logic App tiene su propia System Managed Identity habilitada para autenticar conexiones en tiempo de ejecución
- El servicio genera SAS para que las Logic Apps puedan utilizarlas en tiempo de ejecución para hacer requests a los endpoints de las conexiones on cloud
- Logic Apps con flujos sin estados (stateless) que se ejecutan en memoria, lo que supone que finalicen y respondan más rápido, tengan un mayor rendimiento y tengan un coste menor porque por defecto no se almacenan los historiales de ejecución ni los datos entre acciones
- Ejecutar, testear y depurar localmente los workflows en Visual Studio Code
- Monitorización mediante Application Insights



# Azure Logic Apps (Preview): Stateless y Stateful

- **Stateful:**
  - Guardan los inputs / outputs de cada acción y sus estados en un almacenamiento externo, lo que permite revisar detalles de la ejecución y del historial
  - Tienen una alta resiliencia si se producen interrupciones, y tras la recuperación del servicio se puede retomar o volver a ejecutar el workflow hasta completarse
  - Pueden estar ejecutándose de forma prolongada hasta máximo un año
- **Stateless:**
  - Guardan los inputs / outputs de cada acción y sus estados en memoria
  - Menor resiliencia, si se producen interrupciones se debe volver a desencadenar la ejecución de forma manual
  - Tienen ejecuciones más cortas que no superan los 5 minutos, con un mejor rendimiento y menor coste
  - Solo se pueden ejecutar de forma síncrona





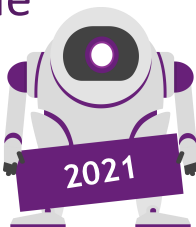
# Azure Logic Apps (Preview): Modelo de facturación

A diferencia de las Logic Apps clásicas que tienen un consumo por uso, para desplegar un workflow, en este nuevo servicio se requiere seleccionar un plan de hosting\*:

- App Service
- Premium

Además, los flujos de tipo stateful utilizan un almacenamiento externo, por lo que se requiere desplegar adicionalmente una Storage Account

\*El nuevo servicio de Azure Logic Apps todavía está en preview, cuando se libere, es posible que incluyan costes adicionales



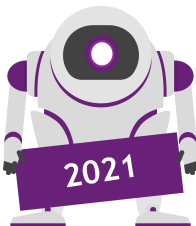
# Azure Logic Apps (Preview): Limitaciones servicio

## Triggers y Actions:

- Algunos triggers del servicio clásico no están soportados
- On-premises data Gateway triggers no está disponible, aunque sí las actions
- Azure Function action únicamente soporta la plantilla HTTP Trigger Function
- Algunos triggers y actions B2B para cuentas de integración no están disponibles
- Varias actions de caja han sido renombradas

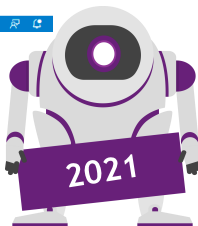
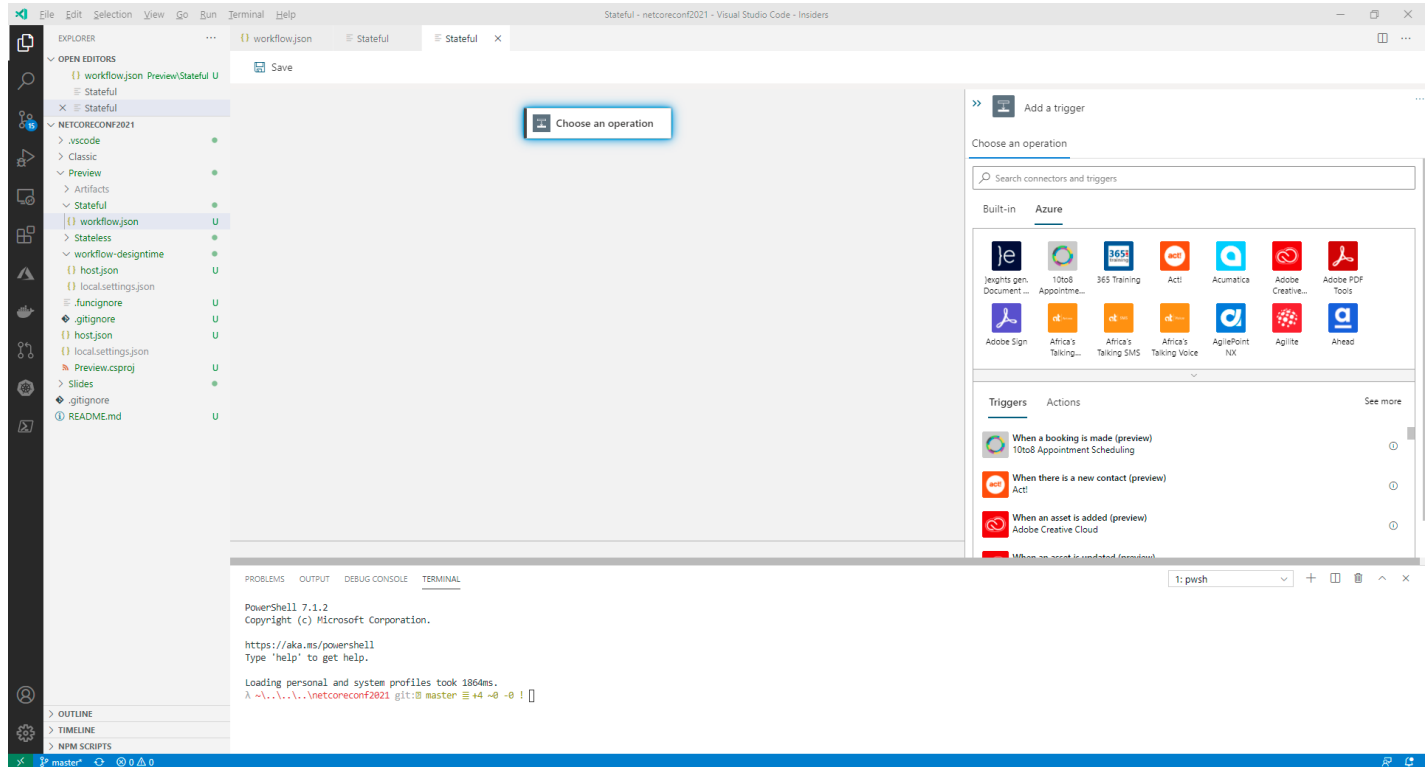
## Hosting

- De momento sólo es compatible con los planes App Service y Premium
- Se puede desplegar en Docker container, pero no está soportado ISE



# Azure Logic Apps (Preview): Limitaciones VS Code

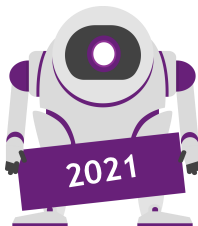
- El designer no funciona en VS Code for Linux OS
- No se pueden añadir ramas paralelas, aunque sí se visualizan si ya están creadas
- Tampoco se dispone de control de zoom
- Se pueden añadir breakpoints únicamente a las acciones, los triggers todavía no lo soportan



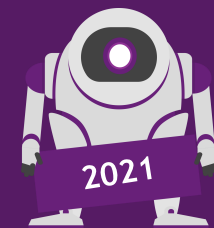
# Azure Logic Apps (Preview): Requisitos

Para poder desarrollar Logic Apps en este nuevo servicio necesitamos:

- Almacenamiento
  - Azure Storage Emulator 5.1.0
  - SQL Server 2019 Express Edition
- Herramientas
  - Visual Studio Code 1.30.1 o superior
  - Extensión Azure Account
  - C# for Visual Studio Code
  - Azure Functions Core Tools 3.0.2931 o superior
  - Azure Logic Apps (Preview) extension for Visual Studio Code
  - ngrok

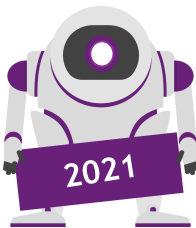


# Demo



#netcoreconf

# SPONSORS





More information:  
[info@netcoreconf.com](mailto:info@netcoreconf.com)  
@Netcoreconf

Visit on:  
[netcoreconf.com](http://netcoreconf.com)