

# Preparación de Entorno para Azure Functions (.NET y PowerShell)

Este documento describe paso a paso todo lo necesario para instalar, configurar y preparar tu entorno local para desarrollar, probar y desplegar Azure Functions usando .NET y PowerShell.

## 1. Requisitos Previos

- Sistema Operativo: Windows 10/11 (64-bit) o equivalente con WSL2 para Linux.
- Cuenta de Azure activa (puedes usar la capa gratuita en <https://azure.microsoft.com/free/>).
- Conexión a internet estable.

## 2. Instalación de .NET SDK

- Descargar desde <https://dotnet.microsoft.com/download>.
- Instalar la versión LTS más reciente compatible con Azure Functions (actualmente .NET 6 o superior).
- Verificar instalación: ejecutar ``dotnet --version`` en la terminal.

## 3. Instalación de PowerShell Core

- Descargar desde <https://aka.ms/powershell>.
- Instalar la versión estable más reciente.
- Verificar instalación: ejecutar ``pwsh --version``.

## 4. Azure Functions Core Tools

- Instalar mediante npm (requiere Node.js instalado previamente): ``npm install -g azure-functions-core-tools@4 --unsafe-perm true``.
- Verificar instalación: ejecutar ``func --version``.
- Permite crear, ejecutar y depurar funciones localmente.

## 5. Azure CLI

- Descargar e instalar desde <https://learn.microsoft.com/cli/azure/install-azure-cli>.
- Verificar instalación: ``az --version``.
- Iniciar sesión: ``az login``.

## 6. Visual Studio Code

- Descargar e instalar desde <https://code.visualstudio.com/>.
- Extensiones recomendadas:

- - Azure Functions
- - Azure Account
- - C#
- - PowerShell
- - Azure Storage

## 7. Emulador de Azure Storage (Azurite)

- Instalar como extensión en Visual Studio Code o con npm: ``npm install -g azurite``.
- Iniciar con ``azurite`` en la terminal.
- Útil para pruebas locales de triggers/bindings de Blob, Queue y Table.

## 8. Configuración Inicial

- Iniciar sesión en Azure CLI: ``az login``.
- Configurar suscripción activa: ``az account set --subscription ``.
- En Visual Studio Code, vincular la cuenta de Azure desde la extensión Azure Account.

## 9. Prueba de Verificación

- Crear una Function App local: ``func init MyFuncApp --worker-runtime dotnet`` o ``func init MyFuncApp --worker-runtime powershell``.
- Agregar una función HTTP Trigger: ``func new --template "HTTP trigger" --name TestFunction``.
- Ejecutar localmente: ``func start``.
- Probar en navegador o con curl en ``http://localhost:7071/api/TestFunction``.

## 10. Despliegue Rápido

- Crear Function App en Azure: ``az functionapp create --resource-group MyResourceGroup --consumption-plan-location eastus --runtime dotnet --functions-version 4 --name --storage-account ``.
- Desplegar desde VS Code usando la extensión Azure Functions (clic derecho en el proyecto > Deploy to Function App).