1. Introducción
   1. Contexto
   2. Actores
   3. Definiciones
   4. El problema
2. Justificación
3. Alcance
   1. Objetivos
   2. Requerimientos
   3. Riesgos
4. Metodología
5. Planificación temporal
   1. Planificación Inical
   2. Planificación Actual
6. Anàlisi de technologies
   1. Pipelines
   2. Azure
      1. Function App
      2. Web App
      3. Azure kubernetes service
      4. Azure sql
   3. Monitorización
7. Sistema de ficheros
8. Ejemplo
9. Gestión económica
   1. Presupuesto
   2. Control de gestión
10. Sostenibilidad
11. Bibliografía

**Trabajo hecho:**

En este apartado vamos a detallar en las tecnologías que se han usado para este trabajo, y como se han usado para poder terminar el proyecto.

**Bicep**

En este trabajo usamos bicep para definir toda la infraestructura de los diferentes recusos.

Principalmente tenemos un fichero principal que se llama main.bicep que es el fichero que se va a ejecutar y después tenemos otros ficheros dentro de la carpeta bicep-templates, que allá están definidos las plantillas de todos los recursos que vamos a usar, y de este forma si un recuso lo vamos a usar más de una vez, entonces no hace falta definirlo más de una vez, solamente con llamarlo las veces que hace falta con pasar los parámetros ya lo tendremos

Los nombres de los ficheros como los recusos siguen las abreviaturas que definen Microsoft para azure :

App Service Plan: plan

Function app: func

Web app: app

Log Analytics Workspace: log

Application Insights: appi

Azure SQL database: sqldb

Azure SQL Database server: sql

Azure Kubernetes Service: aks

Azure Container Registr: acr

Storage Account: st

**Function App**

El Function App en este proyecto sirve como un entorno de ejecución para que el usuario pone sus códigos que quiere que se ejecuten en Python versión 3.9, el usuario solamente pone su código y también las librarías y las versiones que use, y Azure se encarga del resto.

La manera que el usuario tiene para poder lanzar su código dentro del Function App, es usando el URL del Function, y el código se ejecutará y responderá con el resultado esperado

El mismo Function App lo podemos usar para poner más de un código y que todos se ejecuten de forma separada (cada uno se ejecuta con su URL).

Para que el usuario pueda monetizar sus códigos, el Function App se le crea previamente un Application Insights y este Application Insights se conecta con un Log Analytics para que el usuario puede ver todos los logs (excepciones, errores, warning, etc) y las métricas(la memoria disponible, los procesos del CPU, las requestas, etc), en dos modos, en modo directo y también se puede ver las dadas de fechas anteriores que están guardadas.

storage accountPara poder crear el Azure App, previamente necesitamos crear un Storage Account, como un lugar para guardar los códigos que se tiene que ejecutar.