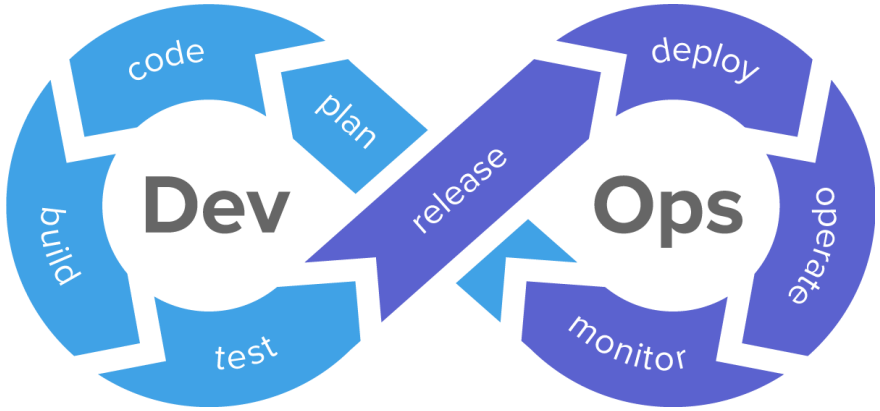


Microsoft Azure Devops (MS-TFS)

Por: Carlos Carreño

ccarreno@cienciadedatos.es

Octubre, 2020

Unidad 8 Recursos de Desarrollo

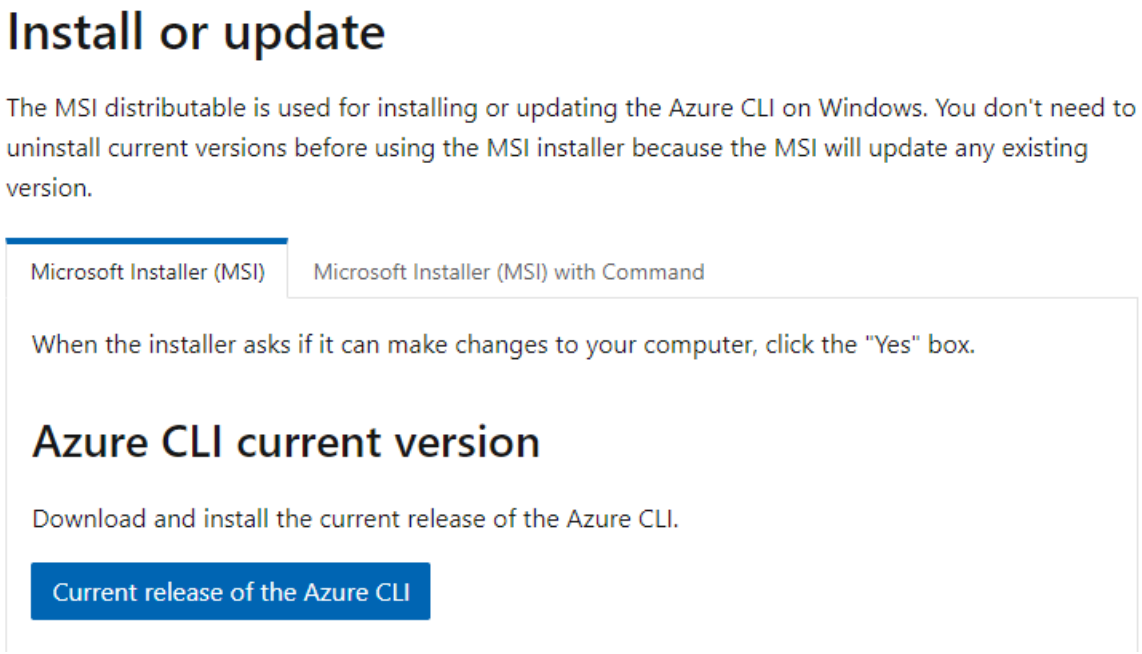
- Command line interface
- REST APIs
- DevOps resource center

Command line interface

- Azure DevOps cuenta con una **extensión** para la interfaz de línea de comandos (CLI) de Azure, puede administrar muchos servicios de Azure DevOps desde la línea de comandos.
- Los comandos de la CLI le permiten **optimizar sus tareas** con un marco de trabajo **interactivo más rápido y flexible**, sin pasar por los flujos de trabajo de la interfaz de usuario.

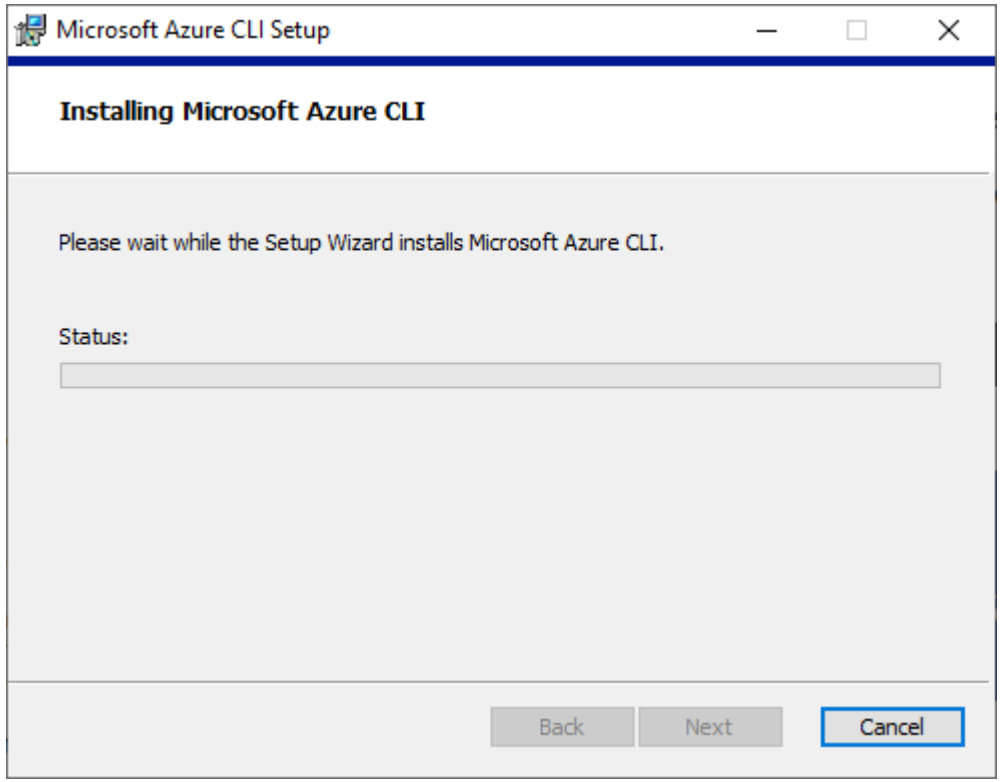
CLI: Instalación del cliente

- Descarga el archivo *.msi desde https://aka.ms/installazurecliwindows



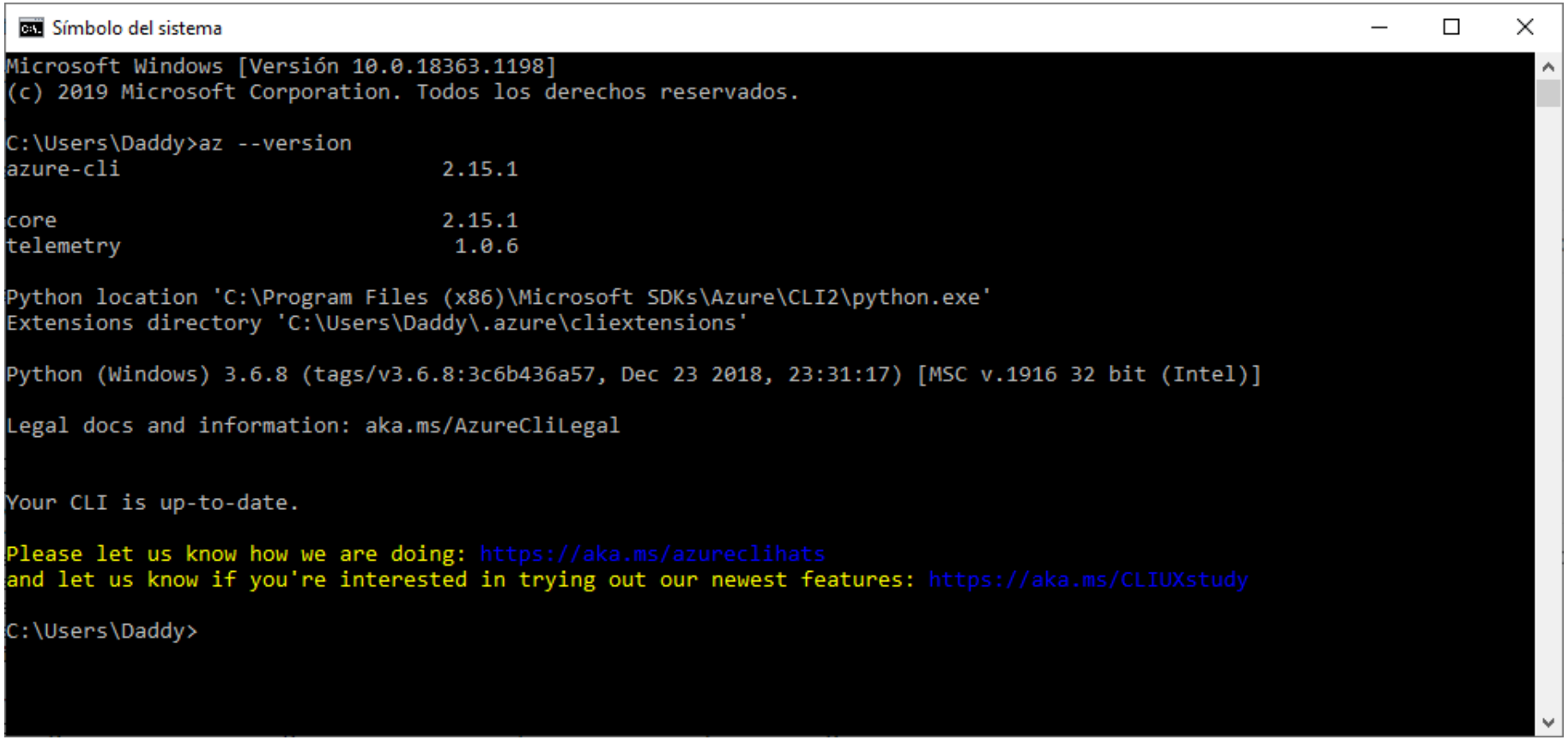
...

- Sigue los pasos del asistente de instalación



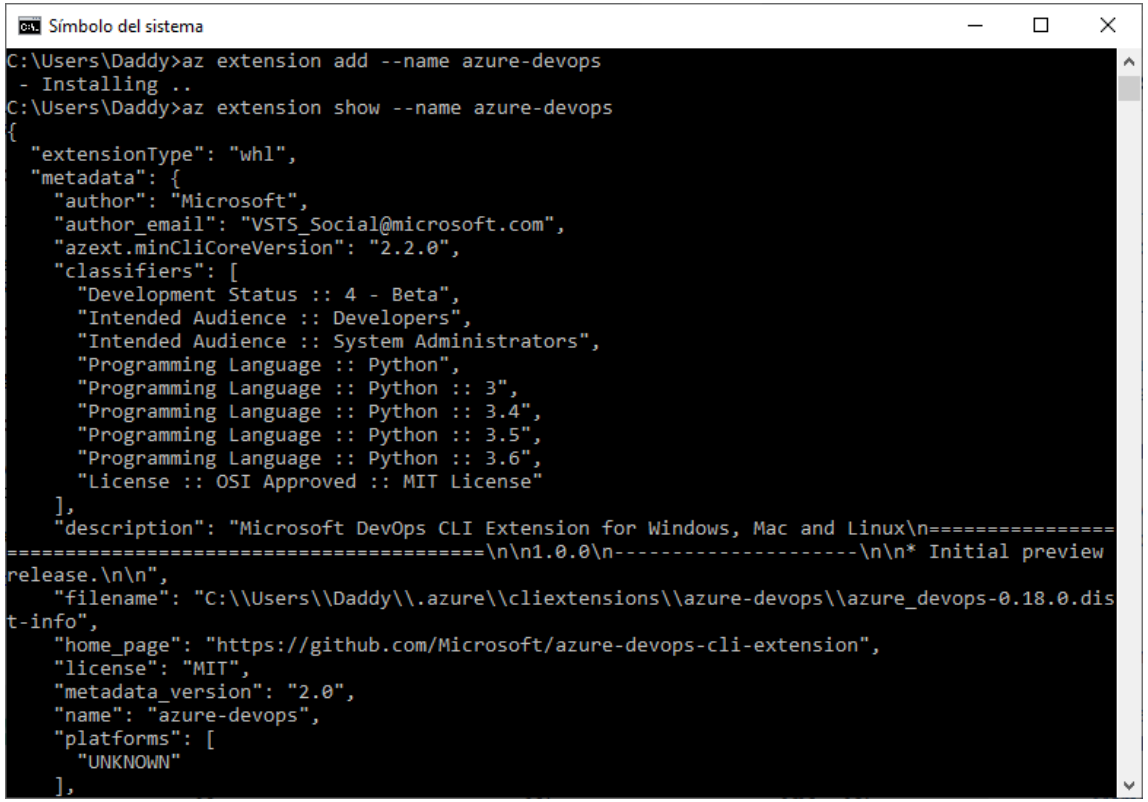
CLI: Verificando la instalación

- Ejecuta: az --version



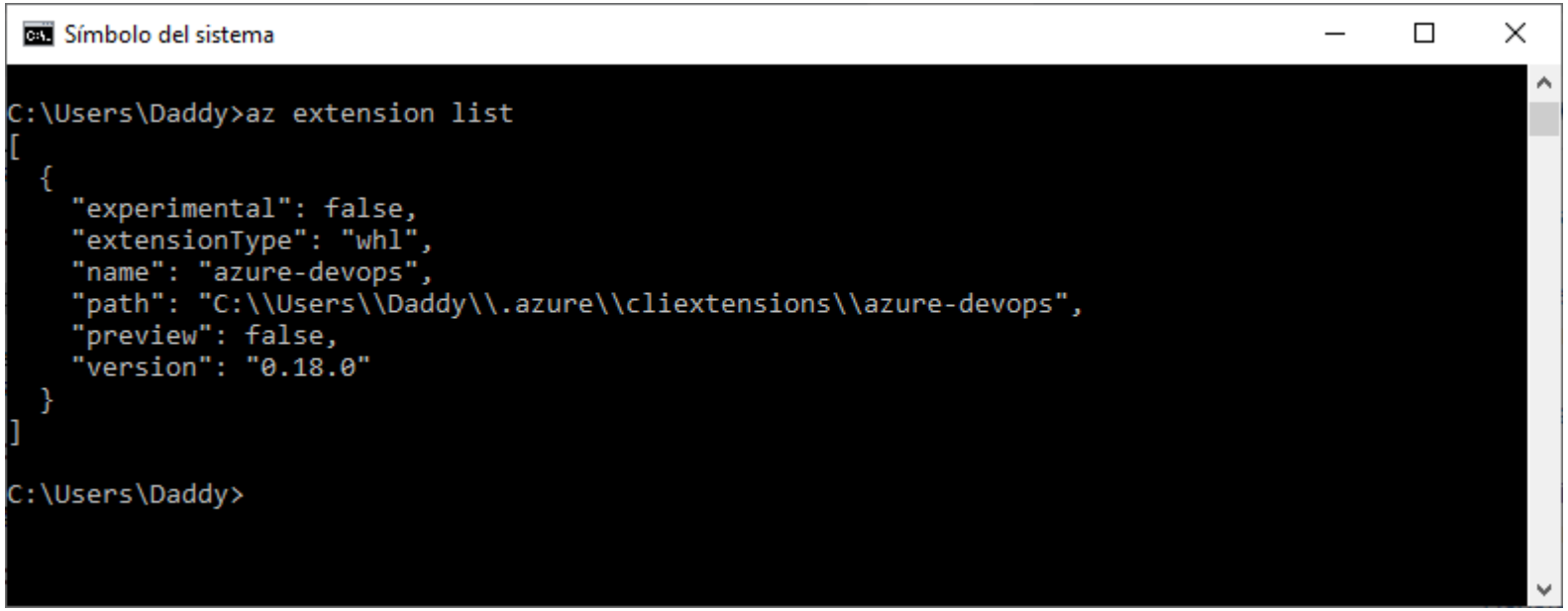
CLI: Agregando la extensión Azure DevOps

- Ejecuta: az extension add --name azure-devops



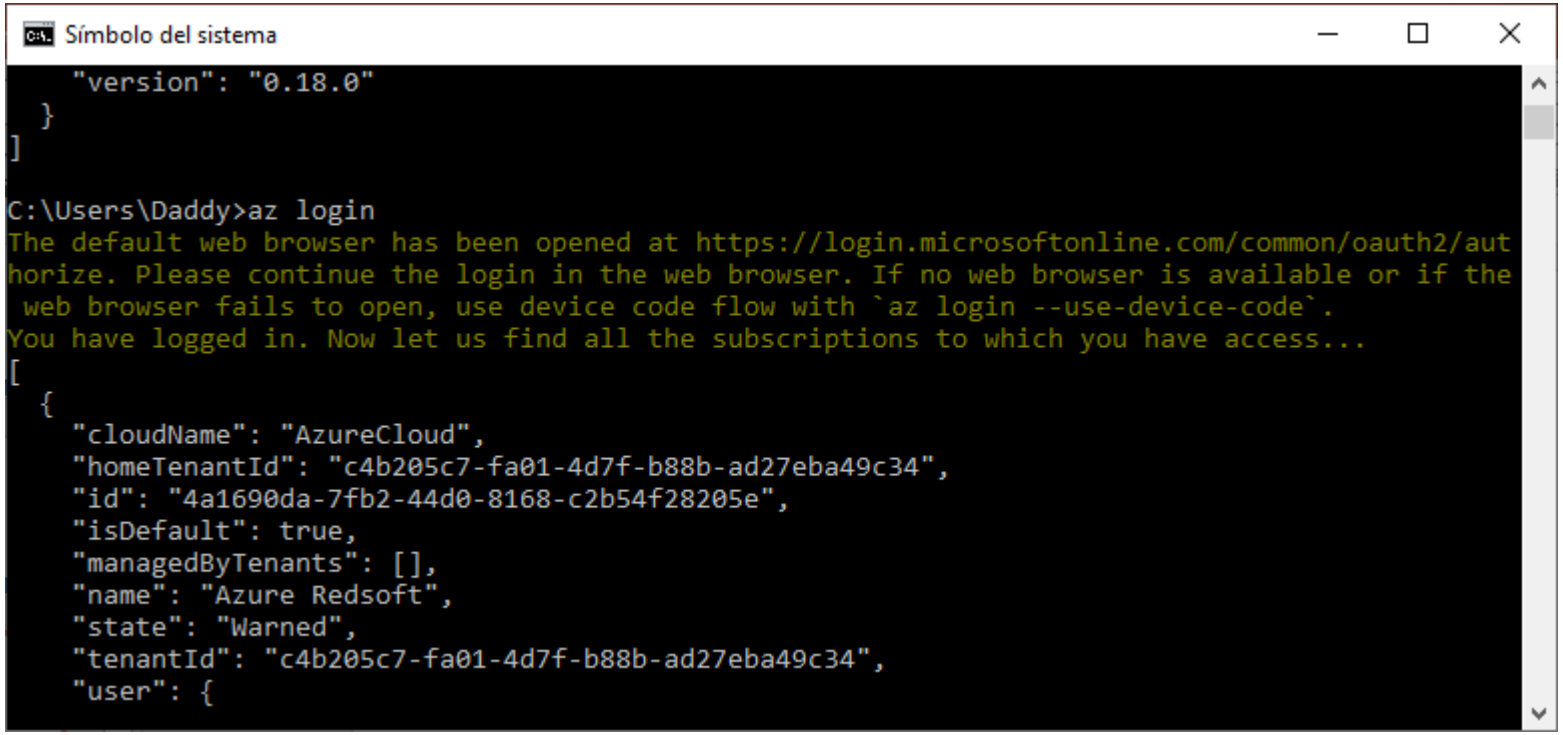
CLI: Lista de extensiones

- Mostrar la lista de extensiones intaladas

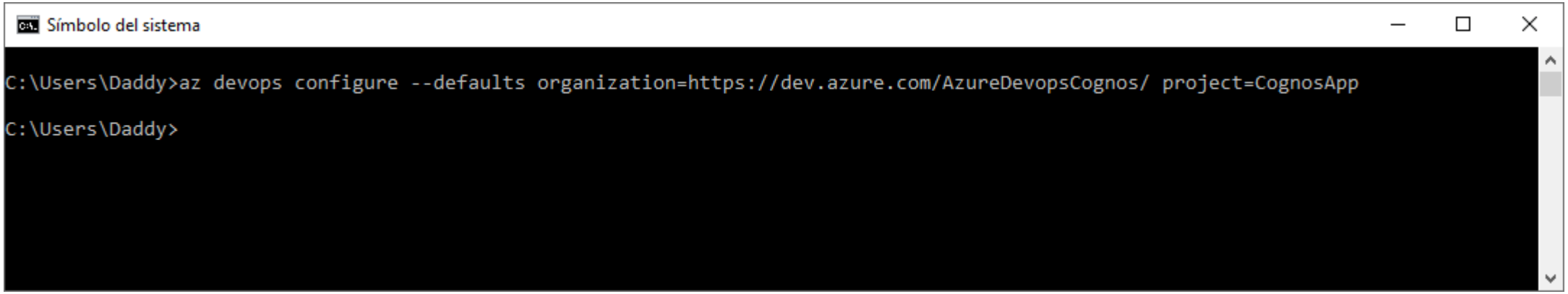


Login en Azure

- Ingresa a Azure desde CLI

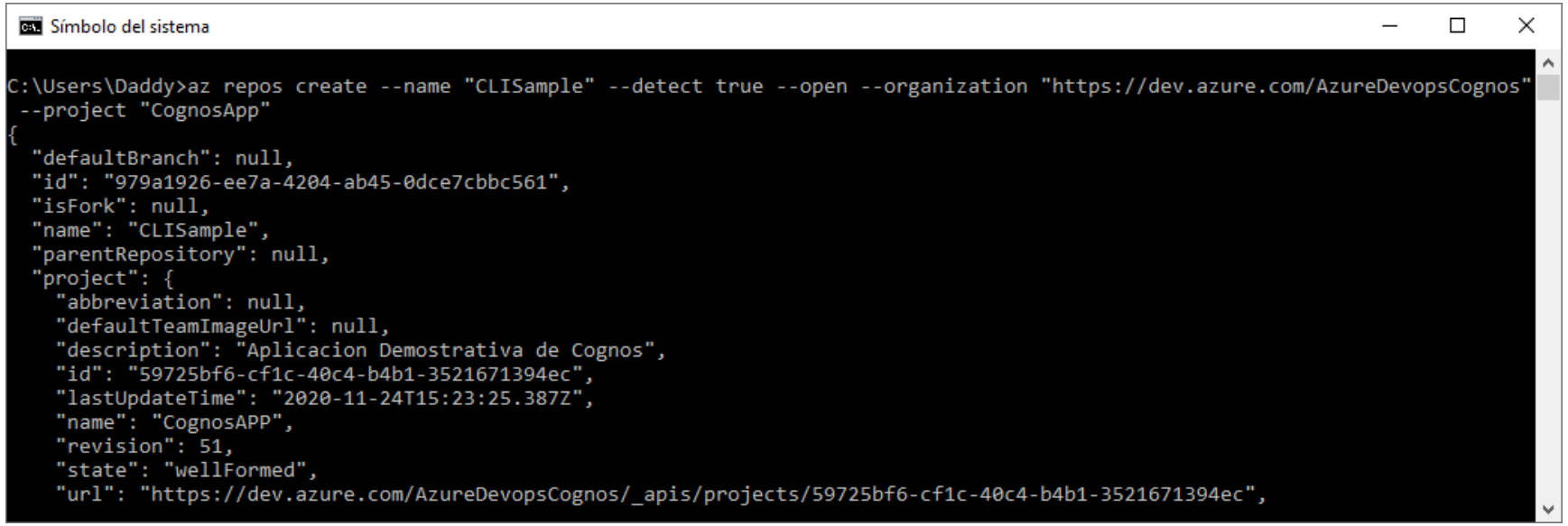


CLI: Definiendo la Organización y proyecto por defecto



CLI: Creando un repo

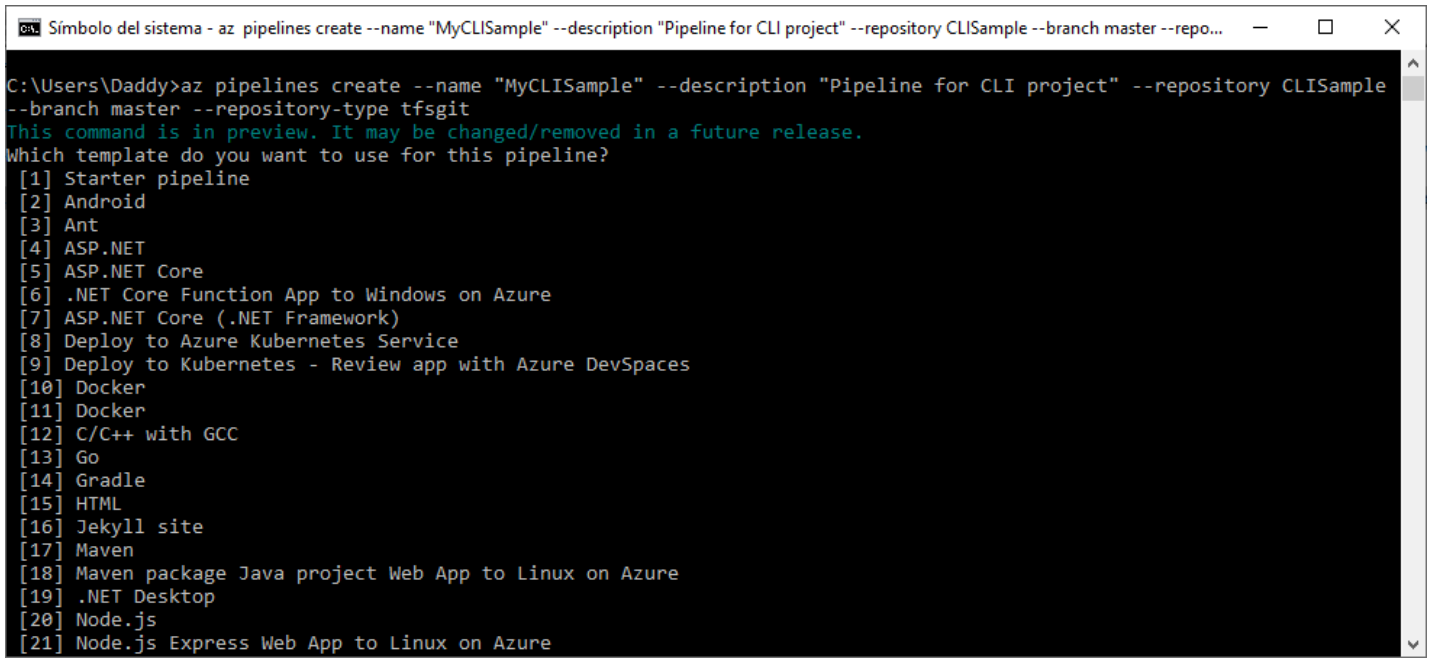
- az repos create --name "CLISample" --detect true --open --organization "https://dev.azure.com/AzureDevopsCognos" --project "CognosApp"



CLI: Creando un pipeline

- Crea un pipeline en el repositorio:

az pipelines create --name "MyCLISample" --description "Pipeline for CLI project" --repository CLISample --branch master --repository-type **tfsgit**



REST APIs

- Las API de **Representational State Transfer** (REST) admiten un conjunto de operaciones (métodos) HTTP, que proporcionan acceso para crear, recuperar, actualizar o eliminar a los recursos del servicio. Este artículo lo guía a través de:
  - ❑ Los componentes básicos de un par de solicitud / respuesta de API REST.
  - ❑ Descripción general de la creación y el envío de una solicitud REST y el manejo de la respuesta.

API REST Components

- Un par de solicitud / respuesta de API REST se puede dividir en cinco componentes:

1. The **request URI**, in the following form: `VERB https://{instance}[/{team-project}]/_apis[/{area}]/{resource}?api-version={version}`
    - *instance*: The Azure DevOps Services organization or TFS server you're sending the request to. They are structured as follows:
        - Azure DevOps Services: `dev.azure.com/{organization}`
        - TFS: `{server:port}/tfs/{collection}` (the default port is 8080, and the value for collection should be `DefaultCollection` but can be any collection)
    - *resource path*: The resource path is as follows: `_apis/{area}/{resource}`. For example `_apis/wit/workitems`.
    - *api-version*: Every API request should include an api-version to avoid having your app or service break as APIs evolve. api-versions are in the following format: `{major}.{minor}[-{stage}[.{resource-version}]]`, for example:
        - `api-version=1.0`
        - `api-version=1.2-preview`
        - `api-version=2.0-preview.1`

2. HTTP **request message header** fields:

    - A required HTTP method (also known as an operation or verb), which tells the service what type of operation you are requesting. Azure REST APIs support GET, HEAD, PUT, POST, and PATCH methods.
    - Optional additional header fields, as required by the specified URI and HTTP method. For example, an Authorization header that provides a bearer token containing client authorization information for the request.

3. Optional HTTP **request message body** fields, to support the URI and HTTP operation. For example, POST operations contain MIME-encoded objects that are passed as complex parameters.

    - For POST or PUT operations, the MIME-encoding type for the body should be specified in the Content-type request header as well. Some services require you to use a specific MIME type, such as `application/json`.

4. HTTP **response message header** fields:

- An HTTP status code, ranging from 2xx success codes to 4xx or 5xx error codes. Alternatively, a service-defined status code may be returned, as indicated in the API documentation.
- Optional additional header fields, as required to support the request's response, such as a `Content-type` response header.

5. Optional HTTP **response message body** fields:

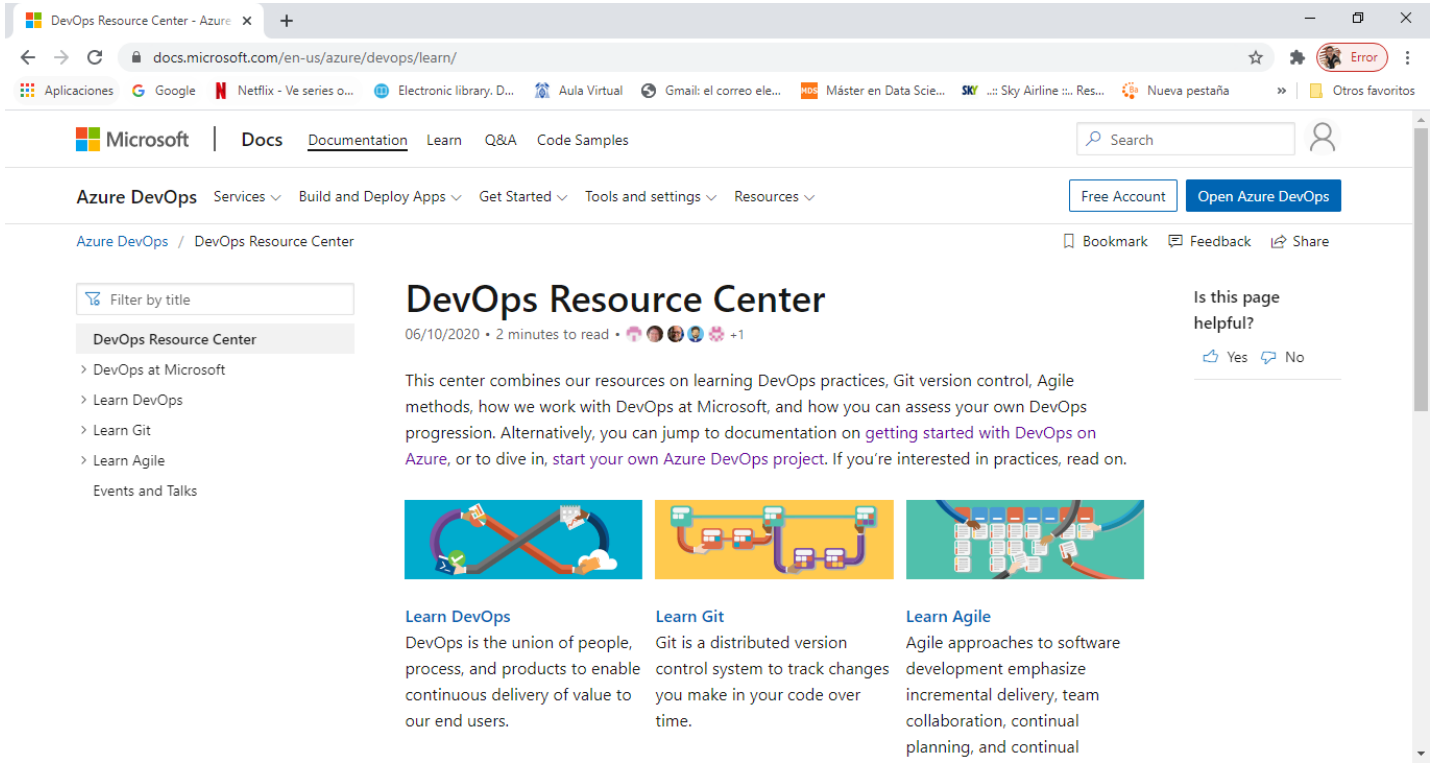
- MIME-encoded response objects may be returned in the HTTP response body, such as a response from a GET method that is returning data. Typically, these objects are returned in a structured format such as JSON or XML, as indicated by the `Content-type` response header. For example, when you request an access token from Azure AD, it will be returned in the response body as the `access_token` element, one of several name/value paired objects in a data collection. In this example, a response header of `Content-Type: application/json` is also included.

Ejemplo de uso de API REST en C#

```
public static async void GetProjects() {
        try {
            var personalaccesstoken = "PAT_FROM_WEBSITE";
            using (HttpClient client = new HttpClient()){
                client.DefaultRequestHeaders.Accept.Add(
                        new System.Net.Http.Headers.MediaTypeWithQualityHeaderValue("application/json"));
                    client.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Basic",
                                            Convert.ToBase64String(
                                                    System.Text.ASCIIEncoding.ASCII.GetBytes(
                                                            string.Format("{0}:{1}", "", personalaccesstoken))));
                    using (HttpResponseMessage response = await client.GetAsync(
                                            "https://dev.azure.com/{organization}/_apis/projects")) {
                                            response.EnsureSuccessStatusCode();
                                            string responseBody = await response.Content.ReadAsStringAsync();
                                            Console.WriteLine(responseBody);
                                    }
                    }
        } catch (Exception ex) {
                    Console.WriteLine(ex.ToString());
        }
}
```

DevOps resource center

- Centro de Recursos de Devops



Laboratorio

- Lab 12 Automating Selenium Tests in Azure Pipelines

Referencias

- https://docs.microsoft.com/en-us/cli/azure/
- https://docs.microsoft.com/en-us/azure/devops/learn/
- https://docs.microsoft.com/en-us/rest/api/azure/devops/?view=azure-devops-rest-6.1&viewFallbackFrom=azure-devops