

Saturnino Pimentel



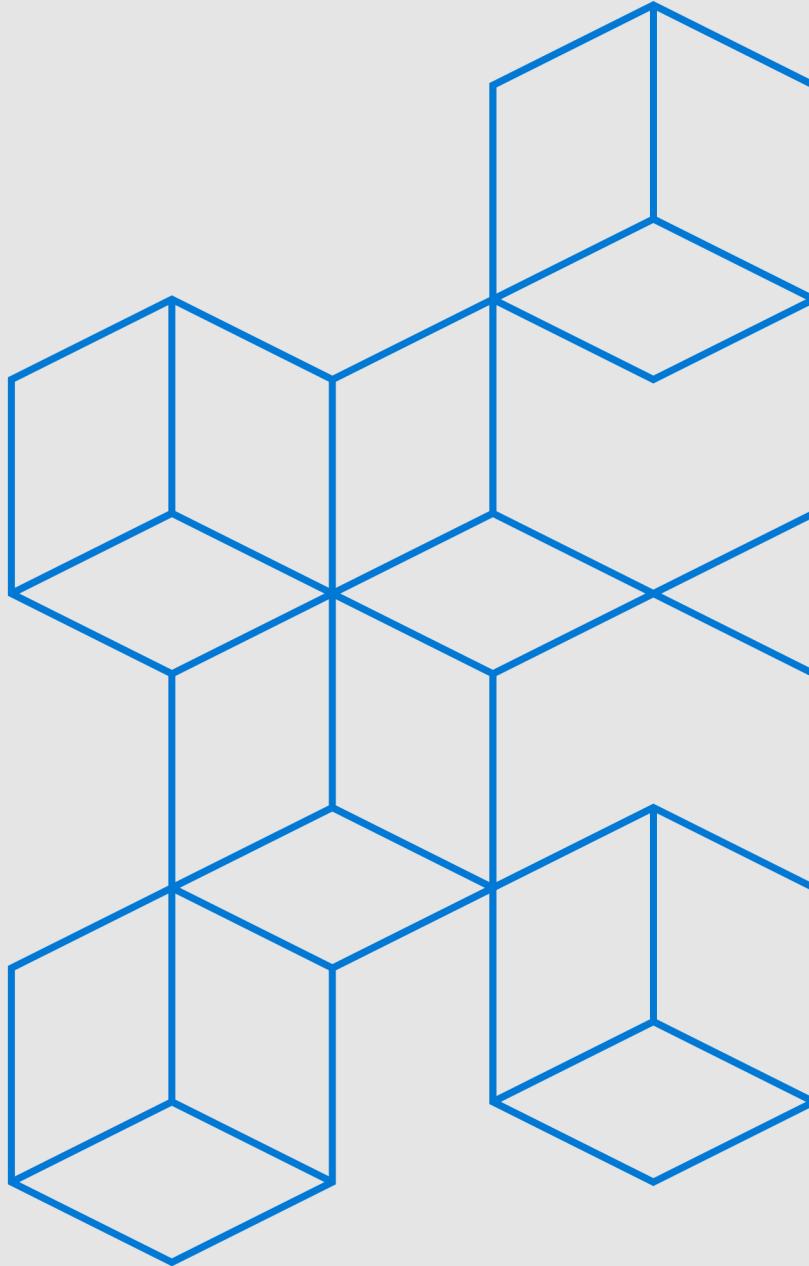
@saturpimentel



saturninopimentel



Serverless con Azure Functions



¿Qué es Serverless?

Serverless es un modelo de ejecución de computación en la nube en el cuál el proveedor de la nube administra dinámicamente la asignación de los recursos de la máquina. El precio se basa en la cantidad real de recursos consumidos por una aplicación.

https://en.wikipedia.org/wiki/Serverless_computing

Sí existe un servidor

Bajo el modelo de ejecución **Serverless** se debería utilizar **PaaS** tanto como sea posible.

Azure Functions

Microsoft nos permite utilizar el enfoque **Serverless** al correr nuestro código personalizado en **Azure Functions** respondiendo a eventos, algo que también es conocido como **Functions as a Service**.

Azure Functions esta basado en los web Jobs.

Sistemas de facturación

Servicio dedicado

Plan de consumo

Plan de consumo Premium

Para mayor información de costos

<https://azure.microsoft.com/es-mx/pricing/details/functions/>

Beneficios

- Centrarte en las necesidades de tu proyecto.
- Evita código repetido.
- Desarrollo más rápido.
- Están basados en Azure Web Apps y te da acceso a muchas de sus características.
- Escalado automático.
- Bajo el plan de consumo tienes un costo más eficiente.
- Tiene soporte para múltiples lenguajes.
- Puedes compartir recursos y configuraciones entre funciones.

¿Desventajas?

- Aumenta la complejidad
- La curva de aprendizaje.
- Manejo de versiones.
- Compartir código entre diferentes funciones.
- Es difícil hacer debug del código.

¡Azure Functions 3.0 ahora disponible!

Enero 2020

Se puede ejecutar .Net Core 3.1 y Node 12.

Es altamente compatible con versiones anteriores. (Es recomendable realizar pruebas exhaustivas).

<https://azure.microsoft.com/es-mx/updates/azure-functions-runtime-30-is-now-available/>

Elementos de una Function

Los **trigger** como el nombre lo dice son los disparadores de la funcionalidad dentro de nuestra **Azure**

Function pueden ser de diferentes tipo timers, HTTP request, mensajes en una cola, etc.

Los **bindings** de entrada/salida proporcionan una forma declarativa de conectarse a los datos desde dentro del código, los bindings son opcionales y las funciones pueden tener varios en su declaración

<https://docs.microsoft.com/es-mx/azure/azure-functions/functions-triggers-bindings#supported-bindings>

Lenguajes soportados

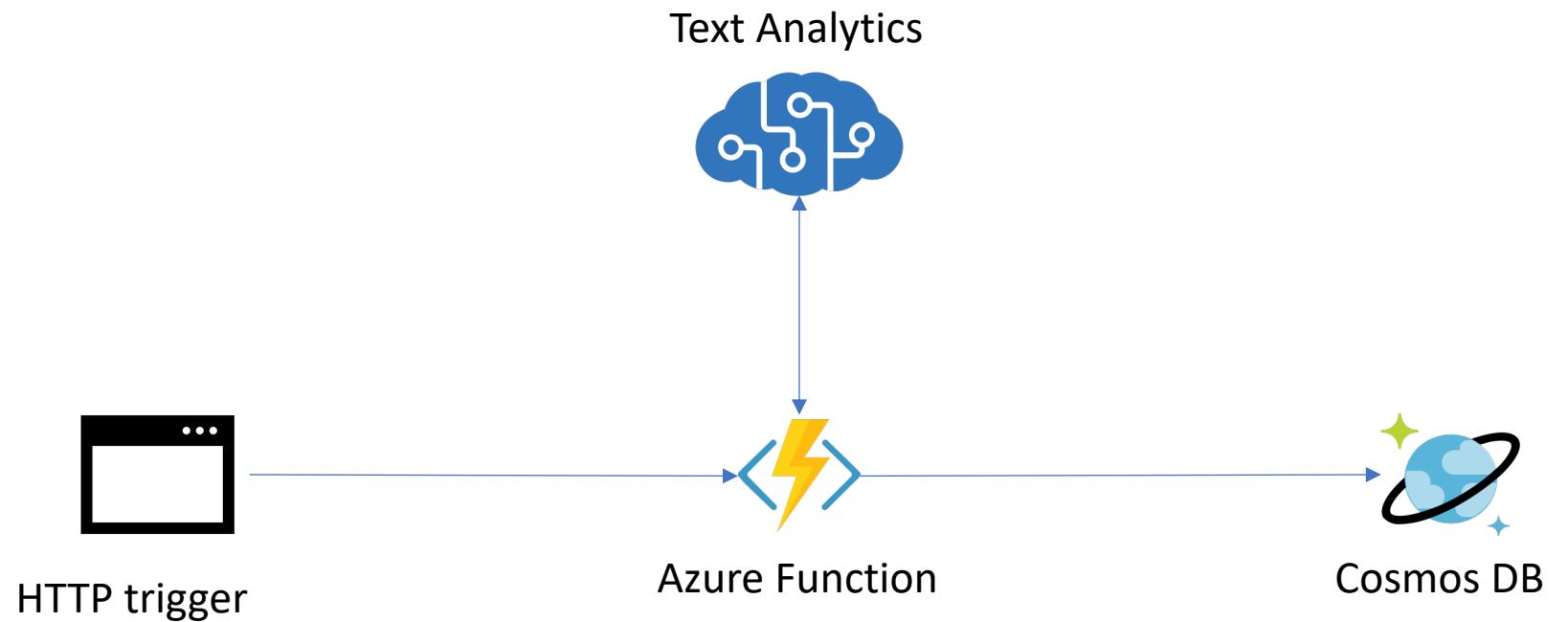
Idioma	1.x	2.x	3.x
C#	Disponibilidad general (.NET Framework 4.7)	Disponibilidad general (.NET Core 2.2)	Disponibilidad general (.NET Core 3.1)
JavaScript	Disponibilidad general (nodo 6)	Disponibilidad general (nodos 8 y 10)	Disponibilidad general (nodos 10 y 12)
F#	Disponibilidad general (.NET Framework 4.7)	Disponibilidad general (.NET Core 2.2)	Disponibilidad general (.NET Core 3.1)
Java	N/D	Disponibilidad general (Java 8)	Disponibilidad general (Java 8)
PowerShell	N/D	GA (PowerShell Core 6)	GA (PowerShell Core 6)
Python	N/D	Disponibilidad general (Python 3.6 y 3.7)	Disponibilidad general (Python 3.6, 3.7 y 3.8)
TypeScript	N/D	Disponibilidad general ¹	Disponibilidad general ¹

¹Se admite mediante la transpilación de JavaScript.

<https://docs.microsoft.com/en-us/azure/azure-functions/supported-languages#languages-by-runtime-version>



Desarrollo





iAzure functions es open source!

The screenshot shows three GitHub repository pages for Azure Functions components:

- Azure / azure-functions-core-tools**: Watched by 170 users, 547 stars, 181 forks.
- Azure / azure-functions-ux**: Watched by 58 users, 149 stars, 59 forks.
- Azure / azure-functions-host**: Watched by 114 users, 427 stars, 153 forks. This is described as "The host/runtime that powers Azure Functions" with a link to <https://functions.azure.com>.

At the bottom, there is a summary bar with the following metrics:

- 1,380 commits
- 53 branches
- 87 releases
- 36 contributors
- MIT license

Action buttons at the bottom include: Branch: dev ▾, New pull request, Create new file, Upload files, Find file, and Clone or download ▾.

Gracias!



@saturpimentel

