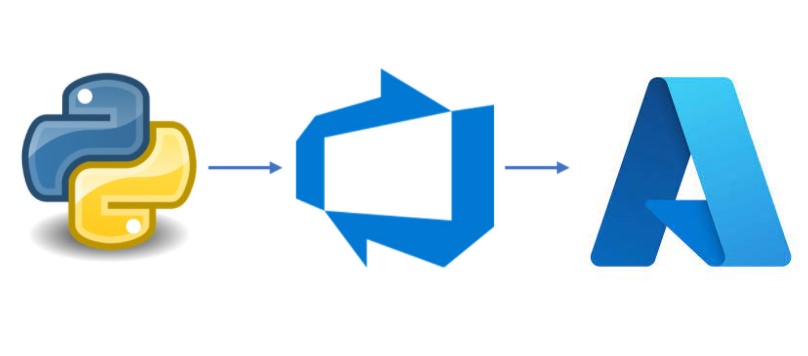
**Desplegar aplicaciónes Python de forma muy sencilla usando Azure**



# Facultad de Informática de Barcelona Grado en Ingeniería

Informática Especialidad de tecnología de la información

25-2-2022

Tutor: Omar Mohamed Mohamed Mahmoud Elkassar

Director: René Serral Gracià

Director GEP: Olga Pons Peregort

**INDEX**

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. **Descripción de las tareas**
   2. **Recursos**
   3. **Planificación de riesgos**
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**

# 1. Introducción

Hoy en día cualquiera proyecto informático que sea grande o pequeño o que tenga la mida que tenga, casi seguro, que este proyecto se le ha hecho un proceso de despliegue antes para que obtenga su forma final en producción.

Este proceso de despliegue hay una gran probabilidad que se ha hecho en una plataforma en la nube, en alguna máquina virtual o en algún servicio que ofrece la nube, que está dedicado este servicio a la ejecución, como el servicio de Azure Service App o Function Apps que los ofrece Azure.

## 1.1. Contexto

Dentro del Grado de Ingeniería Informática en la facultad de informática de Barcelona (FIB), hay cuatro especialidades, especialidad de computación, especialidad de tecnología de la información, especialidad de sistema de la información y finalmente especialidad de software.

Este trabajo de fin de grado pertenece a la especialidad de la tecnología de la información.

El trabajo consistirá en ayudar los proyectos pequeños y medianos en cómo obtener una infraestructura dentro de la nube de Azure para tener en marcha sus aplicaciones que están corriendo en Python, sin tener que invertir mucho esfuerzo.

## 1.2. Actores

Aquí hablaremos sobre los principales actores que se van a aprovechar del proyecto, y principalmente son los desenrollados que tienen una aplicación en Python y quieren desplegar esta aplicación y hacer toda la faena necesaria para el proceso del despliegue.

Este proyecto los ayuda a hacer el despliegue de una forma mucho más fácil y rábida.

## 1.3. Definiciones

En este apartado vamos a estudia los conceptos importantes para poder seguir profundizando en el trabajo.

### Nube

La nube se considera uno de los conceptos más importantes hoy en dia en el mundo informático.

La nube es no tener el código de tus aplicaciones corriendo en tus servidores locales, sino que tú tienes un servicio alquilado con alguna empresa para alquiler la capacidad de computación y tener tus aplicaciones corriendo allá.

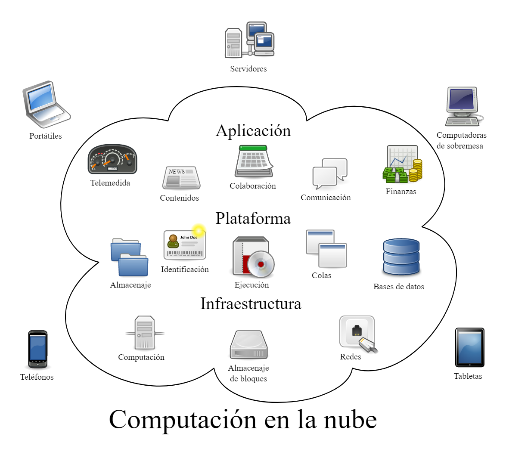


Figure 1: Presentación gráfica de la definición de la nube

### Azure

Es el servicio de nube que ofrece la compaña Microsoft, y se considera uno de los servicios de nube más grandes del mundo, Azure contiene el 20 porciento del mercado global de la computación en la nube.



Figure 2: Logo de la nube de Azure

### Function App

Es un servicio que ofrece Azure que permite a los desarrolladores ejecutar sus códigos sin tener que preocupar de nada, Azure gestiona todo lo necesario para que el código se ejecuta, y también Azure ofrece herramientas al usuario para que el usuario pueda determinar cuándo se tiene que ejecutar el codigo.



Figure 3: Logo de Function App

### Azure Service App

Es una hermanita de Azure que permita crear aplicaciones webs de forma muy rápida y sencilla, en que solo se tiene que proporcionar el cliente del código de la aplicación web, y Azure gestiono el resto.



Figure 4: Logo de Azure Service App

### Kubernetes (K8s)

Es una tecnología que permite crear microservicios y orquestarlos de forma que estos microservicios sean siempre disponibles y funcionales de la manera más optima posible.



Figure 5: Logo de Kubernetes

### Azure Kubernetes Service (AKS)

Es el servicio de Kuberntes que ofrece la nube de Azure, Azure tiene dos modalidades para usar este servicio, la primera Azure te deja la infraestructura fisica de Kubernetes ya creada y tú tienes que gestionarla y desplegar tus microservicios allá, o la segunda modalidad que tu solamente tienes que desplegar tus microservicios, y Azure se encarga del resto. 

Figure 6: Logo de Azure Kubernetes Service (AKS)

**App Service plan:**

Este recurso básicamente presenta la infraestructura física que se utiliza para correr el Function App y la Web App y el usuario normalmente puede decidir las características físicas de la infraestructura (de cuanto memoria y cpu dispone el recuso).



Figure 7: El logo deL App Service Plan

**Bicep**

Bicep es un lenguaje declarativo (que se declara lo que se quiere conseguir, no como se va a conseguir), que se usa para crear recursos en Azure, y de este forma si pasa cualquier problema con la infraestructura y se quiere volver a levantar toda la infraestructura de cero, se hace con solamente volver a ejecutar el código de bicep.

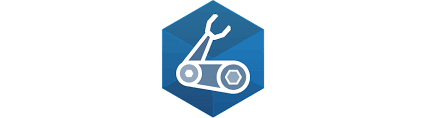


Figure 8: El logo de Bicep

### Azure Repos

Son los repositorios donde se guarden los códigos de las aplicaciones, para que después se usen en el despliegue.



Figure 9: Logo de Azure repos

**Pipelines**

Es una metodología para definir el ciclo de vida de las aplicaciones, de forma que todos los procesos que definen el ciclo de vida de una aplicación (compilar, test, desplegar, etc) se hacen de forma automática mediante código.

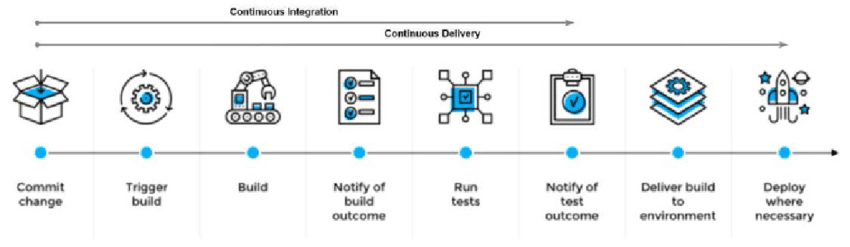


Figure 10: La arquitectura que normalmente forma un pipeline

### Azure Pipelines

Es la hermanita de pipeline que proporciona Azure.



Figure 11: Logo de Azure pipelines

**Azure Sql servers**

Son servidores sql que los proporciona Azure, que dentro de estos servidores se puede gestionar más de una base de datos, que todos van a están en la misma localización geográfica, solamente el usuario tiene que elegir la capacidad computacional del servidor y la localización.



Figure 12: Logo de Azure Sql Server

**Azure Sql databases**

Es el servicio de azure que permite a los usarios gestionar sus bases de datos sql dentro de un servidor sql de azure



Figure 13: Logo de Azure Sql Database

**Log Analytics Workspace**

Es un servicio de Azure, que se utiliza para guardar los logs de los diferentes recursos de Azure en diferentes tablas, y también permite consultar los logs usando un lenguaje que se llama Kusto, que es un lenguaje muy intuitiva, que parce bastante al lenguaje de Sql.

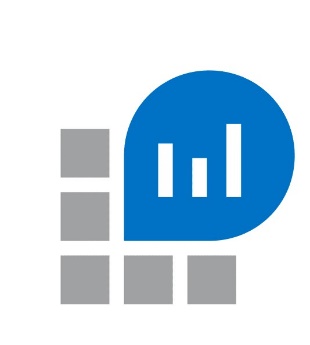


Figure 14: El logo del Log Analytics Workspace

**Application Insights**

Es un servicio de logs que se puede conecta con las aplicaciones, y con el recurso de Function App permite ver los logs en directo sin la necesidad de entrar al Log Analytics



Figure 15:El log de Application Insights

**Azure Container Registery**

Es un recurso de Azure que básicamente se usa para guardar imagines de contenedores, que se puede guardar más de una imagen y cada imagen se puede guardar con más de una versión, todos dentro del mismo recuso



Figure 16: Logo de Azure Container Registery

**Helm**

Es una hermanita que nos ayuda a gestionar nuestros paquetes de Kubernetes, Helm se usa para administrar y también guardar y compartir las aplicaciones dentro de Kubernetes, de una manera que nos ayuda a reutilizar estas aplicaciones varias veces.

Helm también tiene su propio lenguaje para crear templetes de las aplicaciones de Kubernetes, de tal manera que nos ayuda a desplegar las mismas aplicaciones en diferentes entornos sin tener errores.

## 1.4. El problema

En este trabajo se va a desarrollar un conjunto de codigos y también una guía para cómo usar los códigos y customizarlos, para ayudar a los usuarios que están interesados en probar o poner sus aplicaciones en modo de producción con integración continua, lo pueden hacer sin tener que preocuparse mucho por el proceso del despliegue y así concentran aún más en el proceso del desenrollo.

## 2. Justificación

El problema del despliegue del código hoy en día presenta uno de los problemas que están saliendo recientemente, y por esto ahora mismo existe el rol de devops, que antiguamente no existía, la responsabilidad de este rol principalmente consiste en como automatizar el procese del despliegue del código.

A continuación, vamos a estudiar si es necesario el desarrollo de este proyecte de tal manera como se plantea o no, vamos a estudiar las alternativas y también la alternativa de usar otro servicio de nube.

### Template deployment

Es una solución de Azure que permite desplegar aplicaciones webs en Azure Service App de forma fácil, usando ejemplos predefinidos, pero no permite ni usar Function Apps, AKS, ni base de datos y en esto se diferencia este proyecto

### La forma manual

Siempre se puede hacer todo el proceso manualmente poniendo todas las configuraciones a medida, pero el problema de esto es que antes de empezar a hacerlo, tienes que hacer una formación previa y normalmente el proceso del desarrollo de la solución tarda mucho tiempo.

### Usar otro servicio de nube

En esta solución hemos decidido de tirar por Azure, culpa del ecosistema de Microsoft, que se considera un ecosistema muy completa, porque también incluye el Microsoft 365, que incluye soluciones por el tema de gestión de documentos como el Word y SharePoint y también soluciones por el tema de comunicación entre empleados como Teams y Yammer y mucho más, y si en un futuro la empresa se decide hacerse más grande se le ira super bien usar el Microsoft 365.

### Estudios teóricos

[https://medium.com/globant/how-to-create-and-deploy-a-python-azure-function-usingazure-devops-ci-cd-2aa8f8675716](https://medium.com/globant/how-to-create-and-deploy-a-python-azure-function-using-azure-devops-ci-cd-2aa8f8675716) [https://docs.microsoft.com/en-us/azure/devops/pipelines/ecosystems/pythonwebapp?view=azure-devops](https://docs.microsoft.com/en-us/azure/devops/pipelines/ecosystems/python-webapp?view=azure-devops)

<https://microsoft.github.io/PartsUnlimited/iac/200.2x-IaC-CDAzureSQLdbwithVSTSandVS.html>

[https://docs.microsoft.com/en-us/azure/devops/pipelines/ecosystems/kubernetes/akstemplate?view=azure-devops&pivots=pipelines-yaml](https://docs.microsoft.com/en-us/azure/devops/pipelines/ecosystems/kubernetes/aks-template?view=azure-devops&pivots=pipelines-yaml)

Estos tutoriales están super bien explicado y son muy funcionales para poder llegar al mismo resultado que puedes llegar usando este proyecto, pero la diferencia es que usando mi proyecto puedes encontrar todo el material que necesitas en uno solo lugar, la guía este mucho más optimizada para tus aplicaciones de Python.

## 3. Alcance

En este apartado vamos a hablar de los objetivos y subobjetivos y también de los

Requerimientos funcionales y no funcionales

### 3.1. Objetivos

El objetivo principal de este trabajo es el desarrollo de una seria de códigos como a infraestructura y una guía para que el usuario pueda usar estos códigos, para poder crear la infraestructura necesaria para el despliegue de sus aplicaciones y después definir los pipelines para automatizar el proceso de la integración de los nuevos cambios de códigos, de tal manera que lo único que tiene que hacer el usuario es subir los cambios al Azure Repo y el proceso de poner los códigos en funcionamiento en la nube se hace de forma automática.

Aquí están explicados los subobjetivos

* El usuario puede crear un clúster de K8s, para que después publica sus microservicios en este mismo clúster.
* Crear un recurso de Function App y después usarlo para ejecutar los códigos allá.
* Crear un servicio de Web App y publicar las aplicaciones web allá.
* Crear un servicio de Azure Sql Databases, para poder usarlo para gestionar bases de datos allá.
* Configurar Azure Repos y Azure Pipelines para poder hacer los despliegues de forma automática.

### 3.2. Requerimientos

Ahora vamos a hablar sobre los requerimientos funcionales y n o funcionales

#### Requerimientos funcionales

Aquí vamos a describir los requerimientos funcionales para este proyecto

* Entender cómo funciona los diferentes recursos usadas para el despliegue en este proyecto (AKS, Function App, Azure Sql Database y Web App).
* Poder proporcionar un código como infraestructura que se puede usar para desplegar estos recursos y se puede reusar más de una vez.
* Entender como funcionen los Azure pipelines, para poder usarlos para desplegar los recursos y los códigos dentro de los recursos de forma automática.

#### Requerimientos no funcionales

Aquí vamos a describir los requerimientos no funcionales para este proyecto

* Tener una suscripción de Azure para poder hacer pruebas
* Saber cómo crear una guía fácil de leer y entender
* Preocuparse de que el código como a infraestructura se puede reusar más de una vez por diferentes usaurios.

### 3.3. Riesgos

Cualquier proyecto lleva riesgos, pero es muy importante poder clasificar los riesgos desde el principio, ya que así te ayuda de ir con más cuidado y plantear soluciones para ellos.

* Gastarse mucho dinero con Azure haciendo pruebas, ya que Azure no es gratis.
* Después de crear la infraestructura como a código, la configuración de los recursos se cambia y estos códigos ya dejen de funcionar.
* Borrar algún parte del código fuente de la aplicación antes de tenerlo en Azure repos.

## 4. Metodología

El desarrollo de este trabajo está ligado con la investigación de Azure, y no es un proyecto muy grande donde trabajan muchas personas (solo estoy yo), por esto es muy importante usar una metodología que no sea muy complicada y que se usa para proyectos pequeños y el equipo está formado por solo por una persona.

por esto hemos decidido de ir por la metodología scrum y modificarla para que sea solo

por una persona, que nos ayudara en planificar y organizar las diferentes tareas.

Cada 2 semanas (cada sprint) se hará reunión de máximo una hora conmigo mismo, en este reunión se decidirá las tareas que se tiene que hacer en los próximos 14 días y también se evaluar las tareas que se han hecho en los últimos 14 días

Cada día se hará una reunión conmigo mismo en esta reunión se decidirá la faena que se tiene que hacer en este mismo día

Para la organización de la metodología scrum voy a usar Azure bordas, que es un servicio gratuito de Azure para organizar faena usando la metodología scrum.

## 5. Planificación Temporal

## 

En este apartado vamos a explicar la planificación inicial de las tareas y también la planificación actual debido a la desviación del plan inicial

**5.1. Planificación Inicial**

*En este aparatado* vamos a explicar la planificación del plan inicial

## GP - Gestión del proyecto

Este apartado consiste en hacer toda la faena que va relacionada con la planificación y la gestión de cómo hacer el trabajo, esta fase se considera como la fase inicial.

### • *Objetivos (GP1)*

Antes de empezar con cualquier proyecto lo más importante es definir los objetivos del trabajo.

Esto se divide a 2 tipos de objetivos, los objetivos generales que se defienden al principio del proyecto, y los subobjetivos que se definen en el principio de cada sprint por lo que se quiere conseguir en este sprint, se estima que esta tarea llevara a capo unas 18 horas en total.

### • *Planificación (GP2)*

*En esta tarea vamos a definir como hacer el trabajo y se planificará como gestionar los previstos que, consisten en hacer una sesión diaria (yo solo), que en esta sesión voy a escoger las tareas que se tiene que hacer en este mismo día, y también dividir las tareas en subtareas si es necesario, se estima que esta tarea conllevará unas 50 horas.*

### • *Presupuesto (GP3)*

*Aquí vamos a detallar el presupuesto que va a acostar para hacer el proyecto, que incluye el material necesario para poder desarrollar el proyecto, tanto los personales como los gastos de Azure, se calcula que son unas 20 horas.*

### • *Sostenibilidad (GP4)*

En este apartado vamos a crear un informe para estudiar los efectos del proyecto sobre la sociedad y medioambiente, se estima que unas 17 hora para completar la tarea.

### • Documentación (GP5)

Es muy importante poder documentar la faena realizada para poder volver a modificar lo que ya está hecho o bien para valorar si otra persona tiene que ponerse con el código, también se incluye el tiempo invertido en desarrollar la guía de cómo usar el proyecto, se estima que se invertirán unas 75 horas.

o (GP5.1) Hacer la documentación del trabajo conllevará unas 40 horas o (GP5.2) Hacer la guía que saldrá del app de cómo usar los códigos para hacer el despliegue serán unas 35 horas

### • Presentación (GP6)

Una vez el trabajo ya finalizado, es muy importante poder preparar la presentación, esto se estima que son unas 10 horas.

## TP – Trabajo Previo

Aquí vamos a hablar sobre las tareas necesarias hacer, antes de empezar con el desarrollo.

### • Estudiar el entorno de Azure (TP1)

### El trabajo tratará de estudiar el entorno de Azure y como creamos los recursos y desplegamos las aplicaciones dentro de Azure, estudiando e indagando más sobre Azure y Azure pipeline, se calcula que son unas 50 horas.

### • Preparar el entorno del trabajo (TP2)

Esta tarea consiste en preparar la suscripción de Azure para poder trabajar con Azure, Azure no es un entorno gratis, sino que es de pago, pero, existen otras maneras para poder utilizarlo gratuitamente (como usar una suscripción de estudiantes), se calcula que serán unas 10 horas.

## DP – Desarrollo del Proyecto (DP)

En esta fase vamos a explicar las tareas que llevaran a cabo el proceso del desenrollo del proyecto.

### • Preparar la infraestructura de AKS(DP1)

Aquí vamos a preparar todo el material que necesitará el usuario para poder desplegar su aplicación de Python en AKS, calculamos que son unas 125 horas.

* (DP1.1) Crear el clúster dentro del AKS para desplazar los workloads, se considera que son 30 horas.
* (DP1.2) Preparar los workload donde se va a correr las aplicaciones en el clúster, se estima que son 30 horas.
* (DP1.3) Preparar los endpoints donde se va a llamar para acceder a las aplicaciones, serán unas 40 horas.
* DP1.4) Preparar toda la infraestructura como a código para qué después, los pequeños y medianos proyectos la puedan usar para su despliegue de aplicaciones, se considera que son 25 horas.

### • Preparar la infraestructura de Azure Service App (DP2)

Aquí vamos a hacer toda la faena necesaria para que el usuario pueda añadir sus web apps en las Service App de Azure de la manera más sencilla, se estima que serán unas 55 horas.

* (DP2.1) Crear la infraestructura de los Azure Service App serán unas 30 horas.
* (DP2.2) Importar la infraestructura como a código para poder usarlo después en los proyectos, se calcula que son 25 horas.

### • Preparar la infraestructura de Function App (DP3)

Esta es la tarea que vamos a trabajar para que el usuario pueda poner su código en Function App, para que el código se quede corriendo allí, se ha planificado que serán unas 50 horas.

* (DP3.1) Crear los Function App conllevará unas 35 horas.
* (DP3.2) Exportar la infraestructura como a código necesario, serán unas 15 horas.

### • Crear la aplicación de Python para generar los códigos (DP4)

Este parte es muy importante ya que es la interfaz que usara el usuario será unas 20 horas trabajando

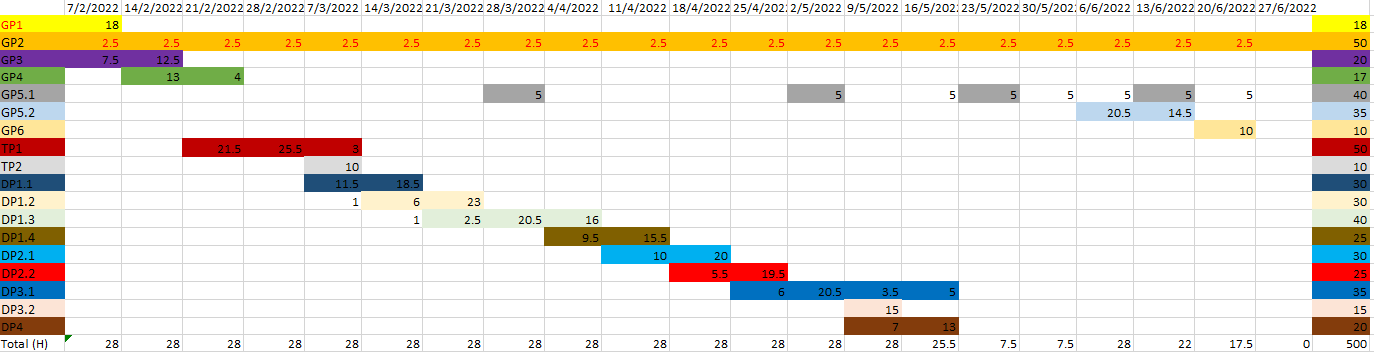


Figure 17: Taula de a planificación inicial del proyecto

* 1. **Planificación Actual**

La desviación respeto el plan inicial ha pasado debido a 4 razones

* En el plan inicial no estaba planificado la opción de incluir una base de datos, la cual tuvimos que incluir 2 tareas nuevas
  + 1. (DP5.1) Crear la base de datos que serán unas 10 horas.
    2. (DP5.2) Exportar la infraestructura como código necesario, serán unas 10 horas.
* Tuvimos un problema al intentar desplegar el recuso de Function App y por esto tuvimos que cambiar la tecnología de código como infraestructura de arm a bicep , y traducir todo el código de arm a bicep que también nos llevó a crear una tarea nueva (DP6) de 20 horas
* Al final se ha decidido de crear una prueba con código para demostrar como el usuario pueda adaptar su código, la cual se ha llevado a crear 4 tareas más
  + 1. (DP1.5) Crear el contenido para aks 20 horas.
    2. (DP2.3) Escribir el código para la web app 10 horas.
    3. (DP3.3) Escribir el código para la Function App 10 horas.
    4. (DP5.3) Crear la base de datos para el azure sql data base 10 horas.
* El proyecto empezó en 3-1 no en 2-1

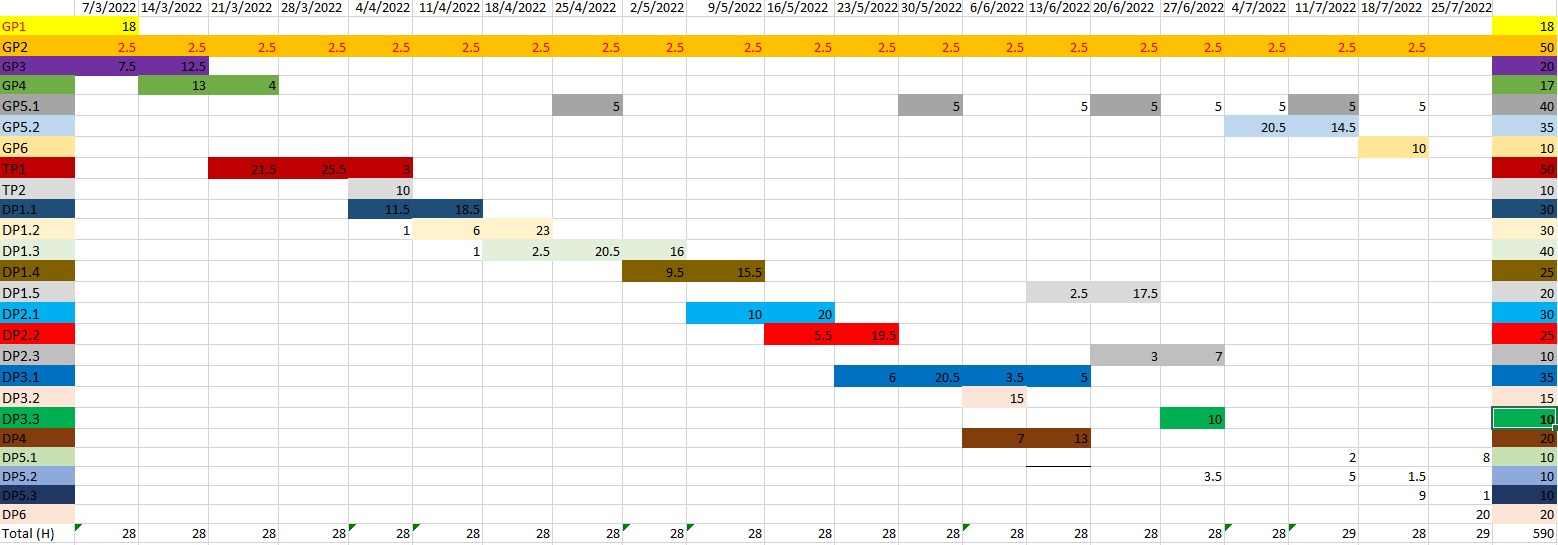


Figure 18: La planificacion real del proyecto

## 5.2. *Recursos*

### *Recursos humanos*

*Aquí vamos a estudiar los roles importantes para seguir desarrollando el proyecto*

* ***Jefe del proyecto (J)***

*Es la persona que toma les decisiones que van a nivel de negocio.*

* ***Director de tecnologías(T)***

*Es el responsable de decidir las tecnologías que se van a usar para el desarrollo del proyecto.*

* ***Desenrrollador (D)***

*Su tarea será seguir las decisiones del jefe del proyecto y el director de tecnologías para continuar con el proceso del desarrollo.*

## *5.3. Planificación de Riesgos*

Es muy importante poder tener una buena planificación de riesgos para dar paso a los improvistos que pueden pasar mientras se está desarrollando el proyecto. Ahora vamos a planificar los riesgos que pueden suceder y las medidas que podemos tomar.

### • Perder el portátil

Yo dispongo de dos portátiles (uno para el trabajo y otro personal) que son muy similares con las características, entonces si el principal que uso (el portátil del trabajo) le pasa algo puedo pasar al secundario, y todo mi código estará en Azure Repos, para poder recuperar el código en cualquier momento.

El tiempo del retraso para cambiar de un portátil a otro, será cuatro horas ya para poder configurar el otro portátil y establecer mi entrono de trabajo allá en el nuevo portátil.

### • El disco duro falla

Si el disco duro falla el final del día antes de subir el material (documentación + código) al Azure Repos, entonces en este caso la faena será para volver a escribir el trabajo perdido y también reemplazar el antiguo disco con uno nuevo.

El tiempo del retraso en este caso será 2 horas para volver a escribir el trabajo perdido y 3 horas para reemplazar el disco antiguo y también para comprar el nuevo disco y instalarlo.

### • Se cambia algo de los códigos de infraestructura de Azure o algo de las librerías que uso

Lo único que puedo hacer será revisar mi código y volver a adaptarlo a los nuevos cambios.

El tiempo del retraso en este caso será depende de lo que ha cambiado, pero será de 1 hora a 4 horas.

### • Mi suscripción gratuita de estudiante de Azure se termina

Cambiar mi suscripción gratuita a una normal de pago.

El tiempo del retraso en este caso será una hora como máximo para introducir los datos de la tarjeta del pago y también el tiempo que tarda Azure hasta que se renovar la suscripción.

### 6. Sostenibilidad

En este apartado vamos a hablar sobre la sostenibilidad del proyecto de fin de grado, primero vamos a hacer una autoevaluación y después discutir el proyecto en los ámbitos sociales y ambiental.

#### • Autoevaluación

En la carrera de informática llevo 5 años, la verdad durante estos 5 años en la carrera he encontrado muchas maneras para poder ver el lado de la sostenibilidad en la informática.

hay una asignatura optativa que habla sobre el lado social de la

informática también en algunas asignaturas hay algunos proyectos que se hacen al final del curso, que se hacen para subir la nota transversal, pero la verdad yo he echado de menos algo que sea obligatoria para que los alumnos tengan más conciencia por el lado de la sostenibilidad de la informática.

Yo me considero una persona bastante consiente del medioambiente y también de la influencia de la informática sobre nuestra sociedad y pienso que es algo muy importante porque nos puede ayudar mejorarnos como sociedad.

Lo que yo creo que me faltaría, es formaciones de cómo hacer propuestos económicas de proyecto informáticos, este tema nunca he tocado.

#### • Dimensión Ambiental

Para mí el medio ambiente es una de las cosas más importantes en un proyecto, el medio ambiente es la tierra donde vivimos es todo lo que tenemos que cuidar lo máximo posible.

Microsoft es una de las empresas que más que se importen por el medio ambiente, Microsoft han abierto el año pasado un Datacenter en Suecia que la energía que usa es 100% energía renovable.

Microsoft por 2025 tiene un plan que toda la energía que consume la empresas sea energías de recursos renovables.

Esto implica que la energía que se consume desde los Azure Datacenters, sea energía renovable, lo que implicar reducir los polución sobre el medio ambiente ,cuando la gente usan la nube de Azure.



Figure 19: El Datacenter de azure en Suecia

#### • Dimensión social

Este trabajo no tiene muchos afectos sociales ya que no es una red social ni tampoco un video juego, sino es una hermanita que ayudar a desplegar aplicaciones en Azure.

Pero así de efectos Social lo k podemos considerar, es animarse a más gente a desplegar sus aplicaciones en la nube y dejar de tener sus aplicaciones en Onpremises, lo que permite a la gente poder gestionarse mejor tema de espacios y también tener menos preocupación por sus proyectos, ya que la parte física no la gestionen ellos.

### 7. Gestión económica

Ahora vamos a hablar sobre la gestión económica del proyecto final de carrera y vamos a dividir la gestión económica en dos partes: presupuestos y control de gestión.

#### 7.1. Presupuestos

Aquí en presupuestos vamos a discutir los presupuestos previstos para pagar

##### Coste personal

Como ya hemos discutido anteriormente los tres roles que necesitamos para el desarrollo del proyecto son jefe del proyecto, director de tecnologías y desenrollador.

En la siguiente tabla vamos a detallar el coste de cada rol por hora y también mensualmente vamos a calcular lo que nos va a costar.

|  |  |  |
| --- | --- | --- |
| **Rol** | **Coste por hora** | **Coste mensual** |
| *Jefe del proyecto* | 35 euro/Hora | 5600 euro |
| *Director de tecnologías* | 30 euro/Hora | 4800 euro |
| desenrollador | 20 euro/Hora | 3200 euro |

Figure 20: Tabla de salarios

##### Costes generales

Ahora vamos a hablar sobre lo que nos va a costar los recursos necesarios por el proyecto que básicamente son dos recursos (el Pc y la suscripción de Azure).

|  |  |
| --- | --- |
| Recurso | Coste |
| Portátil | 800 euro |
| Suscripción de Azure | 100 euro |

Figure 21: Tabla de los costos de los recursos

11

Aquí vamos a hacer la gestión del presupuesto, a nivel de tareas, también vamos a calcular los costes incluyendo el coste de la seguridad social (lo vamos a calcular multiplicando el coste por 1,3).

Así que lo que vamos a contar aquí, va a ser el coste personal por cada tarea por separado.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tarea | Tiempo | Dependencia | Roles | Coste (euro) | CosteSS(euro) |
| GP  GP1  GP2  GP3  GP4  GP5  GP5.1  GP5.2  GP6 | 190  18  50  20  17  75  40  35  10 | -  -  GP1  GP2  -  DP  DP  DP  DP | J  J  J  J  J  J  J  J  J | 6,825  630  1,750  700  595  2,625  1,400  1,225  525 | 8,872.5  819  2,275  910  773.5  3,412.5  1,820  1,592.5  682.5 |
| TP  TP1  TP2 | 60  50  10 | GP1  GP1  GP1 | T  T  T | 1,800  1,500  300 | 2,340  1,950  390 |
| DP  DP1  DP1.1  DP 1.2  DP 1.3  DP 1.4  DP2  DP2.1  DP 2.2  DP3  DP3.1  DP3.2  DP4 | 250  125  30  30  40  25  55  30  25  50  35  15  20 | TP  TP  -  DP1.1  DP1.2  DP1.3  TP  -  DP2.1  TP  -  DP3.1  DP1 Y DP2 Y DP3 | D  D  D  D  D  D  D  D  D  D  D  D  D | 5,000  2,500  600  600  800  500  1100  600  500  1000  700  300  400 | 6500  3250  780  780  1,040  650  1,430  780  650  1300  910  390  520 |

Figure 22: Tabla de costes por tarea

##### Contingencia

Ahora vamos a calcula las Contingencia, las Contingencia la calculamos multiplicando el valor inicial por 15 porciento .

|  |  |  |
| --- | --- | --- |
| Tipo | Coste | Contingencia |
| Portátil | 800 euro | 120 euro |
| Azure suscripción | 100 euro | 15 euro |
| personal | 17,715.5 euro | 2,657.325 euro |
| total | 18,615.5 euro | 2,792.325 euro |

Figure 23: Tabla de Contingencia

### 7.2.Control de gestión

Una vez ya tenemos definido el presupuesto, vamos a definir los mecanismos para evitar desviaciones, por ejemplo, indicadores numéricos que nos van a ayudar con el control. En las reuniones del final de cada sprint, cada vez que se acabe una tarea, actualizamos el presupuesto con las horas reales y lo vamos a comprar con las horas estimadas desde el inicio.

Para poder controlar los imprevistos, al finalizar de cada tarea vamos a comparar los gastos extra que hemos gastado con los imprevistos y contingencia. De esta manera, podemos detectar las desviaciones y predecir si se necesita recortar alguna tarea o subir el presupuesto, ahora vamos a explicar los indicadores que vamos a usar para predecir las desviaciones.

* Desviación coste personal por tarea: (coste estimado − coste real)  horas reales
* Desviación realización tareas: (horas estimadas − horas reales) \* coste real
* Desviación coste recursos por tarea: (coste estimado − coste real)  horas reales
* Desviación total en la realización de tareas: coste estimado total − coste real total
* Desviación de coste de imprevistos: coste estimado imprevistos − coste real

#### 8. Bibliografía

1. (4 Companies Control 67% of the World's Cloud Infrastructure, 2022)PCMAG. 2022. *4 Companies Control 67% of the World's Cloud Infrastructure*. [online] Available at: <https://www.pcmag.com/news/four-companies-control-67-of-the-worlds-cloudinfrastructure> [Accessed 28 February 2022].
2. *Supporting our customers on the path to net zero: The microsoft cloud and decarbonization*. The Official Microsoft Blog. (2022, February 11). Retrieved March 21, 2022, from https://blogs.microsoft.com/blog/2021/10/27/supportingour-customers-on-the-path-to-net-zero-the-microsoft-cloud-and-decarbonization/ 3. 14, W. W. | D. (2022, February 9). *The Tech behind Microsoft's Sustainable Data Center Plan in Sweden*. Data Center Knowledge | News and analysis for the data center industry. Retrieved March 21, 2022, from

https://www.datacenterknowledge.com/energy/tech-behind-microsoft-ssustainable-data-center-plan-sweden

1. 14, W. W. | D. (2022, February 9). *The Tech behind Microsoft's Sustainable Data Center Plan in Sweden*. Data Center Knowledge | News and analysis for the data center industry. Retrieved March 21, 2022, from

https://www.datacenterknowledge.com/energy/tech-behind-microsoft-ssustainable-data-center-plan-sweden

1. *Microsoft sources 24/7 renewable energy for data centers*. ESG Today. (2021, November 4). Retrieved March 21, 2022, from

https://www.esgtoday.com/microsoft-sources-24-7-renewable-energy-for-datacenters/#:~:text=Unveiled%20in%20July%202021%2C%20the,and%20datacenter s%20globally%20by%202025.

1. Pimpalnerkar, A. (2021, December 22). *How to create and deploy a python Function App using Azure devops CI/CD*. Medium. Retrieved March 21, 2022, from https://medium.com/globant/how-to-create-and-deploy-a-python-azure-functionusing-azure-devops-ci-cd-2aa8f8675716
2. Juliakm. (n.d.). *Build and deploy Python Web Apps - Azure Pipelines*. Build and deploy Python web apps - Azure Pipelines | Microsoft Docs. Retrieved March 21,

2022, from https://docs.microsoft.com/en-

us/azure/devops/pipelines/ecosystems/python-webapp?view=azure-devops

1. Azooinmyluggage. (n.d.). *Deploy to azure kubernetes service with azure pipelines*

*- azure pipelines*. Deploy to Azure Kubernetes Service with Azure Pipelines -

Azure Pipelines | Microsoft Docs. Retrieved March 21, 2022, from https://docs.microsoft.com/en-

us/azure/devops/pipelines/ecosystems/kubernetes/aks-template?view=azuredevops&pivots=pipelines-yaml