\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Configuracion de Taller de DEVOPS**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Elaborado por:**

Pedro Abel

# Objetivo del documento

Este documento tiene como alcance detallar la infraestructura del taller de Devops, su alcance y las configuraciones necesarias para la replicación del escenario.

# Alcance

Tiene como alcance el poder replicar el escenario inicial y detallar las diferentes estructuras utilizadas durante el taller.

# Insumos

Se detallan URL de consumos:

Tabla - matriz de enlaces

|  |  |
| --- | --- |
| Enlace | Descripción |
| <https://apim-tallerdevops.azure-api.net/Devops> | Consumo de apis productivas balanceadas desde el API Management |
| <https://pro2-taller-devops.azurewebsites.net/Devops> | Consumo de API directamente en el recurso publicado en Internet, se podría cambiar el 2 por el 1 para consumir el primer API |
| * <https://apim-tallerdevops.azure-api.net/health/readiness> * <https://apim-tallerdevops.azure-api.net/health/liveness> | Se dejan apis de estabilidad del servicio. |
| <https://github.com/ElAngeluz/Taller_Devops> | Repositorio de código fuente |
| [Pipelines - Recent (azure.com)](https://dev.azure.com/ansofttech/Taller_Devops/_build) | Repositorio de Pipelines. |

# Arquitectura de la implementación

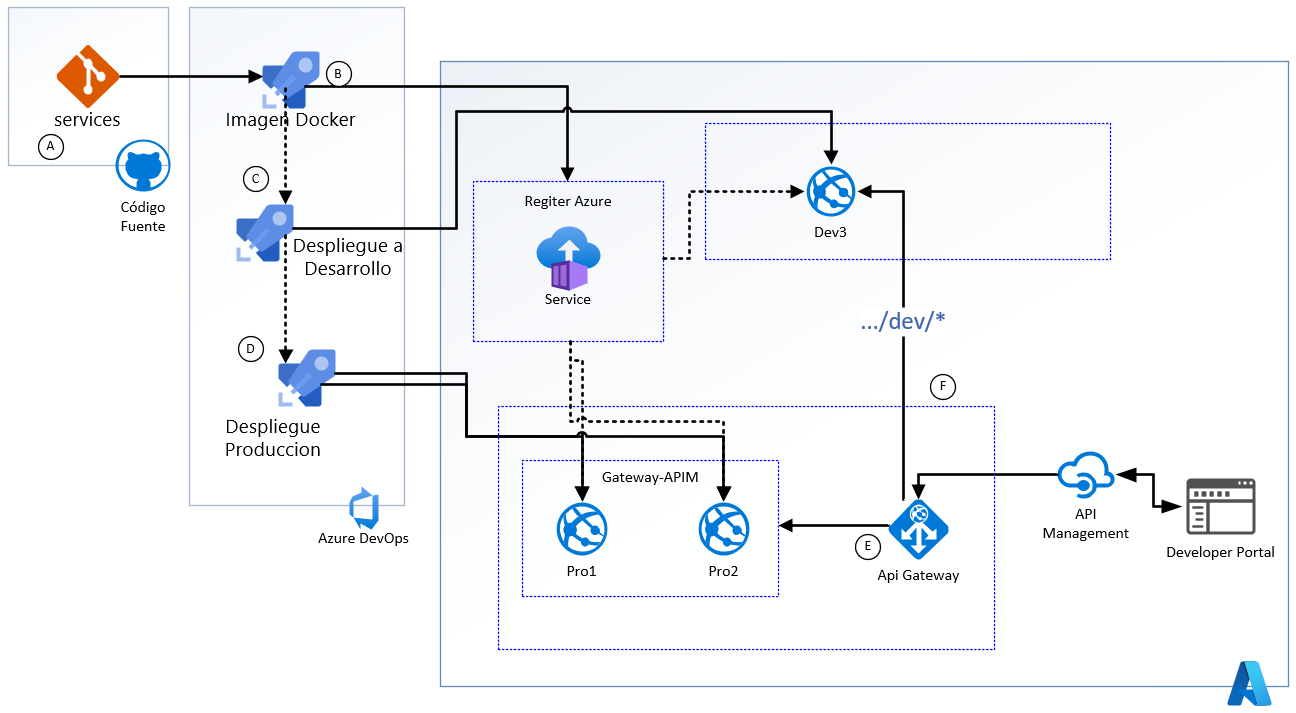


Ilustración - arquitectura de la implementación

## Detalle del proceso

El despliegue de la aplicación consta de 5 fases antes de ponerla en producción:

1. El código fuente es almacenado en el repositorio en github, donde se tiene dos troncales principales, main y desarrollo.
2. cuando se realiza la incorporación de cambios en las troncales principales se genera una imagen Docker según el ambiente:
   1. Desarrollo: se procede a generar una imagen en el registro de repositorio en Azure con el postfijo: -DEV y como tag el numero de la versión de compilación.
   2. Main: mismo proceso de desarrollo y se genera otra imagen con la etiqueta “LATEST”.
3. Una vez que la imagen en desarrollo se coloco en el registro de repositorio, se procede al despliegue continuo en el ambiente de desarrollo.
4. Cuando se realiza la incorporacion de cambios en la rama main entonces se hace el despliegue continuo hacia los 2 servidores en producción.
5. El api management adminsitra la entrada hacia el path de producción para la consulta de las APIS balanceadas
6. El API management administra la entrada hacia el path de desarrollo con el prefijo en la ruta dev.

# Integración y Despliegue continua

Se deja el Pipeline de CI [Azure-pipeline](https://github.com/ElAngeluz/Taller_Devops/blob/main/azure-pipelines.yml) y el archivo [taller-devop-CD](https://github.com/ElAngeluz/Taller_Devops/blob/main/Taller-devops-CD.json)

Para el despliegue se tienen los siguientes stage:

Interfaz de usuario gráfica

Descripción generada automáticamente

Ilustración - STAGES DE TALLER DEVOPS

A pesar de que se tengan secuencialmente unidos se tiene un filtro de separación para desplegar en los servicios de Producción:

