Trabajo Práctico 8

Implementación de Contenedores en Azure y Automatización con Azure CLI

4- Desarrollo:

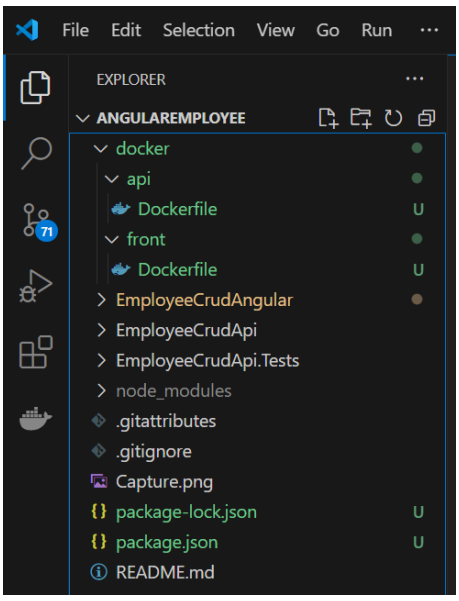
Prerequisitos:

- **Azure CLI instalado**

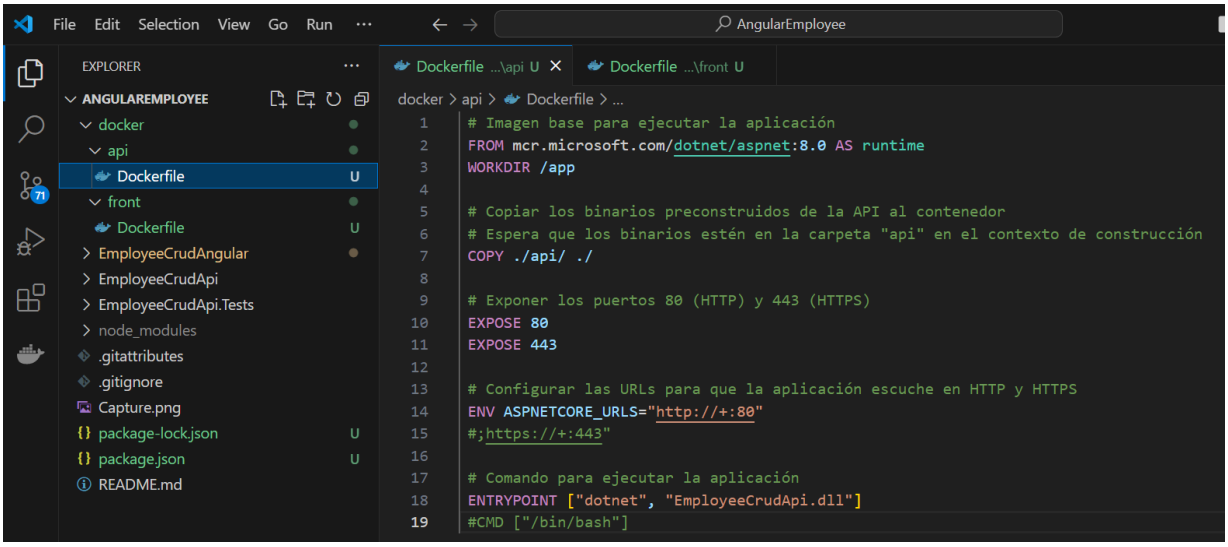
```
C:\Users\Usuario>ng version

     _                      _                 ____ _     ___
    / \   _ __   __ _ _   _| | __ _ _ __     / ___| |   |_ _|
   / △ \ | '_ \ / _` | | | | |/ _` | '__|   | |   | |    | |
  / ___ \| | | | (_| | |_| | | (_| | |      | |___| |___ | |
 /_/   \_\_| |_|\__, |\__,_|_|\__,_|_|       \____|_____|___|
                |___/


Angular CLI: 18.2.6
Node: 20.17.0
Package Manager: npm 10.8.2
OS: win32 x64

Angular:
...

Package                      Version
------------------------------------------------------
@angular-devkit/architect    0.1802.6 (cli-only)
@angular-devkit/core         18.2.6 (cli-only)
@angular-devkit/schematics   18.2.6 (cli-only)
@schematics/angular          18.2.6 (cli-only)
```

4.1 Modificar nuestro pipeline para construir imágenes Docker de back y front y subirlas a ACR

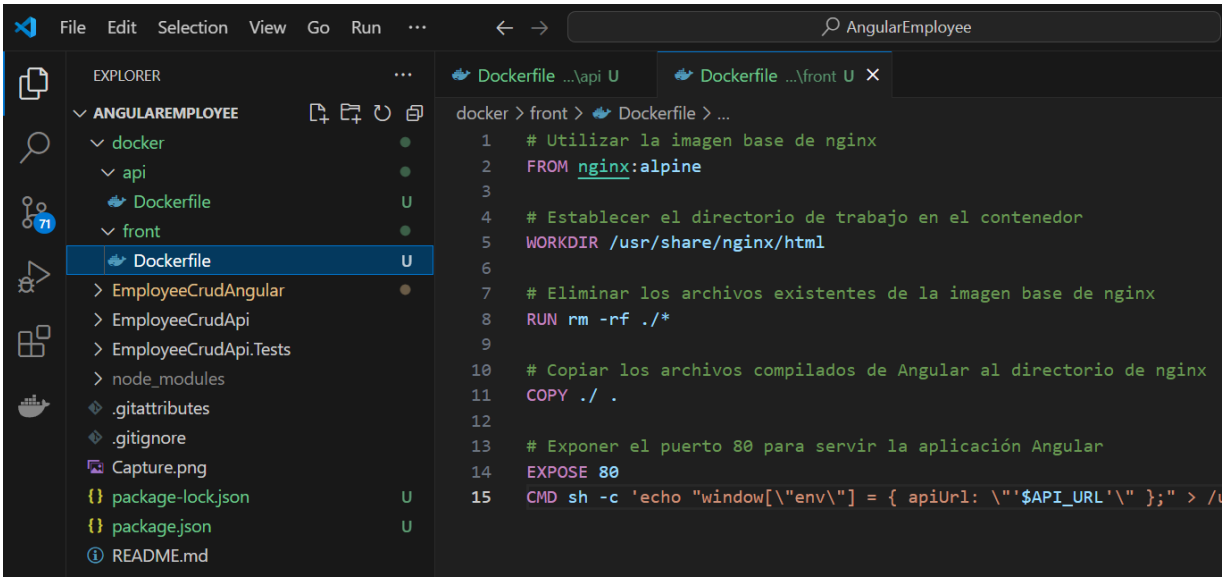
- **4.1.1 Crear archivos DockerFile para nuestros proyectos de Back y Front**
  - **En la raiz de nuestro repo crear una carpeta docker con dos subcarpetas api y front, dentro de cada una de ellas colocar los dockerfiles correspondientes para la creación de imágenes docker en función de la salida de nuestra etapa de Build y Test**



DockerFile Back:

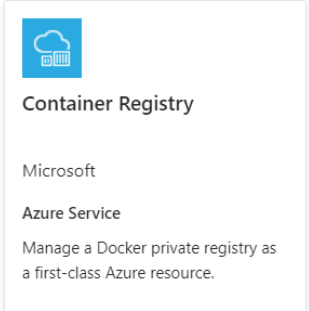


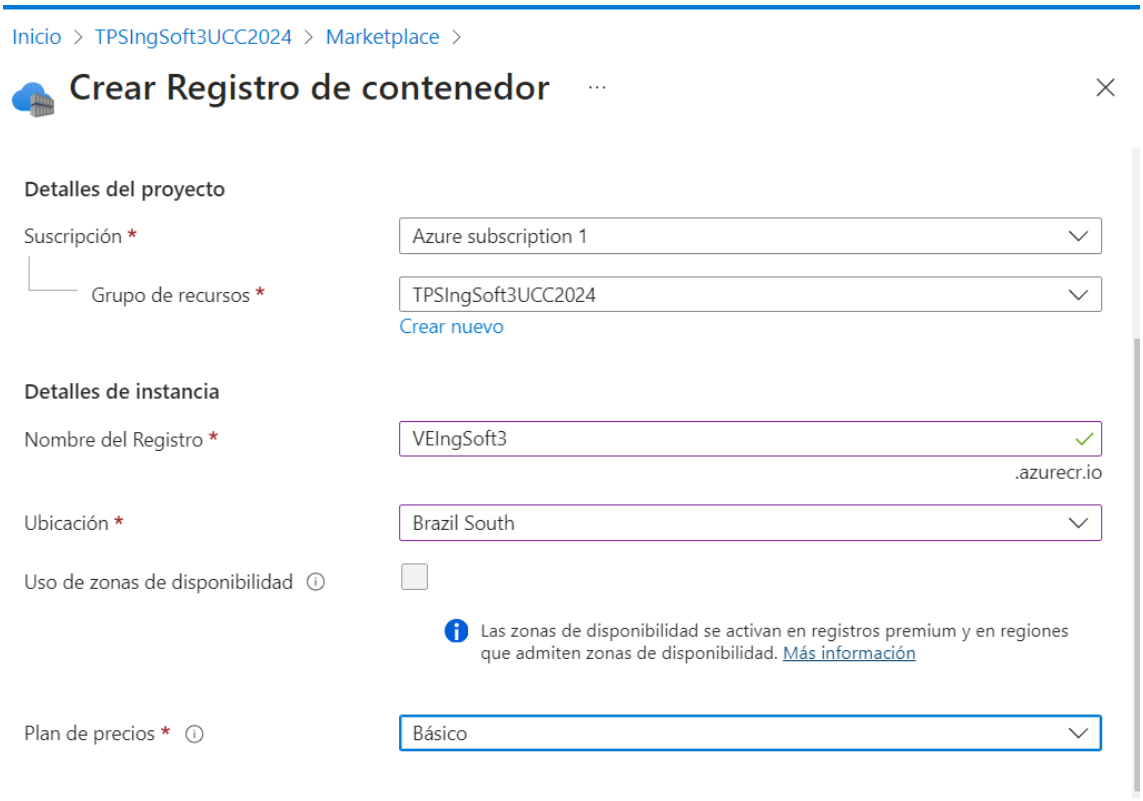
DockerFile Front:

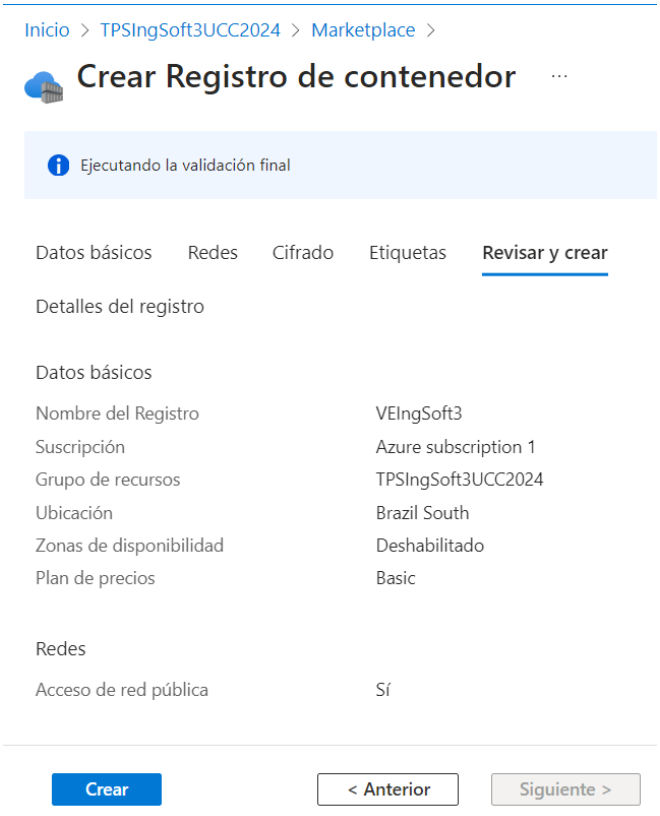


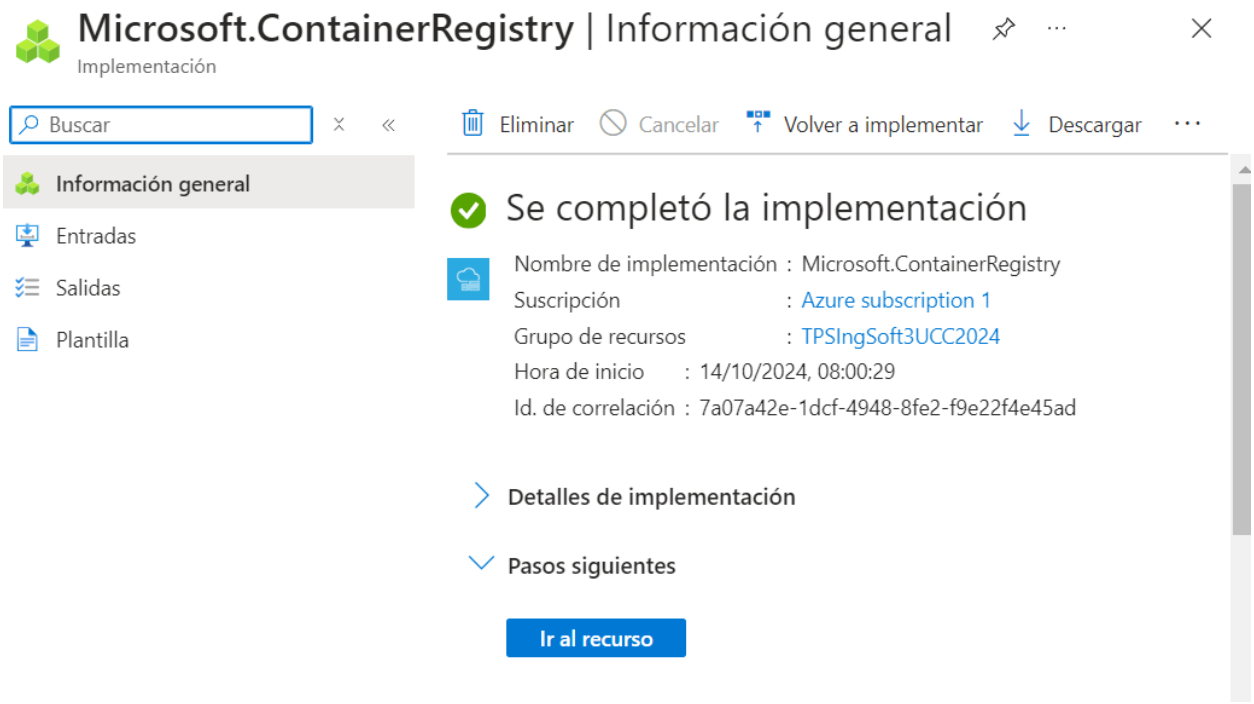
- **4.1.2 Crear un recurso ACR en Azure Portal siguiendo el instructivo 5.1**

container registry".

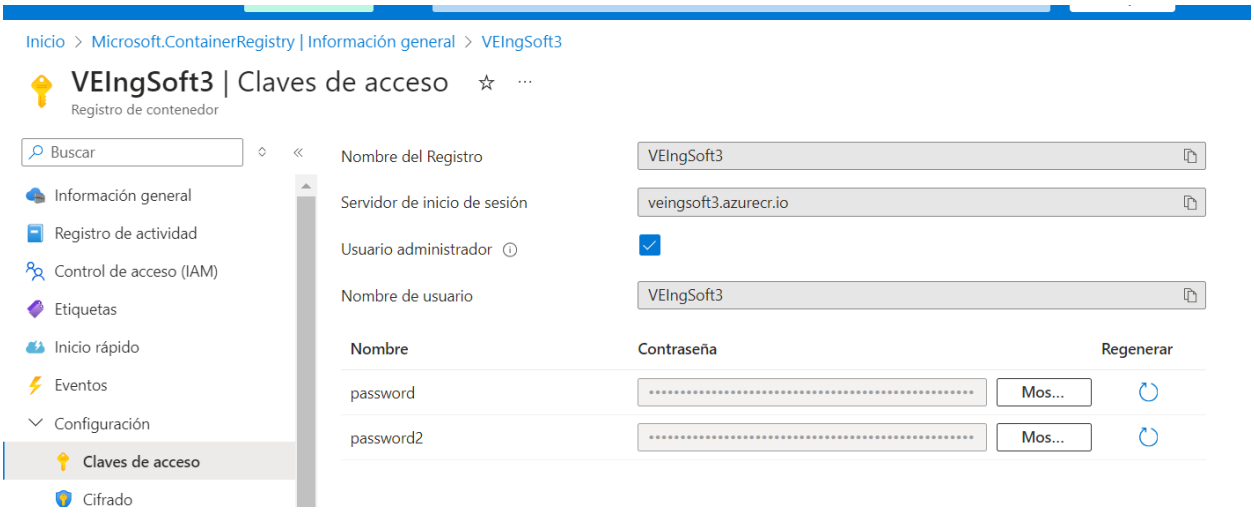



Inicio > TPSIngSoft3UCC2024 > Marketplace >
Crear Registro de contenedor
Detalles del proyecto
Suscripción *
Azure subscription 1
Grupo de recursos *
TPSIngSoft3UCC2024
Crear nuevo
Detalles de instancia
Nombre del Registro *
VEIngSoft3
.azurecr.io
Ubicación *
Brazil South
Uso de zonas de disponibilidad
Las zonas de disponibilidad se activan en registros premium y en regiones que admiten zonas de disponibilidad. Más información
Plan de precios *
Básico


Inicio > TPSIngSoft3UCC2024 > Marketplace >
Crear Registro de contenedor
Ejecutando la validación final
Datos básicos
Redes
Cifrado
Etiquetas
Revisar y crear
Detalles del registro
Datos básicos
Nombre del Registro VEIngSoft3
Suscripción Azure subscription 1
Grupo de recursos TPSIngSoft3UCC2024
Ubicación Brazil South
Zonas de disponibilidad Deshabilitado
Plan de precios Basic
Redes
Acceso de red pública Sí
Crear
< Anterior
Siguiente >



Copiar la url de nuestro recurso ACR: veingsoft3.azurecr.io



Mweq/t3nGningiAvjyqeiJCs35ur3/wf+5H6T6GXi6+ACRBTHUCe / password

u51WKamMCLGPI4VIQ7nPnr5FXKaDcUH63obHo6TjUr+ACRBHNpXE / password2

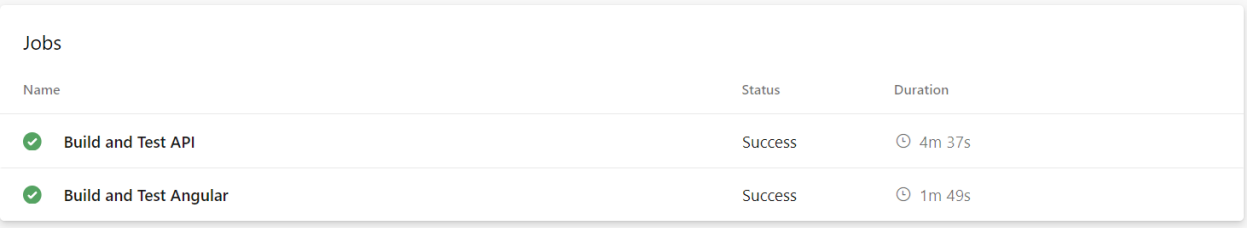
4.1.3 Modificar nuestro pipeline en la etapa de Build y Test

- **Luego de la tarea de publicación de los artefactos de Back agregar la tarea de publicación de nuestro dockerfile de back para que esté disponible en etapas posteriores:**

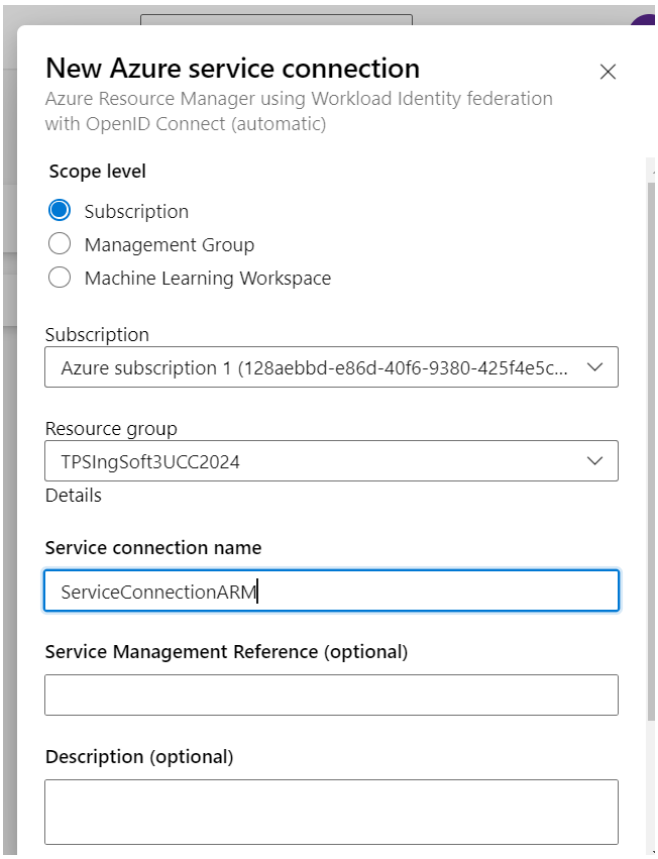
```
Settings
    - task: PublishBuildArtifacts@1
      displayName: 'Publicar artefactos de compilación'
      inputs:
        PathtoPublish: '$(Build.ArtifactStagingDirectory)'
        ArtifactName: 'api-drop'
        publishLocation: 'Container'

Settings
    - task: PublishPipelineArtifact@1
      displayName: 'Publicar Dockerfile de Back'
      inputs:
        targetPath: '$(Build.SourcesDirectory)/docker/api/dockerfile'
        artifact: 'dockerfile-back'

  - job: BuildAngular
    displayName: "Build and Test Angular"
    pool:
      vmImage: 'ubuntu-latest'
    steps:
Settings
    - task: NodeTool@0
      displayName: 'Instalar Node.js'
```

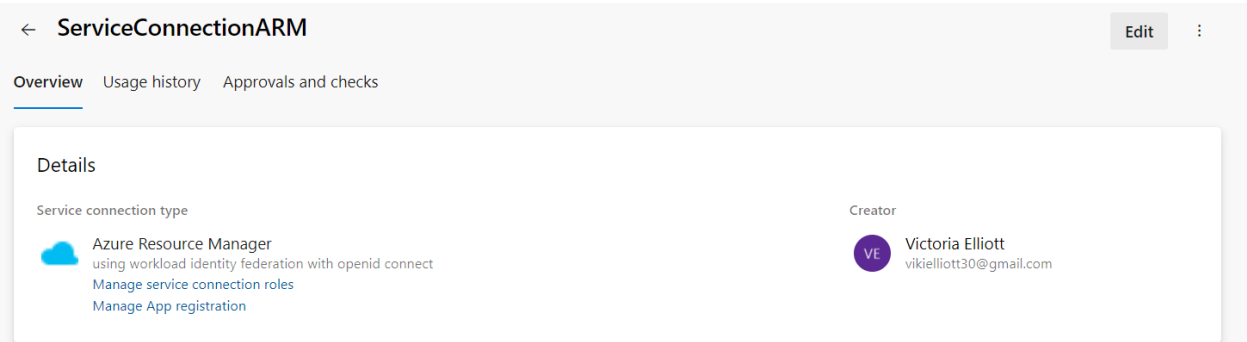
Luego de la tarea de publicación de los artefactos de Front agregar la tarea de publicación de nuestro dockerfile de front para que esté disponible en etapas posteriores:

```
Settings
    - task: CmdLine@2
      displayName: 'Compilar el proyecto Angular'
      inputs:
        script: npm run build
        workingDirectory: $(frontPath)
Settings
    - task: PublishBuildArtifacts@1
      displayName: 'Publicar artefactos Angular'
      inputs:
        PathtoPublish: '$(frontPath)/dist'
        ArtifactName: 'front-drop'

Settings
    - task: PublishPipelineArtifact@1
      displayName: 'Publicar Dockerfile de Back'
      inputs:
        targetPath: '$(Build.SourcesDirectory)/docker/front/dockerfile'
        artifact: 'dockerfile-front'
```



- **4.1.4 En caso de no contar en nuestro proyecto con una ServiceConnection a Azure Portal para el manejo de recursos, agregar una service connection a Azure Resource Manager como se indica en instructivo 5.2**





4.1.5 Agregar a nuestro pipeline variables

```yaml
trigger:
  branches:
    include:
    - main
pool:
  vmImage: 'windows-latest'

variables:
  solution: '**/*.sln'
  buildPlatform: 'Any CPU'
  buildConfiguration: 'Release'
  frontPath: './EmployeeCrudAngular'
  ConnectedServiceName: 'ServiceConnectionARM' #Por ejemplo 'ServiceConnectionARM'
  acrLoginServer: 'veingsoft3.azurecr.io' #Por ejemplo 'ascontainerregistry.azurecr.io'
  backImageName: 'employee-crud-api' #Por ejemplo 'employee-crud-api'

```

4.1.6 Agregar a nuestro pipeline una nueva etapa que dependa de nuestra etapa de Build y Test

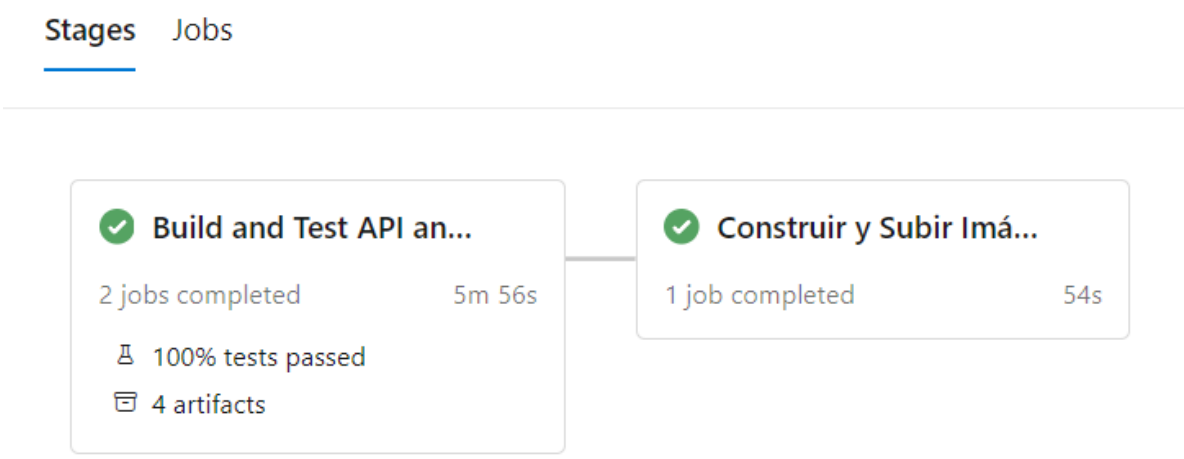
- **Agregar tareas para generar imagen Docker de Back**

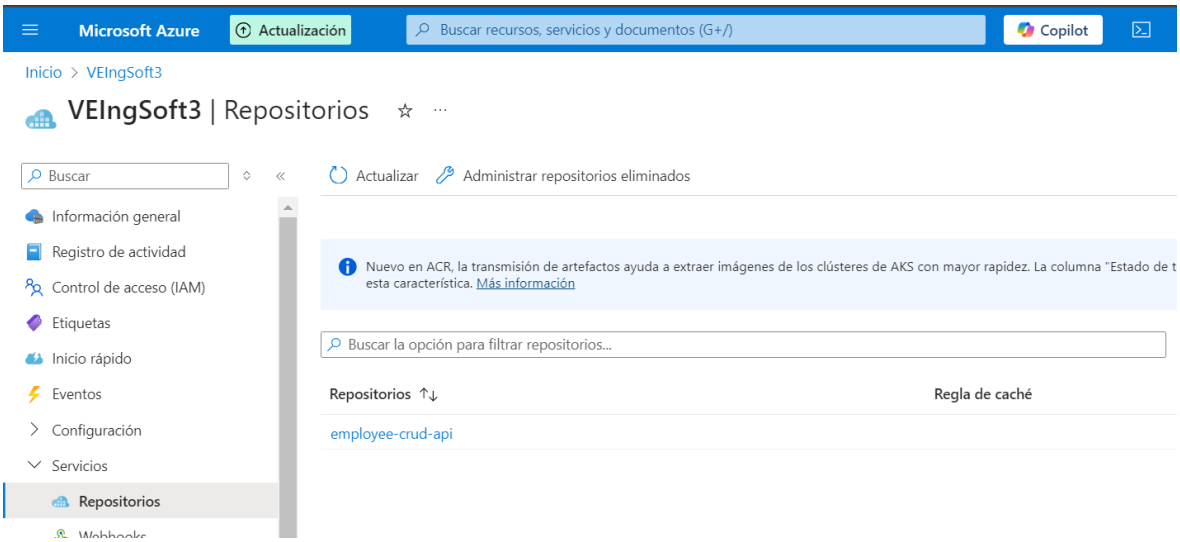
← tp7-angular

main ⌄ tp7-angular / azure-pipelines.yml *

```yaml
# #
# ### STAGE BUILD DOCKER IMAGES Y PUSH A AZURE CONTAINER REGISTRY
# #-----------------------------------------------------------

- stage: DockerBuildAndPush
  displayName: 'Construir y Subir Imágenes Docker a ACR'
  dependsOn: BuildAndTestBackAndFront #NOMBRE DE NUESTRA ETAPA DE BUILD Y TEST
  jobs:
    - job: docker_build_and_push
      displayName: 'Construir y Subir Imágenes Docker a ACR'
      pool:
        vmImage: 'ubuntu-latest'

      steps:
        - checkout: self

        #-----------------------------------------------------------
        # BUILD DOCKER BACK IMAGE Y PUSH A AZURE CONTAINER REGISTRY
        #-----------------------------------------------------------

Settings
        - task: DownloadPipelineArtifact@2
          displayName: 'Descargar Artefactos de Back'
          inputs:
            buildType: 'current'
```

4.1.7 - Ejecutar el pipeline y en Azure Portal acceder a la opción Repositorios de nuestro recurso Azure Container Registry. Verificar que exista una imagen con el nombre especificado en la variable backImageName asignada en nuestro pipeline



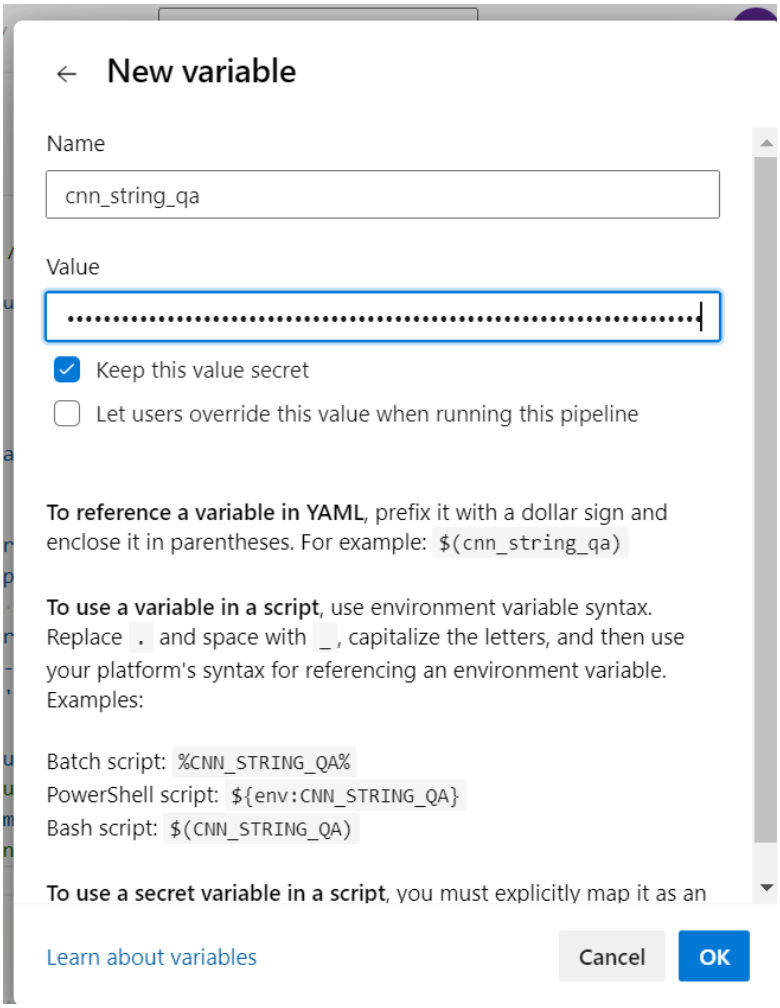


4.1.9 - Agregar a nuestro pipeline una nueva etapa que dependa de nuestra etapa de Construcción de Imagenes Docker y subida a ACR

- Agregar variables a nuestro pipeline:

```
27    - name: 'ResourceGroupName'
28      value: 'VEIngSoft3' #Por ejemplo 'TPS_INGSOFT3_UCC'
29    - name: 'backContainerInstanceNameQA'
30      value: 've-crud-api-qa' #Por ejemplo 'as-crud-api-qa'
31    - name: 'backImageTag'
32      value: 'latest'
33    - name: 'container-cpu-api-qa'
34      value: 1 #CPUS de nuestro container de QA
35    - name: 'container-memory-api-qa'
36      value: 1.5 #RAM de nuestro container de QA
```

- Agregar variable secreta cnn-string-qa desde la GUI de ADO que apunte a nuestra BD de SQL Server de QA como se indica en el instructivo 5.3



Server=tcp:ve-sql-server01.database.windows.net,1433;Initial Catalog=ve-sql-db;Persist Security Info=False;User ID=sqladmin;Password=Azure@456;MultipleActiveResultSets=False;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;

Tenemos que modificar program.cs para que lea la variable de entorno y use esa cadena de conexión

Program.cs

Edit

Contents History Compare Blame

Committed b0ba7997: Updated Program.cs

Create a pull request

```
using EmployeeCrudApi.Data;
using Microsoft.EntityFrameworkCore;
using Microsoft.Extensions.Configuration;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.
builder.Services.AddCors(o => o.AddPolicy("MyPolicy", builder =>
{
    builder.AllowAnyOrigin()
           .AllowAnyMethod()
           .AllowAnyHeader();
}));

// Obtener la cadena de conexión de una variable de entorno o de la configuración
var connectionString = Environment.GetEnvironmentVariable("cnn_string_qa")
                       ?? builder.Configuration.GetConnectionString("DefaultConnection");

if (string.IsNullOrEmpty(connectionString))
{
    throw new InvalidOperationException("La cadena de conexión no está configurada.");
}
```

Agregar tareas para crear un recurso Azure Container Instances que levante un contenedor con nuestra imagen de back

```
        #-------------------------------------------------------
        # DEPLOY DOCKER BACK IMAGE A AZURE CONTAINER INSTANCES QA
        #-------------------------------------------------------

    Settings
    - task: AzureCLI@2
      displayName: 'Desplegar Imagen Docker de Back en ACI QA'
      inputs:
        azureSubscription: '$(ConnectedServiceName)'
        scriptType: bash
        scriptLocation: inlineScript
        inlineScript: |
          echo "Resource Group: $(ResourceGroupName)"
          echo "Container Instance Name: $(backContainerInstanceNameQA)"
          echo "ACR Login Server: $(acrLoginServer)"
          echo "Image Name: $(backImageName)"
          echo "Image Tag: $(backImageTag)"
          echo "Connection String: $(cnn-string-qa)"

          az container delete --resource-group $(ResourceGroupName) --name $(backContainerInstanceNameQA) --yes

          az container create --resource-group $(ResourceGroupName) \
            --name $(backContainerInstanceNameQA) \
            --image $(acrLoginServer)/$(backImageName):$(backImageTag) \
```

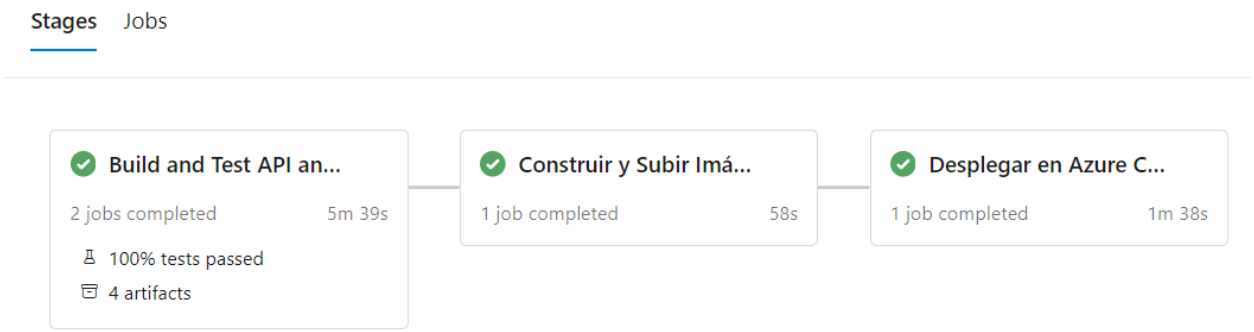
Para que funcione, hay que habilitar el acceso administrativo de ACR desde azure CLI

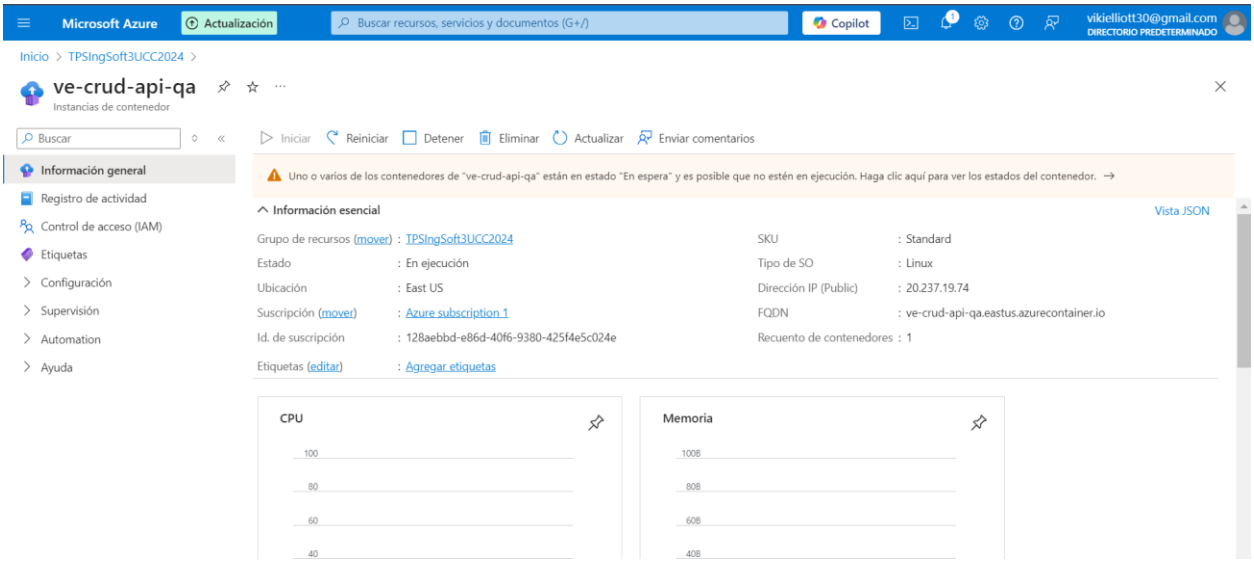
```
PS C:\Users\Usuario> az acr update --name VEIngSoft3 --resource-group TPSIngSoft3UCC2024 --admin-enabled true
{
  "adminUserEnabled": true,
  "anonymousPullEnabled": false,
  "creationDate": "2024-10-14T23:00:31.934702+00:00",
  "dataEndpointEnabled": false,
  "dataEndpointHostNames": [],
  "encryption": {
    "keyVaultProperties": null,
    "status": "disabled"
  },
  "id": "/subscriptions/128aebbd-e86d-40f6-9380-425f4e5c024e/resourceGroups/TPSIngSoft3UCC2024/providers/Micros
VEIngSoft3",
  "identity": null,
  "location": "brazilsouth",
  "loginServer": "veingsoft3.azurecr.io",
  "metadataSearch": "Disabled",
```

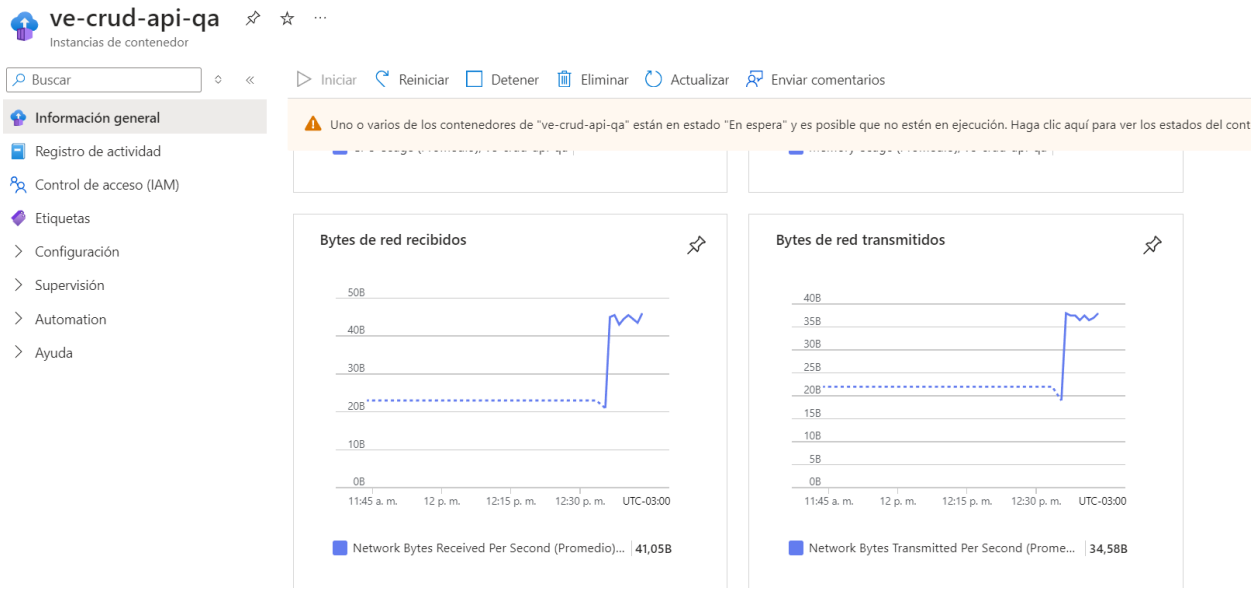
Además, debemos registrar el proveedor de recursos Microsoft.ContainerInstance, que es necesario para crear instancias de contenedores en Azure.

```
PS C:\Users\Usuario> az provider register --namespace Microsoft.ContainerInstance
Registering is still on-going. You can monitor using 'az provider show -n Microsoft.Con
tainerInstance'
```

4.1.10 - Ejecutar el pipeline y en Azure Portal acceder al recurso de Azure Container Instances creado. Copiar la url del contenedor y navegarlo desde browser. Verificar que traiga datos







ve-crud-api-qa.eastus.azurecontainer.io

Pretty-print

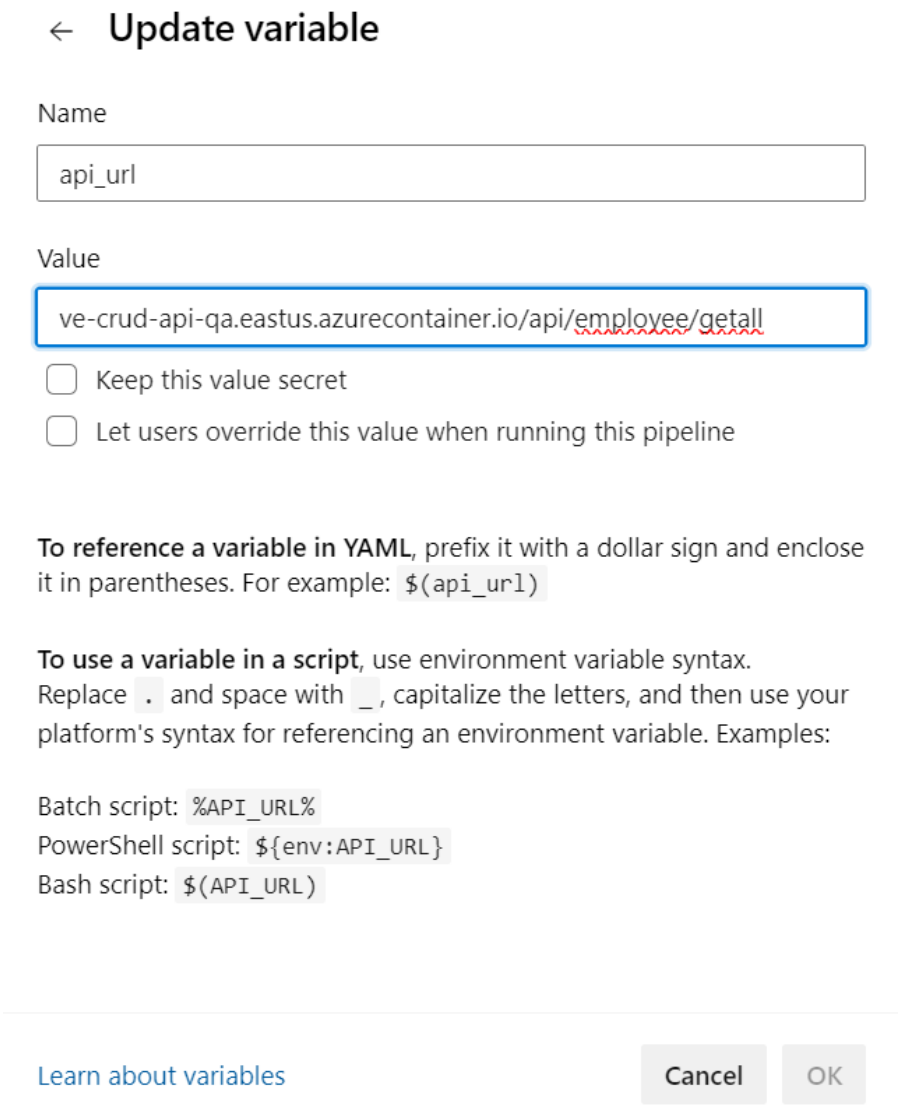

```
[
  {
    "id": 1,
    "name": "Juan Perez",
    "createdDate": "2024-09-05T00:00:00"
  },
  {
    "id": 2,
    "name": "Carla Ruiz",
    "createdDate": "2024-09-05T22:22:47.440572"
  },
  {
    "id": 3,
    "name": "Carlos Gomez",
    "createdDate": "2024-09-06T09:16:50.070943"
  },
  {
    "id": 4,
    "name": "Joaquin Zarate",
    "createdDate": "2024-09-06T09:19:37.898716"
  },
  {
    "id": 5,
    "name": "Luis Rodriguez",
    "createdDate": "2024-09-06T09:23:31.324434"
  }
]
```


- **4.1.11 - Agregar tareas para generar un recurso Azure Container Instances que levante un contenedor con nuestra imagen de front (DESAFIO)**

```
19    backImageName: 'employee-crud-api'
20    frontImageName: 'employee-crud-front'
```

  - A la etapa creada en 4.1.9 Agregar tareas para generar contenedor en ACI con nuestra imagen de Front
    - Tener en cuenta que el contenedor debe recibir como variable de entorno API_URL el valor de una variable container-url-api-qa definida en nuestro pipeline.

- Para que el punto anterior funcione el código fuente del front debe ser modificado para que la url de la API pueda ser cambiada luego de haber sido construída la imagen. Se deja un ejemplo de las modificaciones a realizar en el repo https://github.com/ingsoft3ucc/CrudAngularConEnvironment.git



ve-crud-api-qa.eastus.azurecontainer.io/api/employee/getall

environment.ts

Comm

Contents | Highlight changes

```ts
// This file can be replaced during build by using the `fileReplacements` array.
// `ng build --prod` replaces `environment.ts` with `environment.prod.ts`.
// The list of file replacements can be found in `angular.json`.

export const environment = {
  production: false,
  apiUrl: 've-crud-api-qa.eastus.azurecontainer.io/api/employee'  // URL de la API para desarrollo
};

```

environment.prod.ts

Contents | History | Compare | Blame

Committed 0a9a7b92: Updated environment.prod.ts

Crea

```ts
export const environment = {
  production: true,
  apiUrl: (typeof window !== 'undifined' && window.env.apiUrl)
            ? window.env.apiUrl
            : 've-crud-api-qa.eastus.azurecontainer.io/api/employee'  // URL de la API para prod
};

```

test.ts

Commit

Contents | Highlight changes

Committed a50b1710: Updated environment.prod.ts

Create a pull r

```ts
// This file is required by karma.conf.js and loads recursively all the .spec and framework files

(window as any)['env'] = (window as any)['env'] || {};
(window as any)['env']['apiUrl'] = (window as any)['env']['apiUrl'] || 'http://localhost:7150/api/employee';

import 'zone.js/dist/zone-testing';
import { getTestBed } from '@angular/core/testing';
import {
  BrowserDynamicTestingModule,
  platformBrowserDynamicTesting
} from '@angular/platform-browser-dynamic/testing';

declare const require: {
  context(path: string, deep?: boolean, filter?: RegExp): {
    keys(): string[];
```

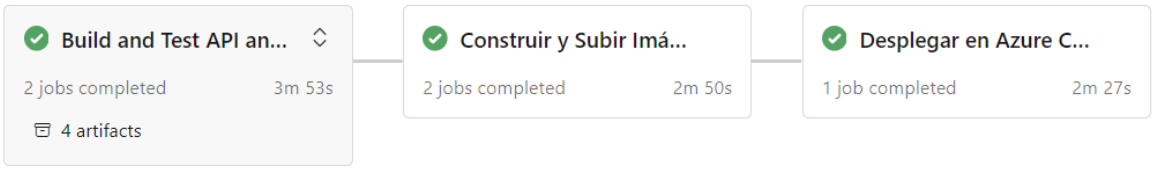
index.html

Contents | Preview | Highlight changes

Committed 5f3db474: Updated test.ts

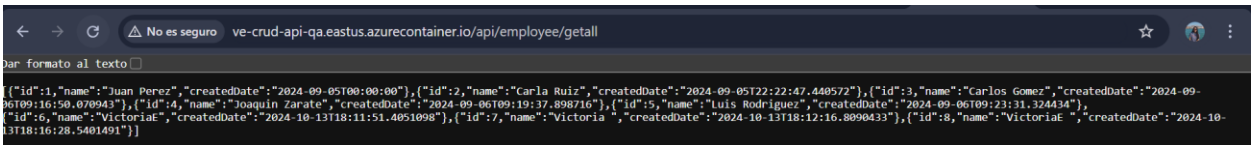
```html
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>EmployeeCrudAngular</title>
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
</head>
<script src="assert/env.js"></script>
<script>
  console.log('Valor de api-ulr desde env.js:', window['env'].apiUrl);
</script>
<body>
  <app-root></app-root>
</body>
</html>

```

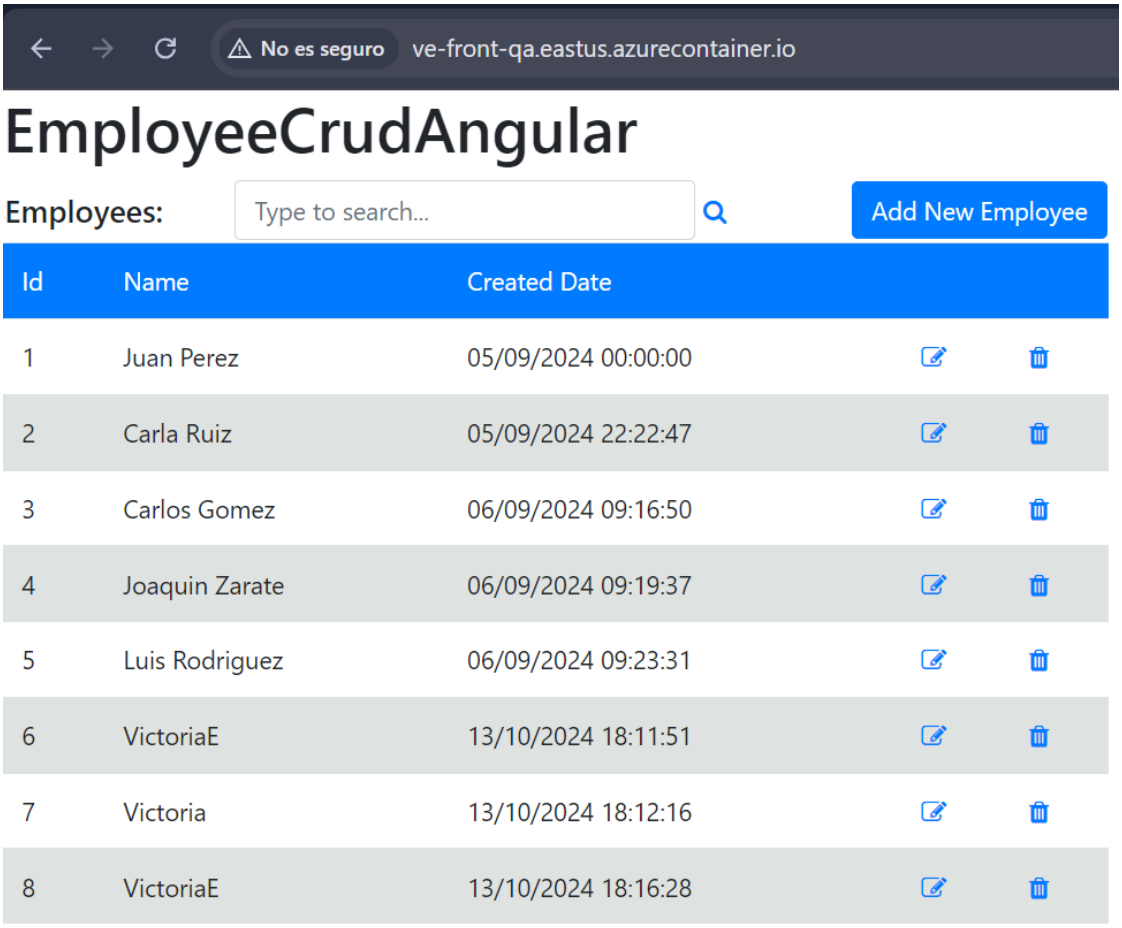
Stages | Jobs



Nombre

- MiEjemplo01-PROD
- MiPlanTP05
- ve-crud-api-qa
- ve-front-qa
- ve-sql-db (ve-sql-server01/ve-sql-db)
- ve-sql-server01
- VEIngSoft3

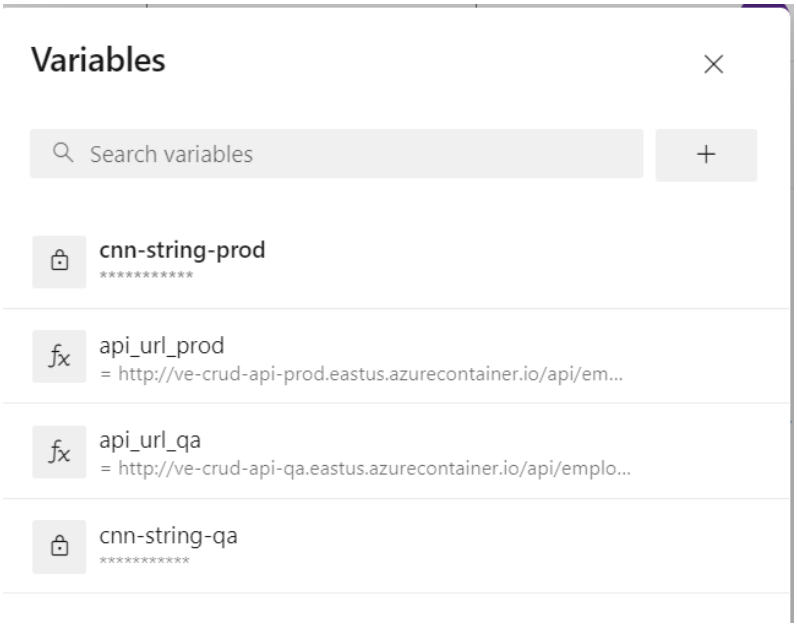




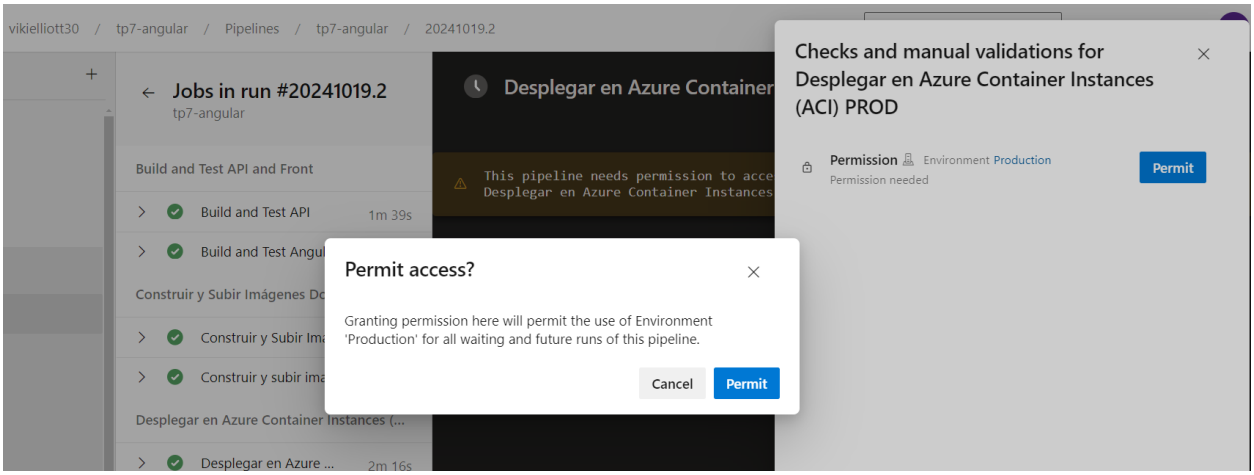
- **4.1.12 - Agregar tareas para correr pruebas de integración en el entorno de QA de Back y Front creado en ACI.**

4.2 Desafíos:

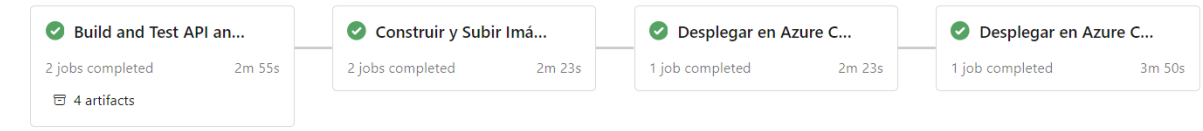
- 4.2.1 Agregar tareas para generar imagen Docker de Front. (Punto 4.1.8) -> COMPLETADO
- 4.2.2 Agregar tareas para generar en Azure Container Instances un contenedor de imagen Docker de Front. (Punto 4.1.11) -> COMPLETADO
- 4.2.3 Agregar tareas para correr pruebas de integración en el entorno de QA de Back y Front creado en ACI. (Punto 4.1.12)
- 4.2.4 Agregar etapa que dependa de la etapa de Deploy en ACI QA y genere contenedores en ACI para entorno de PROD. -> COMPLETADO

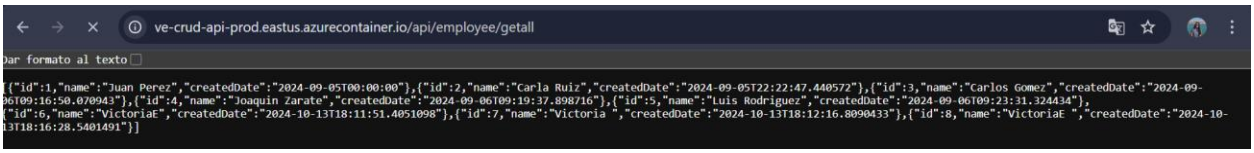



```
32
33  backContainerInstanceNameProd: 've-crud-api-prod'
34  container-cpu-api-prod: 1 #CPUS de nuestro container de PROD
35  container-memory-api-prod: 1.5 #RAM de nuestro container de PROD
36
37  frontContainerInstanceNameQA: 've-front-qa'
38  frontImageTag: 'latest'
39  container-cpu-front-qa: 1
40  container-memory-front-qa: 1.5
41
42  frontContainerInstanceNameProd: 've-front-prod'
43  container-cpu-front-prod: 1
44  container-memory-front-prod: 1.5
```

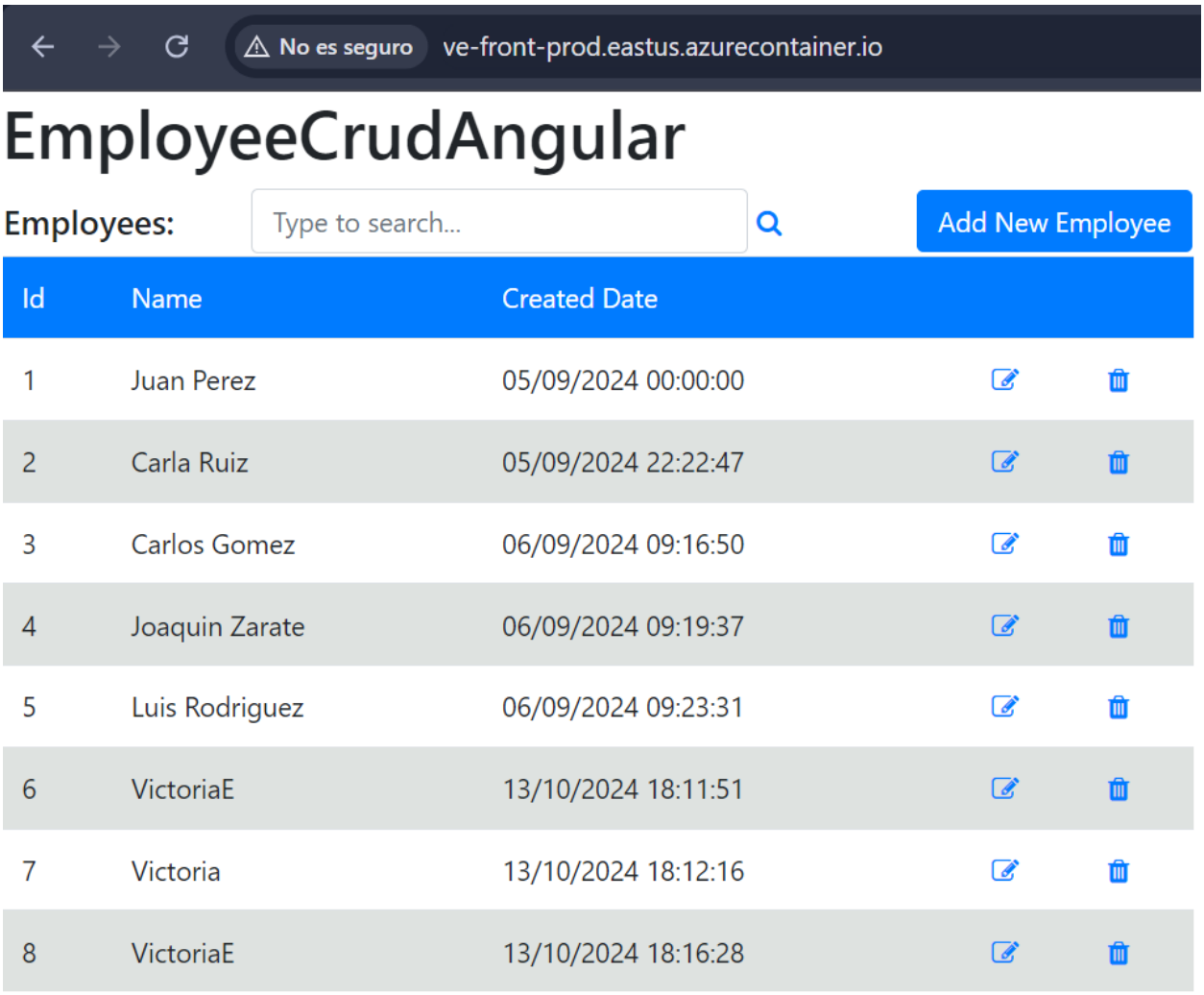


Stages Jobs


Build and Test API an...
2 jobs completed 2m 55s
4 artifacts
Construir y Subir Imá...
2 jobs completed 2m 23s
Desplegar en Azure C...
1 job completed 2m 23s
Desplegar en Azure C...
1 job completed 3m 50s


ve-crud-api-prod.eastus.azurecontainer.io/api/employee/getall
Dar formato al texto
[{"id":1,"name":"Juan Perez","createdDate":"2024-09-05T00:00:00"},{"id":2,"name":"Carla Ruiz","createdDate":"2024-09-05T22:22:47.440572"},{"id":3,"name":"Carlos Gomez","createdDate":"2024-09-06T09:16:50.070943"},{"id":4,"name":"Joaquin Zarate","createdDate":"2024-09-06T09:19:37.898716"},{"id":5,"name":"Luis Rodriguez","createdDate":"2024-09-06T09:23:31.324434"},{"id":6,"name":"VictoriaE","createdDate":"2024-10-13T18:11:51.4051098"},{"id":7,"name":"Victoria ","createdDate":"2024-10-13T18:12:16.8090433"},{"id":8,"name":"VictoriaE ","createdDate":"2024-10-13T18:16:28.5401491"}]


ve-crud-api-qa
ve-front-prod


No es seguro
ve-front-prod.eastus.azurecontainer.io
EmployeeCrudAngular
Employees:
Type to search...
Add New Employee
Id
Name
Created Date
1 Juan Perez 05/09/2024 00:00:00
2 Carla Ruiz 05/09/2024 22:22:47
3 Carlos Gomez 06/09/2024 09:16:50
4 Joaquin Zarate 06/09/2024 09:19:37
5 Luis Rodriguez 06/09/2024 09:23:31
6 VictoriaE 13/10/2024 18:11:51
7 Victoria 13/10/2024 18:12:16
8 VictoriaE 13/10/2024 18:16:28