

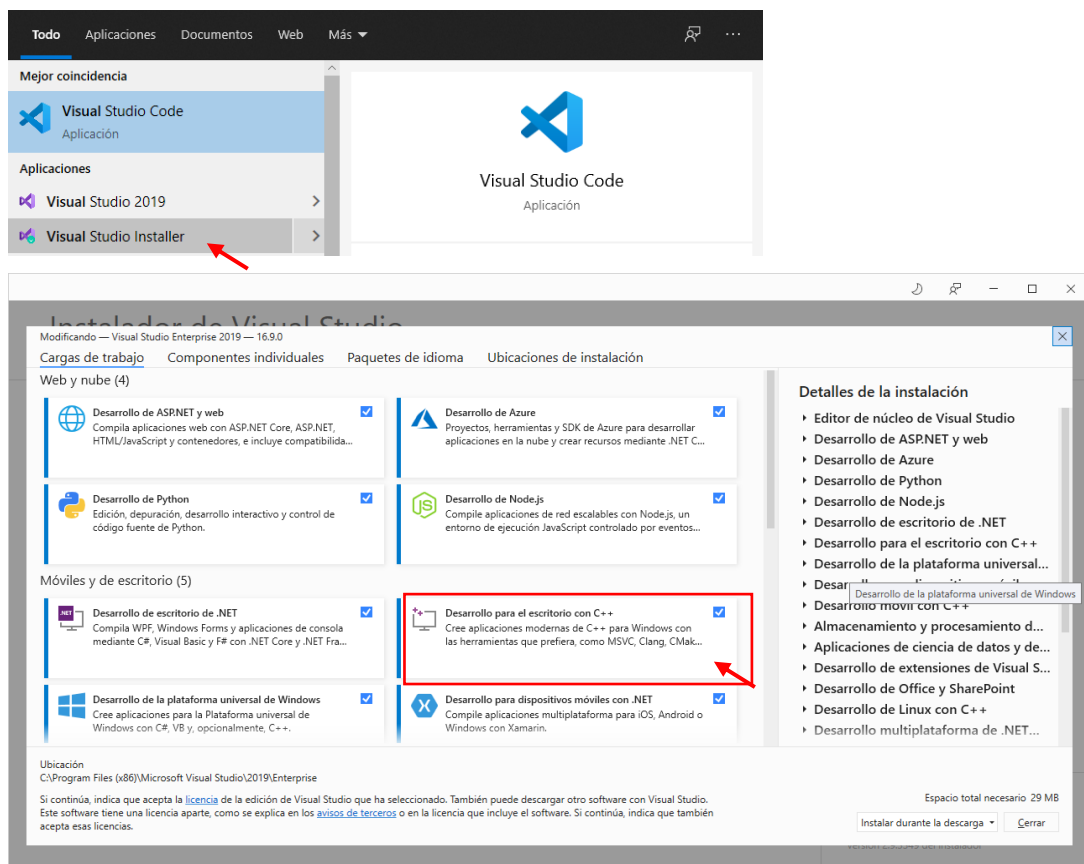
# Configuración de Aprovisionamiento de un dispositivo IoT simulado con claves simétricas

Esta guía permitirá comprender el proceso de configuración del dispositivo iot simulado durante el proceso de fabricación, para permitir el aprovisionarlo automáticamente con IoT Hub. El dispositivo se aprovisiona en función de su mecanismo de atestación, tras el primer arranque y su conexión al servicio de aprovisionamiento.

## Configurar el entorno de desarrollo Windows

La directiva de asignación es una configuración del servicio IoT Hub Device Provisioning que determina cómo se asignan los dispositivos a un centro de IoT.

1. Instalar Visual Studio 2019 con la carga de trabajo "Desarrollo para el escritorio con C++" habilitada.
2. Si ya tiene instalado Visual Studio, deberá buscar en el Menu de Windows la opción Visual Studio Installer y agregar la carga de trabajo indicada.

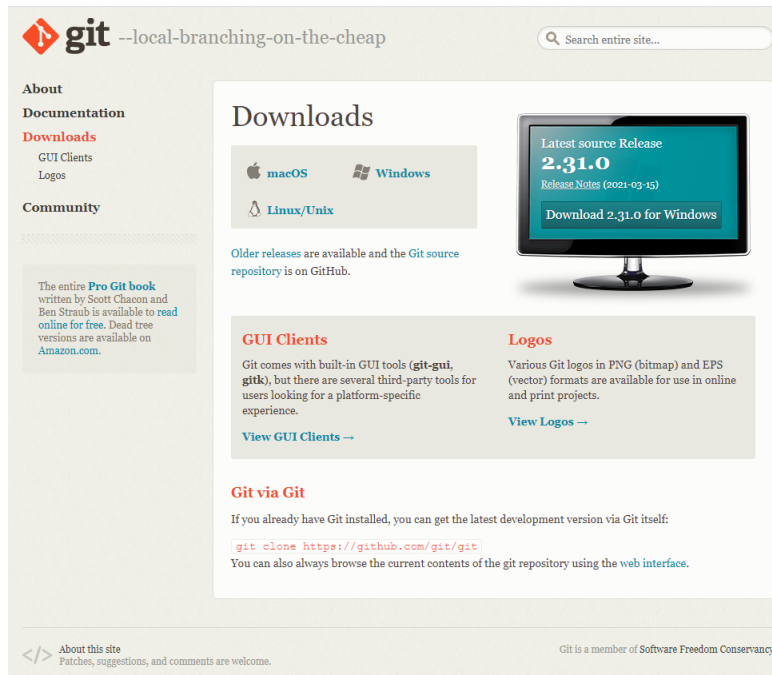


- Puede instalar Visual Studio 2015 o Visual Studio 2017 pero VS 2019 es la version recomendada para este ejemplo.

- En caso su equipo físico no soporte la instalación de VS 2019 puede crear una Máquina Virtual con el sistema operativo Windows 10 (Actualización 20H2)
3. Instalar la versión más reciente de Git.

<https://git-scm.com/download/>

Al instalar conserve las opciones por defecto.



## Compilación de una versión específica de la plataforma del SDK

El SDK de cliente del servicio Device Provisioning permite implementar el software de registro del dispositivo. Pero para poder usarlo, es preciso compilar una versión del SDK específico de su plataforma de cliente de desarrollo y mecanismo de plataforma.

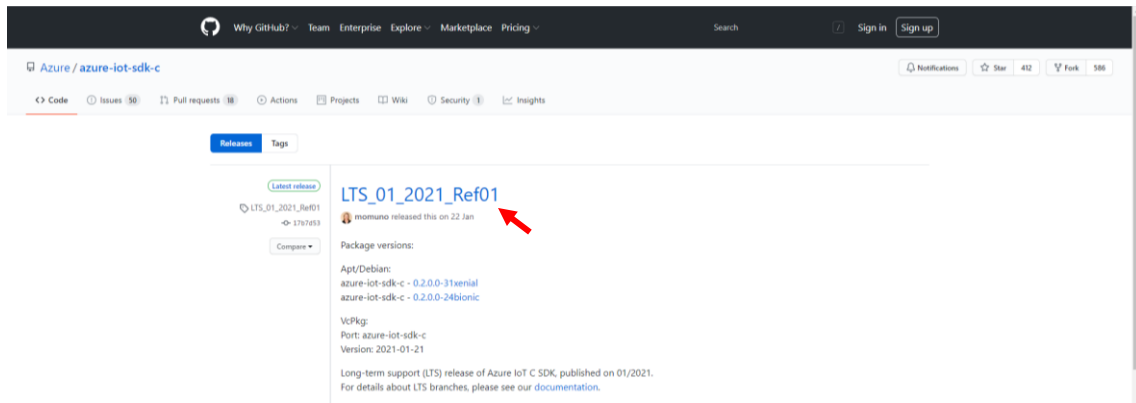
Esta guía compila un SDK que usa Visual Studio en una plataforma de desarrollo de Windows para un tipo de ejecución compatible.

1. Descargue el sistema de compilación CMake.

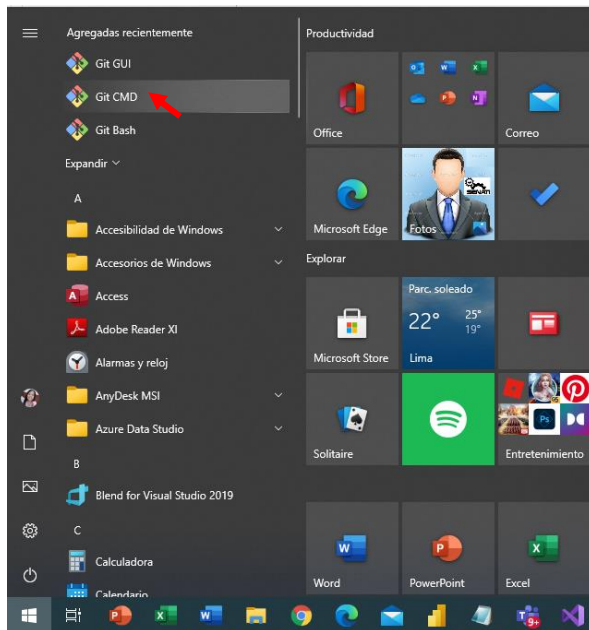
**NOTA:** Antes de comenzar la instalación de CMake, es importante que haya instalado Visual Studio (Visual Studio y la carga de trabajo de desarrollo de escritorio con C++).

2. Busque el nombre de la versión más reciente del SDK.

Link: [https://github.com/Azure/azure-iot-sdk-c/releases/tag/LTS\\_01\\_2021\\_Ref01](https://github.com/Azure/azure-iot-sdk-c/releases/tag/LTS_01_2021_Ref01)

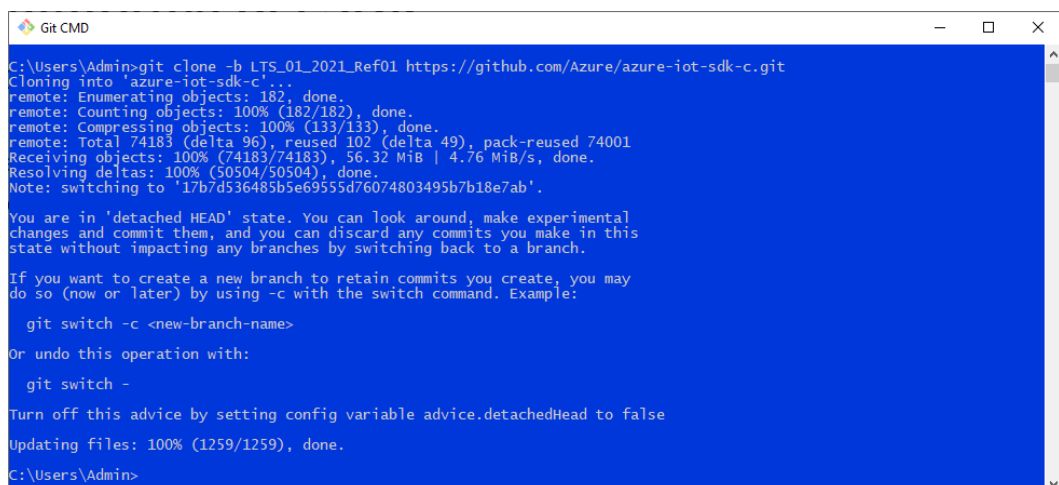


3. Abra un símbolo del sistema o el shell de Bash de Git.

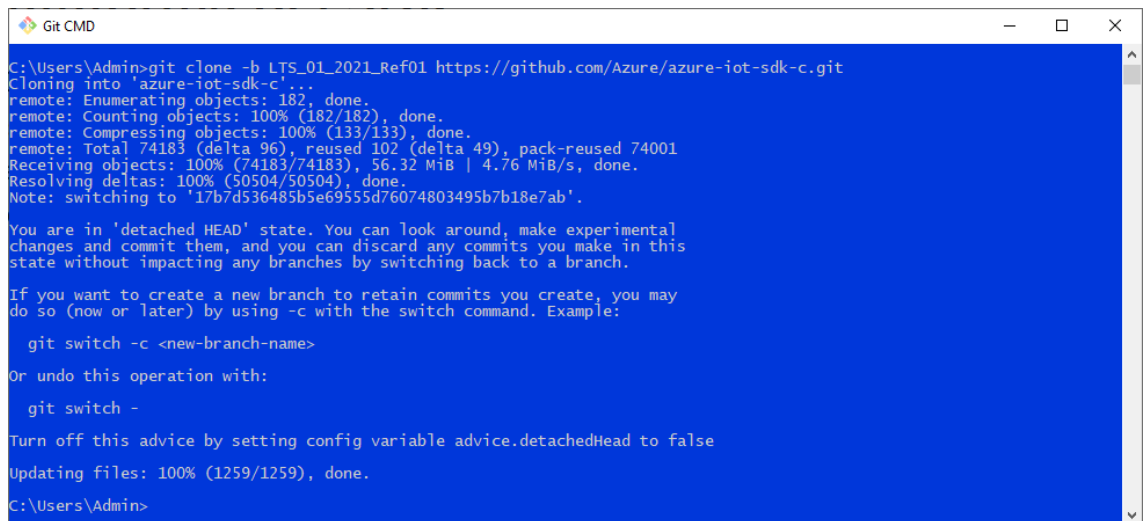


4. Ejecute los siguientes comandos para clonar la versión más reciente del repositorio de GitHub del SDK de Azure IoT para C. Verifique que la etiqueta mas reciente sea la que está marcada.

```
git clone -b LTS_01_2021_Ref01 https://github.com/Azure/azure-iot-sdk-c.git
```



```
cd azure-iot-sdk-c
git submodule update -init
```



```
Git CMD
C:\Users\Admin>git clone -b LTS_01_2021_Ref01 https://github.com/Azure/azure-iot-sdk-c.git
Cloning into 'azure-iot-sdk-c'...
remote: Enumerating objects: 182, done.
remote: Counting objects: 100% (182/182), done.
remote: Compressing objects: 100% (133/133), done.
remote: Total 74183 (delta 96), reused 102 (delta 49), pack-reused 74001
Receiving objects: 100% (74183/74183), 56.32 MiB | 4.76 MiB/s, done.
Resolving deltas: 100% (50504/50504), done.
Note: switching to '17b7d536485b5e69555d76074803495b7b18e7ab'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>

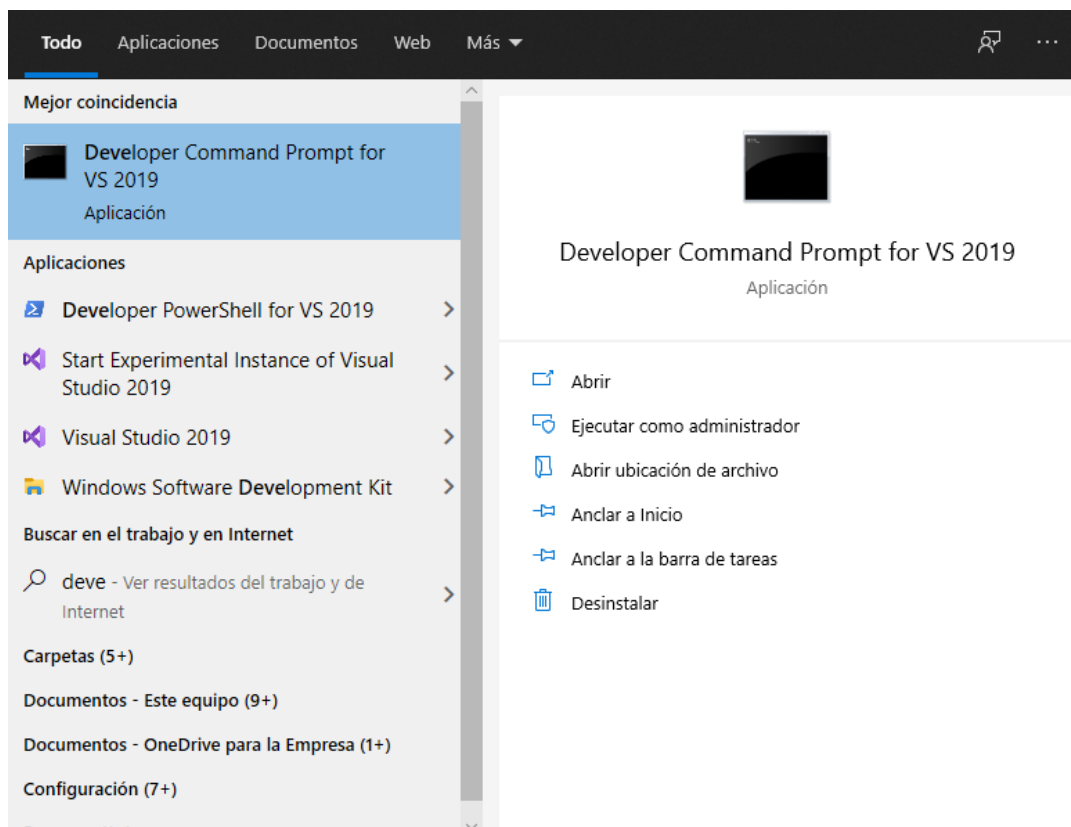
Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false
Updating files: 100% (1259/1259), done.
C:\Users\Admin>
```

Esta operación puede tardar varios minutos en completarse.

##### 5. Ingrese a Developer Command Prompt for VS 2019

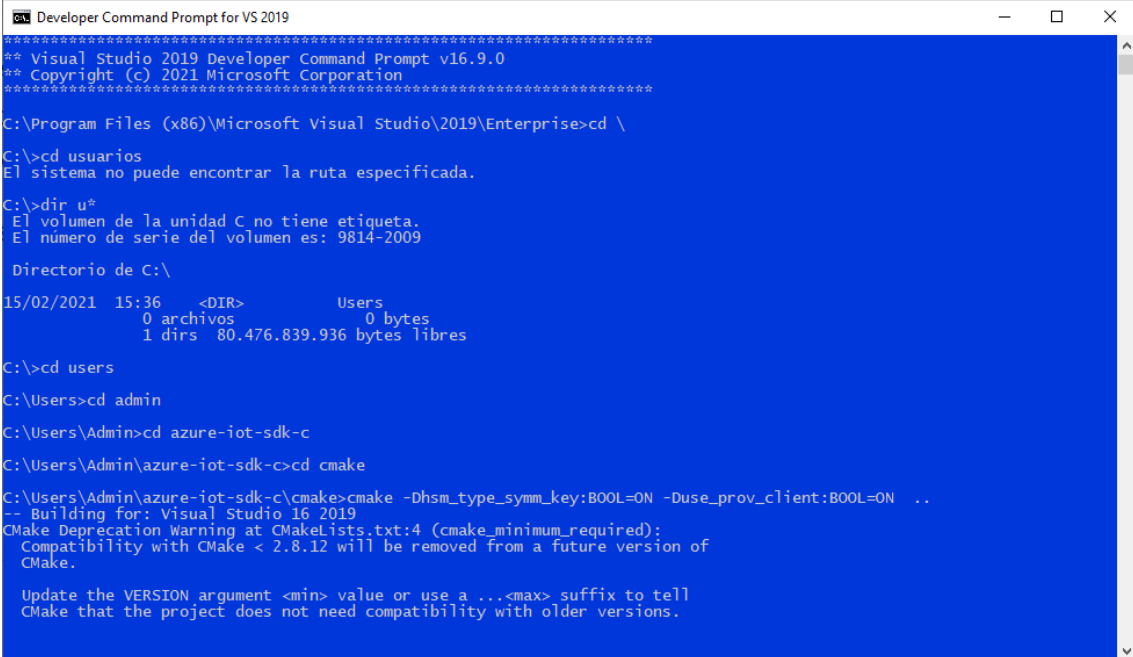


6. Ingrese al repositorio de Git (Carpeta del usuario del equipo con el que instalo el Git)
7. Ingrese a la caperta **azure-iot-sdk-c**
8. Cree un subdirectorio cmake con los siguientes comandos:

```
mkdir cmake
cd cmake
```

9. Ejecute el siguiente comando para compilar una versión del SDK específica para su plataforma de cliente de desarrollo. Se generará una solución de Visual Studio para el código de aprovisionamiento de dispositivos en el directorio cmake.

```
cmake -Dhsm_type_symm_key:BOOL=ON -Duse_prov_client:BOOL=ON ..
```



```
Developer Command Prompt for VS 2019
** Visual Studio 2019 Developer Command Prompt v16.9.0
** Copyright (c) 2021 Microsoft Corporation
C:\Program Files (x86)\Microsoft Visual Studio\2019\Enterprise>cd \
C:\>cd usuarios
El sistema no puede encontrar la ruta especificada.
C:\>dir u*
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: 9814-2009

Directorio de C:\
15/02/2021  15:36    <DIR>          Users
               0 archivos             0 bytes
               1 dirs  80.476.839.936 bytes libres
C:\>cd users
C:\Users>cd admin
C:\Users\Admin>cd azure-iot-sdk-c
C:\Users\Admin\azure-iot-sdk-c>cd cmake
C:\Users\Admin\azure-iot-sdk-c\cmake>cmake -Dhsm_type_symm_key:BOOL=ON -Duse_prov_client:BOOL=ON ..
-- Building for: Visual Studio 16 2019
CMake Deprecation Warning at CMakeLists.txt:4 (cmake_minimum_required):
  Compatibility with CMake < 2.8.12 will be removed from a future version of
  CMake.

  Update the VERSION argument <min> value or use a ...<max> suffix to tell
  CMake that the project does not need compatibility with older versions.
```