Creación de una “Azure Function” utilizando Visual Studio Code y Java.

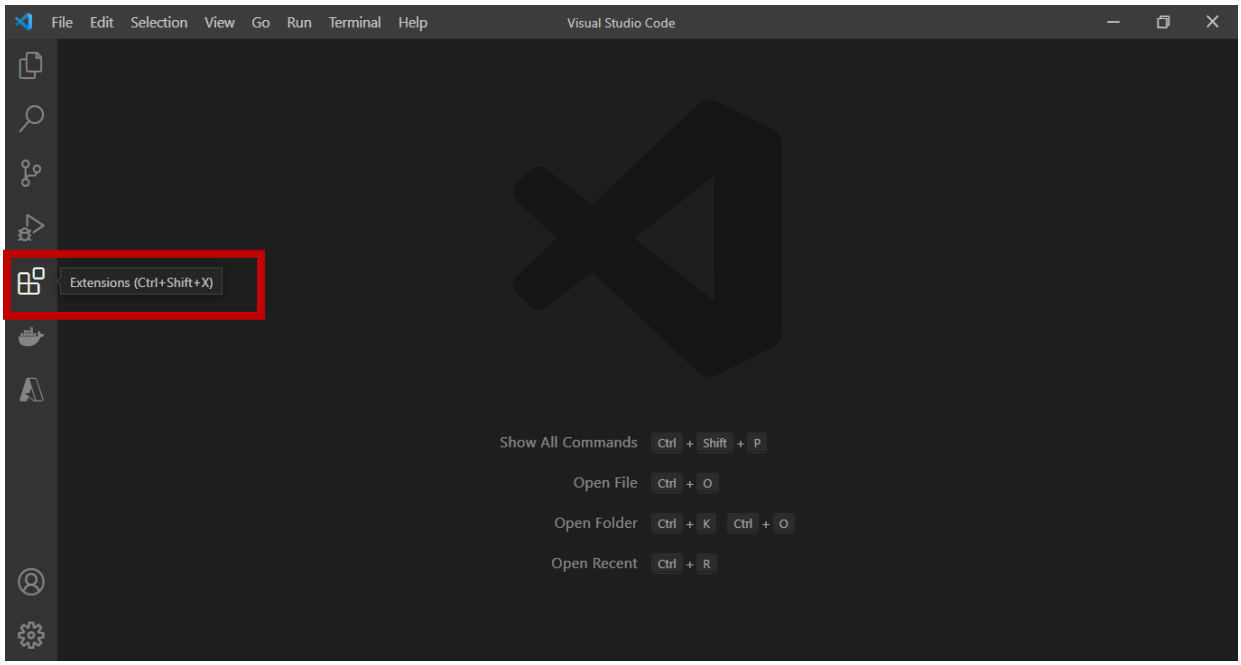
Requisitos Previos:

- Programa Visual Studio Code
- Cuenta en Azure con una suscripción activa.
- Extensión de Azure para Visual Studio Code
- Extensión de Java para Visual Studio Code
- Java Development Kit versión 8 u 11
- Apache Maven 3.0 o posterior

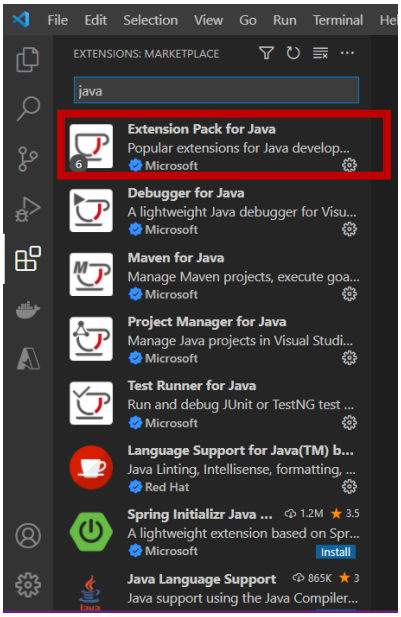
Instrucciones:

Teniendo el programa Visual Studio Code, Maven y el JDK instalado, abrimos Visual Studio Code.

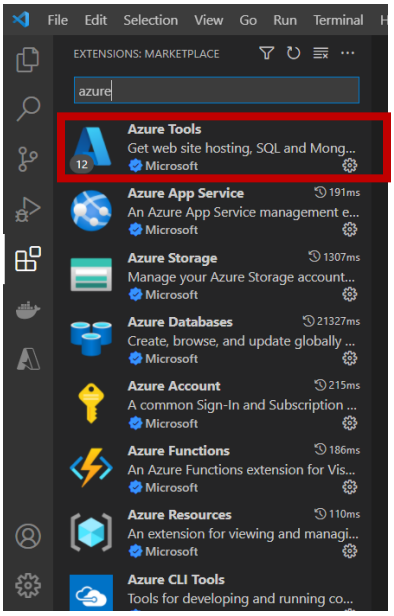
Nos dirigiremos a la barra de opciones ubicada en el lateral izquierdo de la pantalla y seleccionamos la opción de extensiones o acceder mediante el comando Crtl + Shift + X.



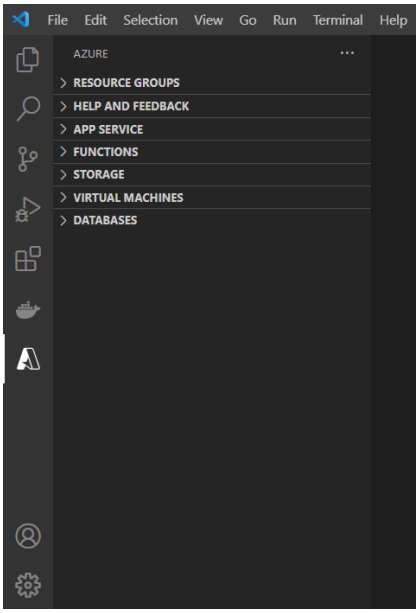
Escribimos en el buscador la palabra “Java” e instalamos Extension Pack for Java. Esta extensión nos ayudará a obtener en Visual Studio Code las herramientas de desarrollo de Java necesarias para poder crear pruebas, correr aplicaciones o debuggearlas directamente en el entrono de desarrollo.



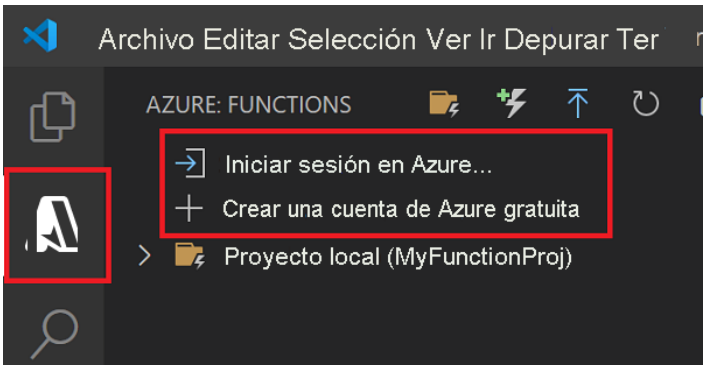
De mismo modo, buscamos la palabra “Azure” en el buscador del Marketplace e instalamos la extensión Azure Tools. Esta extensión nos ayudará a obtener todas las extensiones para Visual Studio Code disponibles para trabajar con Azure.



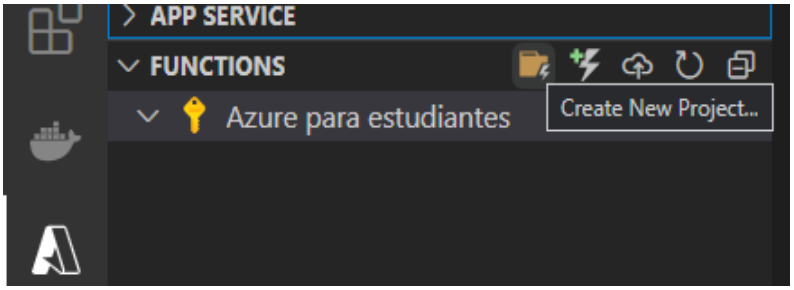
Una vez instaladas ambas extensiones, nos aparecerá en la barra de actividades ubicada en el lateral izquierdo una nueva opción con el logotipo de Azure. En esta opción veremos una serie de secciones que hacen referencia a los servicios de Azure con los que podremos trabajar desde Visual Studio Code.



En la sección de Functions, tenemos la opción de Iniciar sesión en Azure para que podamos tener acceso a todos los recursos creados en la suscripción de Azure seleccionada, así como poder publicar los proyectos locales a nuestro entorno de Azure. Si no disponemos de una suscripción a Azure, tenemos la opción de crear una cuenta con suscripción. En ambas opciones, seremos redirigidos al navegador predeterminado para colocar las credenciales de inicio de sesión en Azure o bien a la página de registro para cuantas nuevas.

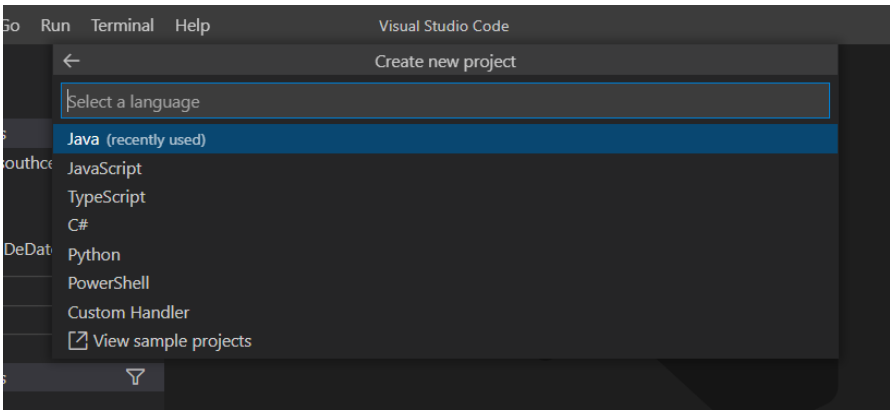


Una vez iniciada sesión, podremos ser capaces de visualizar los recursos creados en Azure y el nombre de la suscripción seleccionada. Para la creación de una nueva función, tendremos que crear un proyecto nuevo de forma local, para ello seleccionamos el botón con el ícono de un folder y un rayo. Nos pedirá seleccionar un folder donde se creará el proyecto.

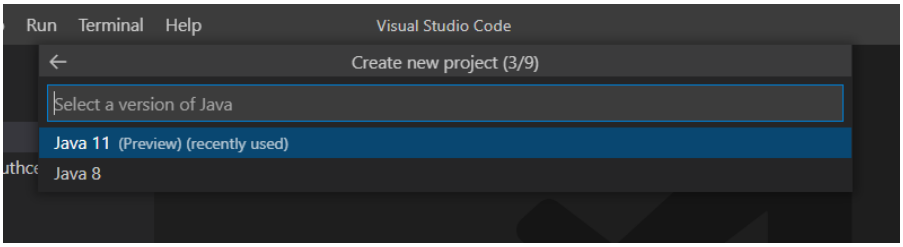


A continuación, Visual Studio Code nos hará una serie de cuestionamientos para conocer el tipo de proyecto que creará.

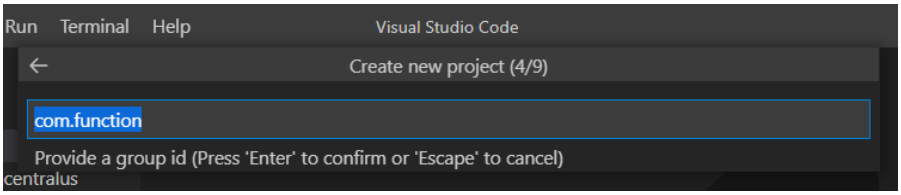
Primero nos preguntará el lenguaje de programación con el que estaremos trabajando, para nuestro ejemplo seleccionaremos el lenguaje de programación Java.



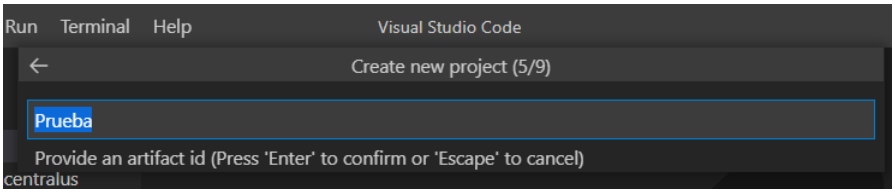
Después debemos seleccionar la versión de Java con la que estaremos desarrollando. Esta versión debe de ser la misma que la versión de JDK instalada previamente.



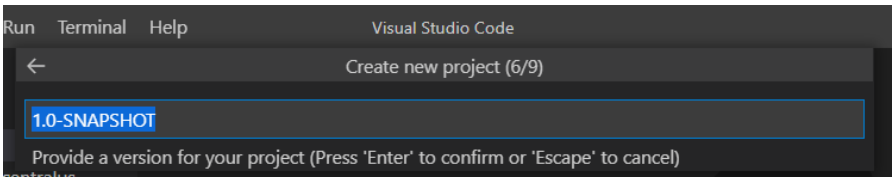
Proveemos un Group id para garantizar la singularidad el proyecto, este puede ser definido por el usuario o utilizar el sugerido por el programa. Por lo general su estructura se conforma de un dominio seguido del nombre de la empresa o creador separados por un punto y en minúsculas.



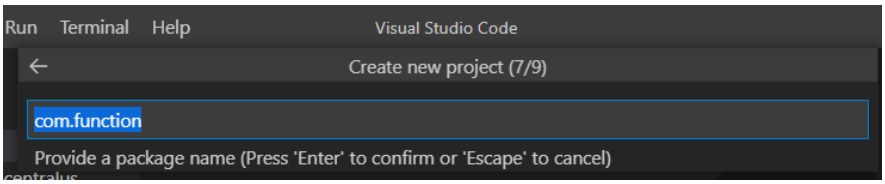
Continuamos con la definición de Artifact id, el cual será el nombre que acompañará al Group id para identificar fácilmente el proyecto. Como sugerencia se nos propondrá el nombre de la carpeta que seleccionamos como el sitio de creación del proyecto.



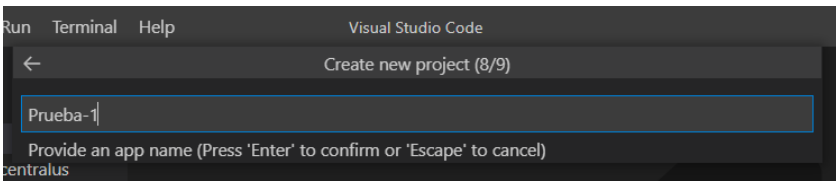
Debemos definirle a nuestro proyecto un número de versión con la finalidad de separar este desarrollo de pasadas o futuras versiones del mismo proyecto.



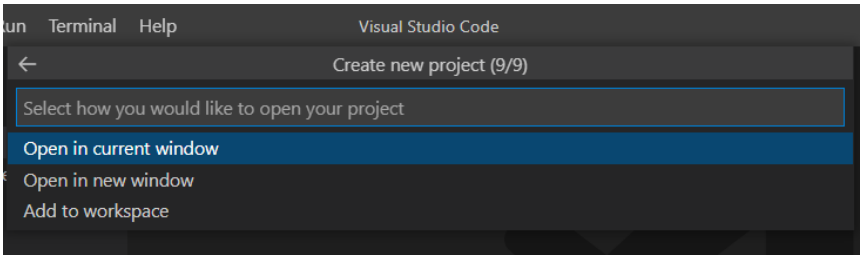
Generamos un nombre de paquete que será la residencia de nuestras clases Java.



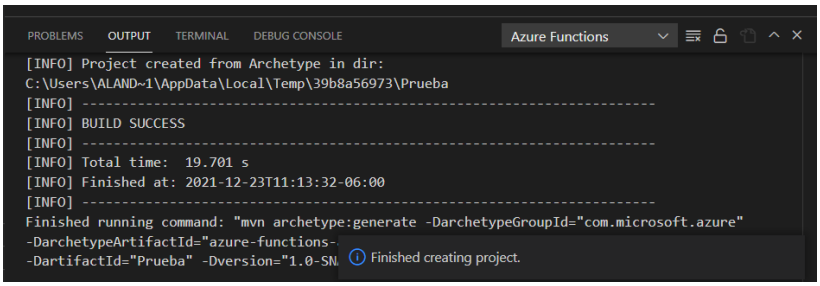
Por último, le definiremos un nombre a nuestra aplicación.



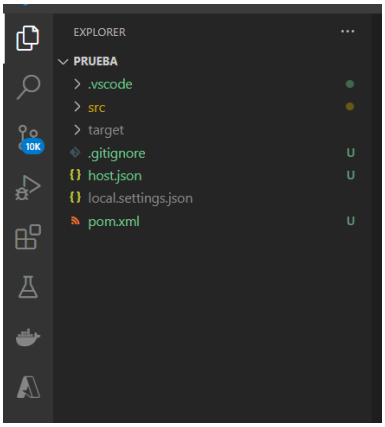
Seleccionamos entre las opciones si queremos que nuestro nuevo proyecto sea abierto en la ventana actual de Visual Studio Code, en una nueva ventana o si solo queremos guardar el proyecto en el espacio de trabajo para un uso posterior.



Sea cual sea la opción, será abierta la ventana de salidas de Visual Studio Code donde veremos los procesos para la creación del proyecto nuevo. Aquí es donde entrará en acción Apache Maven ya que se estarán corriendo comandos propios del software. Al finalizar el proceso, nos notificarán y se realizará la acción seleccionada en el paso anterior.



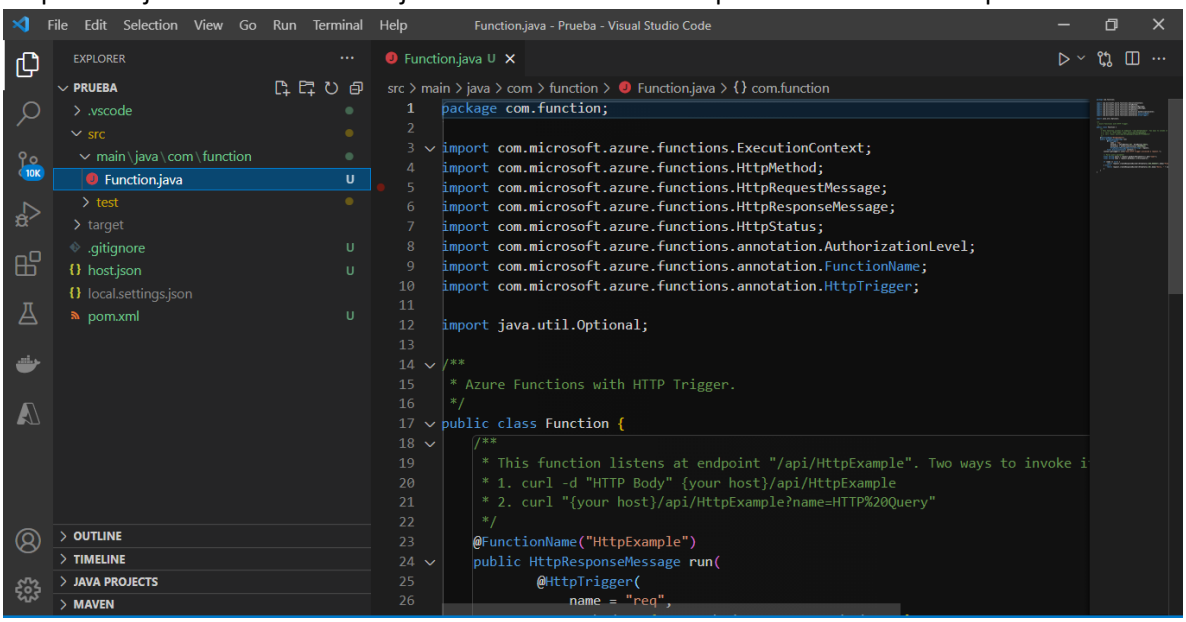
Podemos visualizar en el explorador de Visual Studio Code (Primera opción de la barra de actividades) el nuevo proyecto con la estructura propia de un proyecto Maven.



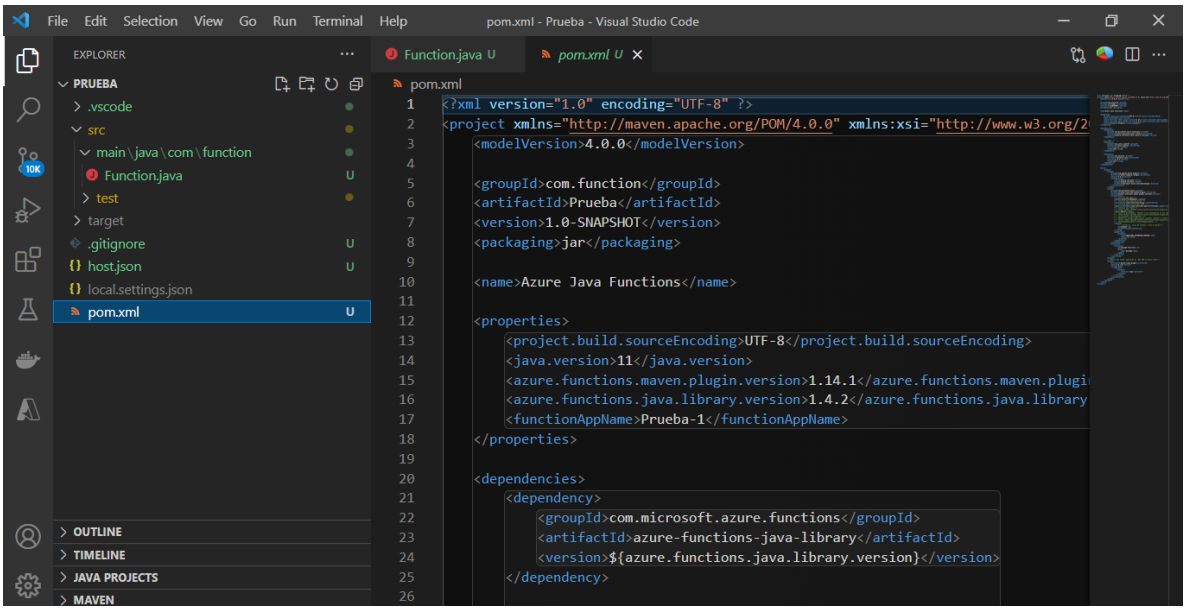
De forma automática, en la carpeta “src” se nos generará la clase function.java en donde tendremos el código de nuestra primera Azure Function.

Este ejemplo consta de un programa que escuchará los métodos de una aplicación HTTP. Se utilizará el desencadenador (trigger) HttpTrigger para que, en cuanto se genere una petición GET O POST, este detecte si se definió correctamente un valor en la variable “name” y este lo muestre

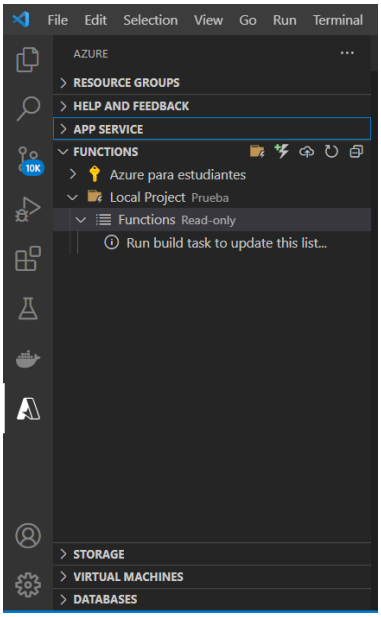
en pantalla junto con otro mensaje de bienvenida así como pintar en consola este proceso.



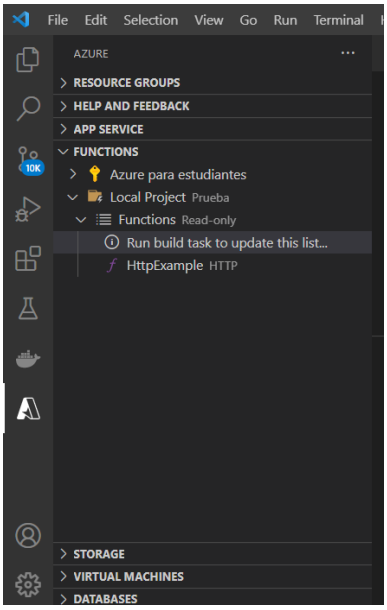
También se nos generará el archivo de configuración pom.xml, el cuál tendrá todaslas dependencias necesarias para correr el aplicativo.



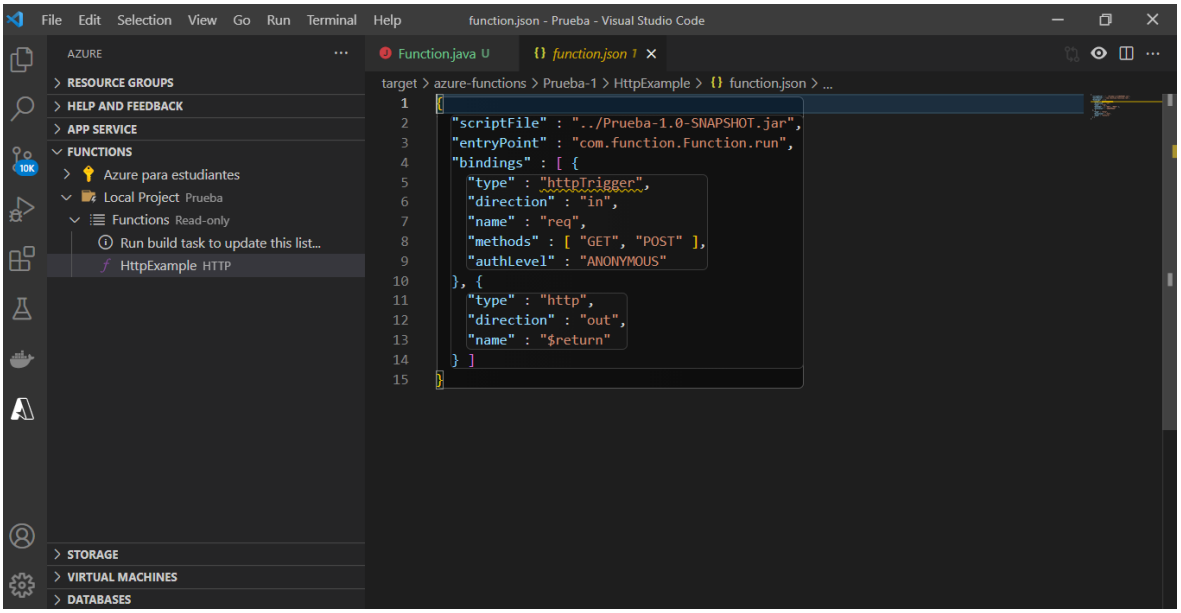
Ahora nos dirigiremos nuevamente a la sección de Azure y podremos observar que, en la sección de Functions, estará nuestro nuevo proyecto definido como un Local Proyect. Abriremos la carpeta y nos encontraremos con un mensaje que nos indicará que debemos correr el build del proyecto. Podemos hacerlo con la tecla F5 o dando click al mensaje



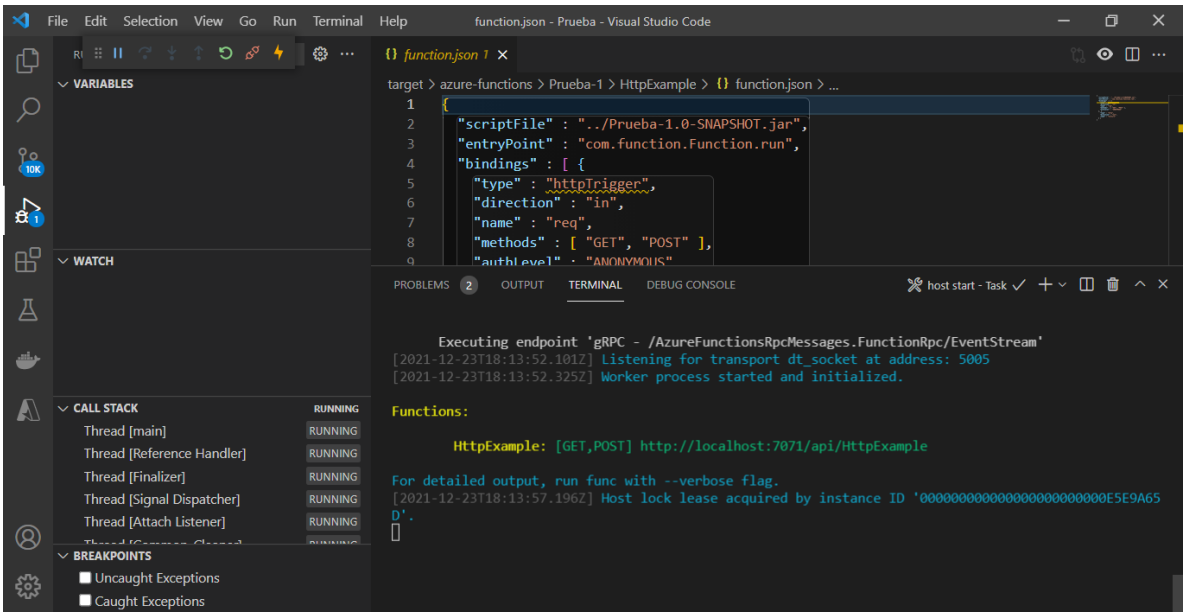
Al terminar el proceso Build, podremos observar que en nuestra lista aparecerá nuestra función



Al hacer click sobre la función, podemos visualizar el archivo function.json.

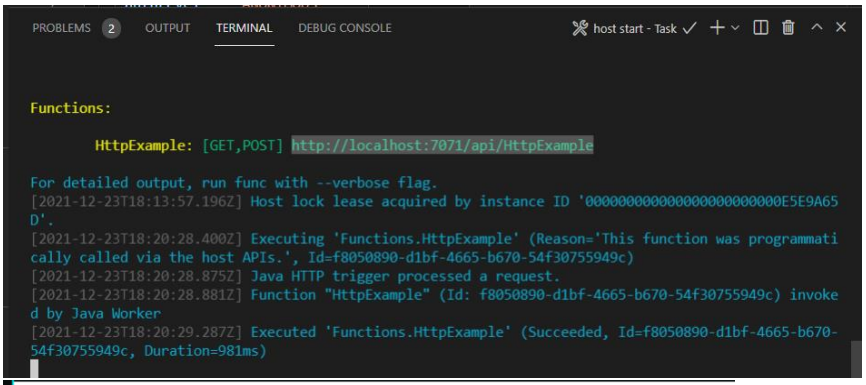


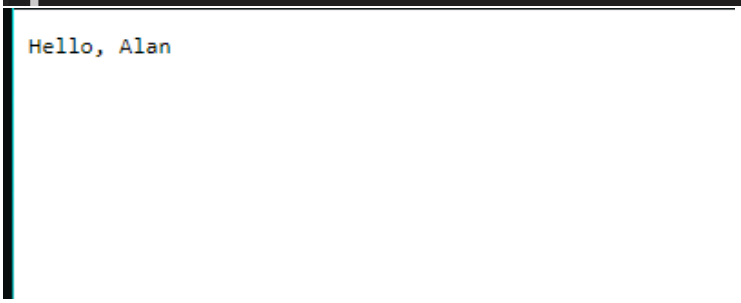
Por el momento nuestra aplicación correrá de forma local, para ello seleccionamos la carpeta Local Project de la sección Functions y presionamos la tecla F5. De nuevo se abrirá la terminal de Visual Studio Code pero obtendremos en consola la dirección HTTP con la cual estaremos trabajando



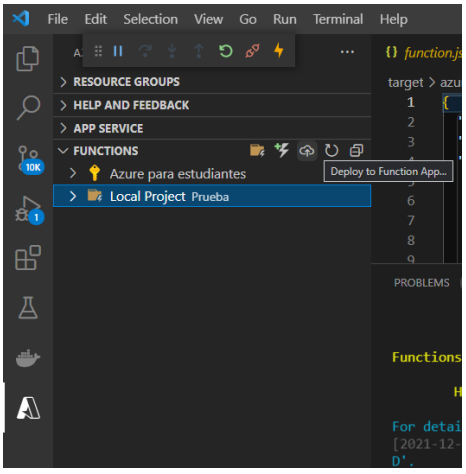
Ahora con la URL generada, podemos ser capaces de realizar pruebas de nuestra aplicación, puede ser mediante Postman o por medio del navegador. Para lanzar una petición, solo tenemos que agregar después de la URL lo siguiente: ?name=<YOUR_NAME> Ejemplo: http://localhost:7071/api/HttpExample?name=Alan

Los resultados obtenidos serán los siguientes:

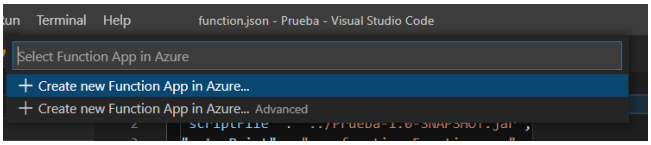




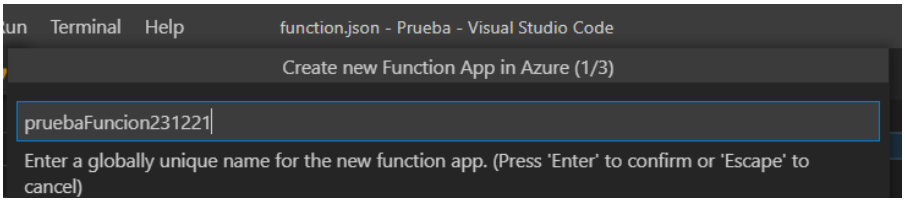
Para que nuestro ejemplo pueda tener conexión con nuestro entrono de Azure, debemos de subir nuestro proyecto. Para esto solo tenemos que posicionarnos sobre la sección Functions del menú de Azure y seleccionar el botón “Deploy to Function App”



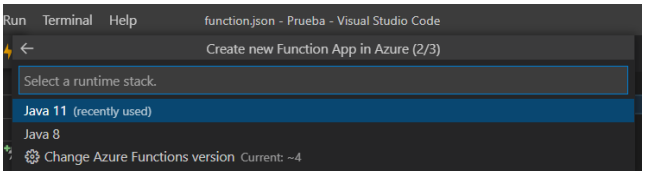
De nuevo tendremos una serie de cuestionamientos. Primero seleccionaremos la opción de generar una nueva Function App en Azure



Le daremos un nombre único a esta nueva Function App

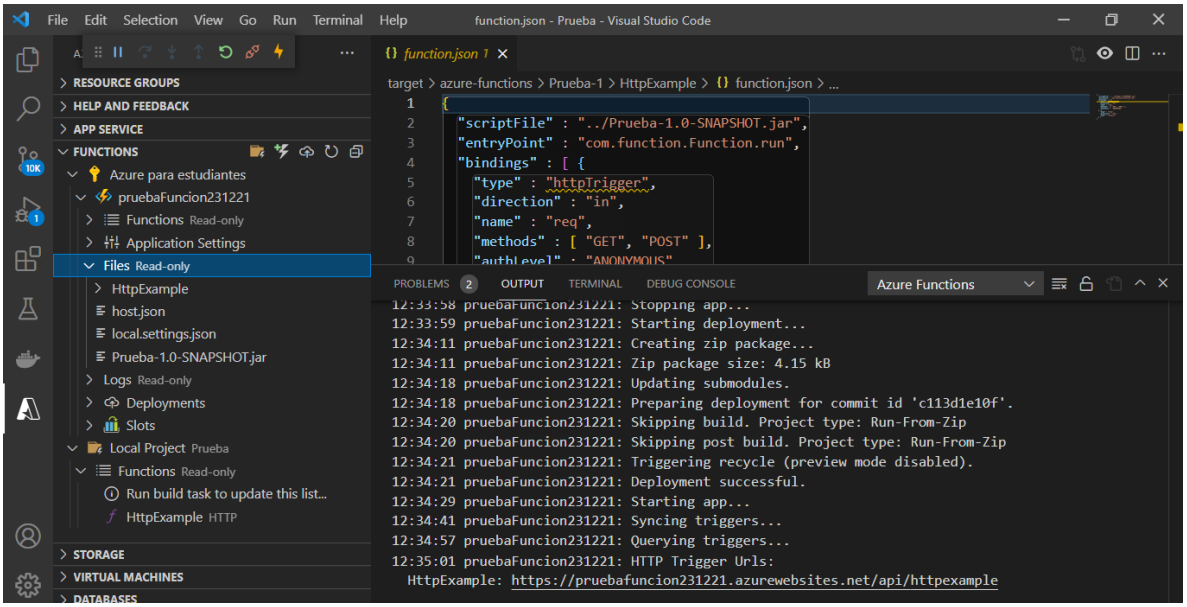


Seleccionamos la versión de Java sobre la cual estaremos ejecutando la aplicación.



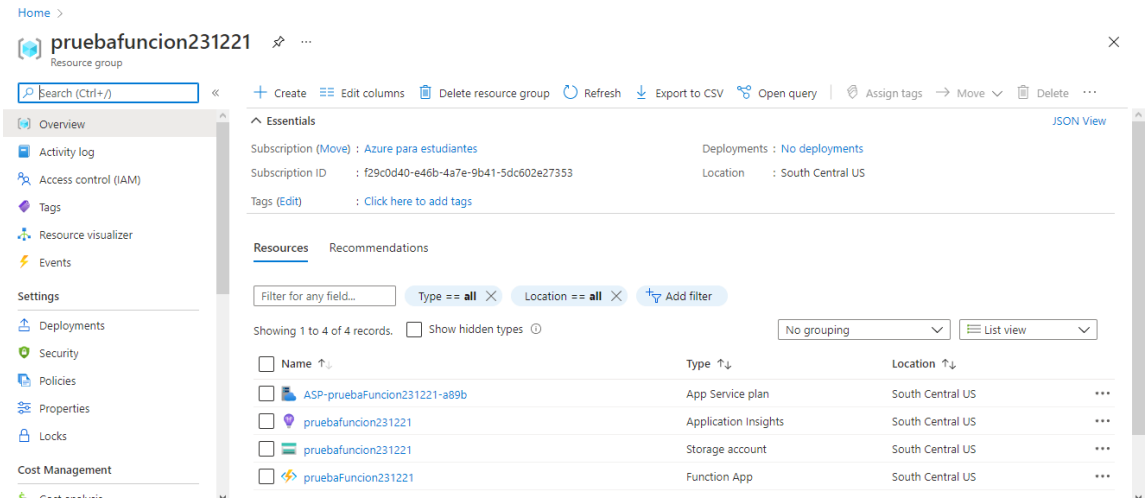
Seleccionamos la región en donde queremos que se hospede nuestro proyecto.

Empezará el proceso de creación de la Function App. Al finalizar nos mostrará una notificación en donde podremos ver la pantalla de salida de este proceso. También podremos ver en la sección Functions en la parte donde se encuentra nuestra suscripción que se generó una copia de nuestro proyecto local.



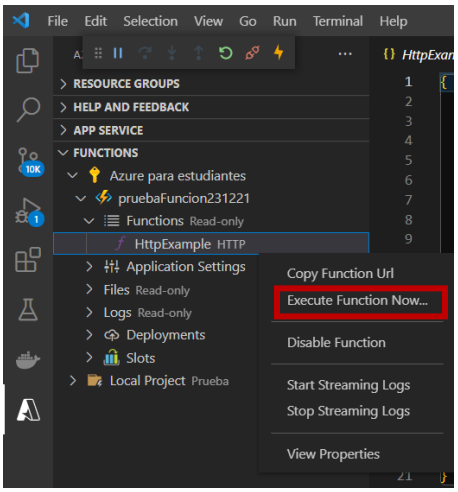
También se nos creará un grupo de recursos dentro de nuestra suscripción que contiene los recursos que requiere el ejemplo para funcionar, en este caso se crearon App Service Plan, una

cuenta de almacenamiento, la Function App y un Application Insights para supervisar en directo la aplicación. Todo esto lo podemos ver en el portal de Azure.

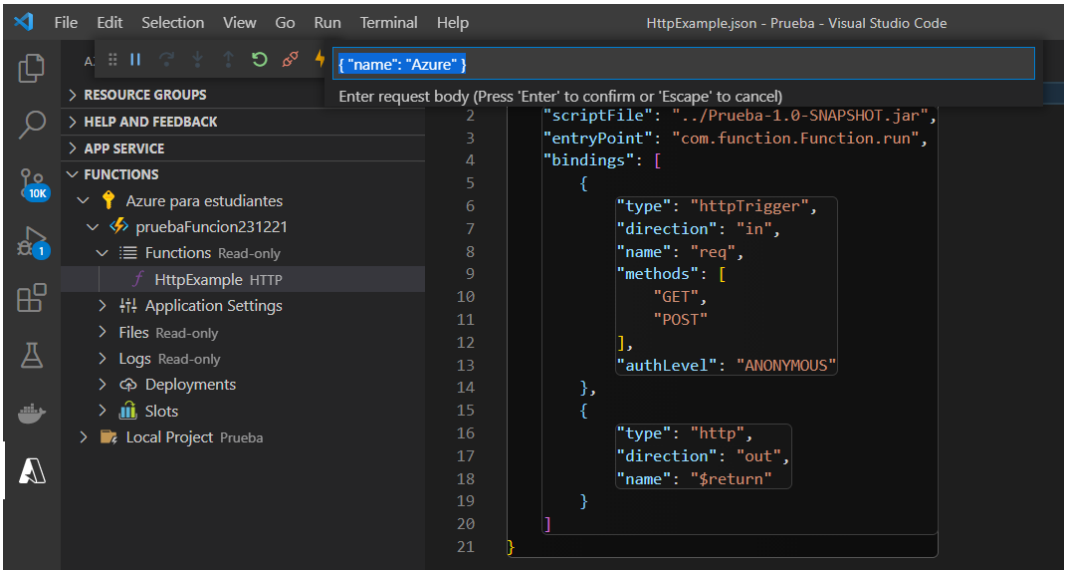


En la ventana de output, se nos generará una URL en donde podremos correr el mismo programa, pero ahora desde Azure. Podemos realizar las pruebas de los métodos desde el navegador, como si fuera un Local Project, o desde Visual Studio Code.

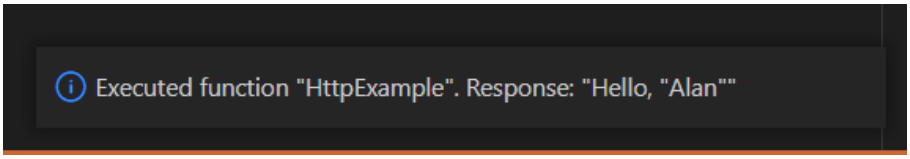
Realizaremos la invocación de la función alojada en Azure desde Visual Studio Code. Sobre la función HttpExample, damos click derecho y seleccionamos la opción “Execute Function Now”



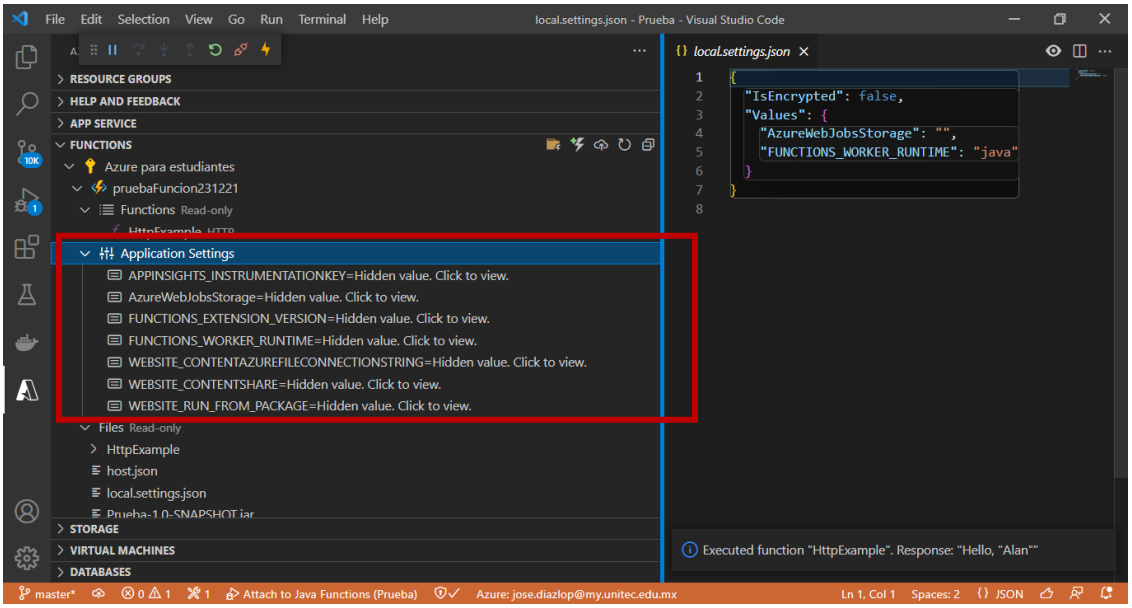
Se nos preguntará un “request body”, en este caso el request body es el valor que le daremos al String “name”.



Presionamos Enter y ahora en vez de tener un mensaje en consola, Visual Studio Code nos dará una notificación con el mensaje que anteriormente era visualizado en log.



Algo importante de aclarar es que, tradicionalmente, los métodos para conectar la aplicación con nuestro entorno de Azure se declaraban en el archivo de configuración local.settings.json pero ahora, al construir la Function App, se generarán las variables de conexión de forma automática en una sección llamada “Application Settings”, estas por defecto están ocultas pero pueden ser visibles y, si se requiere, podemos copiarlas y pegarlas en el archivo de configuración.



Repositorio GitHub

El proyecto generado en este ejemplo puede ser descargado en el siguiente repositorio de GitHub:

https://github.com/PADSA-github/Cloud/tree/main/Azure/Azure-Function-Java