API MANAGEMENT

TG3-IMPLEMENTACIÓN DE LAS TECNOLOGÍAS SELECCIONADAS

DESARROLLO CON TECNOLOGÍAS EMERGENTES

# 

Contenido

[1. Autores del trabajo, planificación y entrega 2](#_Toc448254544)

[1.1 Autores 2](#_Toc448254545)

[1.2 Planificación 2](#_Toc448254546)

[1.3 Entrega 2](#_Toc448254547)

[2. Requisitos del prototipo a implementar 3](#_Toc448254548)

[2.1 Requisitos funcionales 3](#_Toc448254549)

[2.2 Otros requisitos 3](#_Toc448254550)

[3. Criterios de comparación en la implementación 4](#_Toc448254551)

[3.1 Criterio 1: Nombre del criterio 4](#_Toc448254552)

[3.2 Criterio 2: Nombre del criterio 4](#_Toc448254553)

[3.N Criterio N: Nombre del criterio 4](#_Toc448254554)

[4. Proyecto de implementación de un prototipo del sistema utilizando la tecnología A 5](#_Toc448254555)

[4.1 Documentación de diseño 5](#_Toc448254556)

[4.2 Documentación de construcción 5](#_Toc448254557)

[4.3 Documentación de pruebas 5](#_Toc448254558)

[4.4 Documentación de instalación 5](#_Toc448254559)

[4.5 Manual de usuario 5](#_Toc448254560)

[5. Proyecto de implementación de un prototipo del sistema utilizando la tecnología B 6](#_Toc448254561)

[5.1 Documentación de diseño 6](#_Toc448254562)

[5.2 Documentación de construcción 6](#_Toc448254563)

[5.3 Documentación de pruebas 6](#_Toc448254564)

[5.4 Documentación de instalación 6](#_Toc448254565)

[5.5 Manual de usuario 6](#_Toc448254566)

[6. Comparación de las dos implementaciones 7](#_Toc448254567)

[6.1 Evaluación de los criterios en la implementación usando la tecnología A 7](#_Toc448254568)

[6.2 Evaluación de los criterios en la implementación usando la tecnología B 7](#_Toc448254569)

[7. Comparación de la implementación de las tecnologías 8](#_Toc448254570)

[8. Conclusiones 10](#_Toc448254571)

# 1. Autores del trabajo, planificación y entrega

## 1.1 Autores

Grupo T7:

* Laura Guillén Casal 03149840J (Coordinador)
* Francisco Miguel Sáez Bravo 50644226C
* Jesús Gail Bohórquez 09072386J
* Jorge Pérez Campos 09053996T
* Sergio Llanos García 09054884Z

## 1.2 Planificación

En este apartado se debe incluir copias de pantalla de la planificación del trabajo con diagramas Gantt: o bien un enlace (URL) a la web donde esté disponible la planificación si se ha utilizado una herramienta online de diagramación Gantt (por ejemplo, [Teamweek](https://teamweek.com/free-online-gantt-chart.html), [GanttPro](https://ganttpro.com/), [tomsplanner](https://plan.tomsplanner.es/), [sinnaps](https://www.sinnaps.com/), u otra).

Hay que tener en cuenta que cada participante del grupo debe tener asignadas tareas que sumen al menos 45 horas. El peso de este trabajo en la calificación total de la asignatura es de un 30%, por tanto requiere de una dedicación de 45 horas del total de 150 horas de la asignatura.

## 1.3 Entrega

En este apartado debe incluirse un enlace (URL) a un repositorio en GitHub creado para el trabajo.

En dicho repositorio debe encontrarse, al menos los siguientes archivos en la rama máster:

* Informe del trabajo: con el nombre TG3\_final.docx
* Presentación del trabajo: TG3\_final.pptx
* Prototipos obtenidos implementando cada una de las tecnologías (deben incluir el código fuente y todos los archivos necesarios para la instalación y uso de cada prototipo):
  + PrototipoTecnologiaA\_final.zip (o .rar)
  + PrototipoTecnologiaB\_final.zip (o .rar).

Dichos archivos serán los que se tendrán en cuenta para la calificación del trabajo.

# 2. Requisitos del prototipo a implementar

## 2.1 Requisitos funcionales

A continuación, encontramos una tabla con los requisitos funcionales comunes para las dos tecnologías utilizadas. Cabe destacar que la selección de estos requisitos funcionales se encuentra claramente condicionada por las funcionalidades presentes en las versiones de prueba y por lo tanto accesibles para nosotros

| **REQ.** | **DESCRIPCIÓN** |
| --- | --- |
| RF01- Creación/Publicación de una API | Capacidad de crear una API y realizar su posterior implementación |
| RF02- Suscripción a un API | Posibilidad de suscribirse a una API, pudiendo invocarla posteriormente |
| RF03- Políticas de seguridad | Introducción de métodos de autenticación |
| RF04- Edición de la API | Capacidad de modificar y adaptar el código y diseño de la API que hayas creado anteriormente |
| RF05- Control de acceso | Métodos de gestión de acceso concurrente a las APIs |
| RF06- API Store | Portal donde se alojan las APIs y se presentan al usuario para su futura suscripción |
| RF07- Servidor Proxy | Creación de un servidor encargado de recibir las peticiones y procesarlas |
| RF08- Herramientas de Análisis | Conjunto de herramientas que ofrece la plataforma para analizar una API, su rendimiento, funcionamiento, etc. |

## 2.2 Otros requisitos

En la siguiente tabla se muestran los requisitos no funcionales que se tendrán en cuenta para el estudio de ambas tecnologías:

| **REQ.** | **DESCRIPCIÓN** |
| --- | --- |
| R01- Portal de desarrolladores | Apartado de comunicación entre los desarrolladores y miembros de la comunidad |
| R02- Soporte de la compañía | Existencia de un servicio permanente de ayuda a los usuarios por parte de la compañía |
| R03- Documentación | Información suministrada por la compañía para el uso de la plataforma |
| R04- Personalización | Capacidad de customizar tu API |
| R05- Testeabilidad | Posibilidad de probar el código y funcionalidad de la API |

# 3. Criterios de comparación en la implementación

Se trata de criterios del tipo” “horas empleadas en el desarrollo del sistema”, “velocidad de funcionamiento del sistema”, “recursos necesarios”, etc.

## 3.1 Criterio 1: Dificultad del proceso de creación y publicación

*Nombre del criterio: Dificultad del proceso de creación y publicación.*

*Descripción: Grado de complejidad a la hora de crear y publicar una API.*

*Tipo de valor: Numérico (Escala 1-5).*

## 3.2 Criterio 2: Horas empleadas en la construcción del prototipo

*Nombre del criterio: Horas empleadas en la construcción del prototipo.*

*Descripción: Números de horas utilizadas para llevar a cabo la construcción de los prototipos.*

*Tipo de valor: Numérico (Horas).*

## 3.3 Criterio 3: Recursos necesarios

*Nombre del criterio: Recursos necesarios.*

*Descripción: Número de recursos que deben utilizarse para la utilización de las tecnologías.*

*Tipo de valor: Texto libre.*

## 3.4 Criterio 4: Configuración del servidor Proxy

*Nombre del criterio: Configuración del servidor Proxy.*

*Descripción: Complejidad y rapidez en la puesta en funcionamiento del servidor proxy.*

*Tipo de valor: Numérico (Escala 1-5).*

## 3.5 Criterio 5: Grado de personalización

*Nombre del criterio: Grado de personalización.*

*Descripción: Capacidad para modificar a gusto del usuario la interfaz de la API creada por el mismo.*

*Tipo de valor: Numérico (Escala 1-5).*

## 3.6 Criterio 6: Calidad del soporte

*Nombre del criterio: Calidad del soporte.*

*Descripción: Facilidad de contacto con el soporte de la plataforma y eficiencia en la respuesta de estos.*

*Tipo de valor: Numérico (Escala 1-5).*

## 3.7 Criterio 7: Calidad de la documentación

*Nombre del criterio: Calidad de la documentación.*

*Descripción: Nivel de detalle y claridad de la información que brinda la plataforma.*

*Tipo de valor: Numérico (Escala 1-5).*

## 3.8 Criterio 8: Facilidad de uso

*Nombre del criterio: Facilidad de uso.*

*Descripción: Grado de facilidad de uso y acceso a las distintas funcionalidades de las plataformas.*

*Tipo de valor: Numérico (Escala 1-5).*

## 3.9 Criterio 9: Implementación de seguridad y control de concurrencia

*Nombre del criterio: Implementación de seguridad y control de concurrencia.*

*Descripción: Capacidad de implementar seguridad y control de acceso a la API creada.*

*Tipo de valor: Numérico (Escala 1-5).*

## 3.10 Criterio 10: Profundidad del análisis

*Nombre del criterio: Profundidad del análisis.*

*Descripción: Cantidad y variedad de parámetros que ofrece la plataforma para el análisis de las APIs.*

*Tipo de valor: Numérico (Escala 1-5).*

## 3.11 Criterio 11: Capacidad de prueba

*Nombre del criterio: Capacidad de prueba.*

*Descripción: Posibilidad de realizar un test de prueba del código de la API.*

*Tipo de valor: Booleano (Si/No)*

# 4. Proyecto de implementación de un prototipo del sistema utilizando la tecnología A

Se trata de incluir en este apartado la documentación del desarrollo del proyecto de implementación, utilizando la tecnología A, del sistema cuyos requisitos funcionales se enumeraron en el apartado 2.

## 4.1 Documentación de diseño

Hay que incluir la descripción del diseño del prototipo, incluyendo diagramas, y el diseño de la interfaz de usuario.

## 4.2 Documentación de construcción

Hay que incluir una descripción de la construcción del prototipo, incluyendo algún extracto de código fuente. No es necesario todo el código. Sólo algún extracto para ver cómo se ha comentado.

## 4.3 Documentación de pruebas

Casos de prueba establecidos y resultados de las pruebas y acciones de corrección. No es creíble que no hayan aparecido errores en los caso de prueba.

## 4.4 Documentación de instalación

Descripción suficiente para que una persona que no ha participado en el proyecto pueda instalar el prototipo.

## 4.5 Manual de usuario

Descripción suficiente para que una persona que no ha participado en el proyecto pueda utilizar toda la funcionalidad que ofrece el prototipo. Que debe coincidir con los requisitos funcionales incluidos en el apartado 2.

# 5. Proyecto de implementación de un prototipo del sistema utilizando la tecnología B

Se trata de incluir en este apartado la documentación del desarrollo del proyecto de implementación, utilizando la tecnología B, del sistema cuyos requisitos funcionales se enumeraron en el apartado 2.

## 5.1 Documentación de diseño

Hay que incluir la descripción del diseño del prototipo, incluyendo diagramas, y el diseño de la interfaz de usuario.

## 5.2 Documentación de construcción

Hay que incluir una descripción de la construcción del prototipo, incluyendo algún extracto de código fuente. No es necesario todo el código. Sólo algún extracto para ver cómo se ha comentado.

## 5.3 Documentación de pruebas

Casos de prueba establecidos y resultados de las pruebas y acciones de corrección. No es creíble que no hayan aparecido errores en los caso de prueba.

## 5.4 Documentación de instalación

Descripción suficiente para que una persona que no ha participado en el proyecto pueda instalar el prototipo.

## 5.5 Manual de usuario

Descripción suficiente para que una persona que no ha participado en el proyecto pueda utilizar toda la funcionalidad que ofrece el prototipo. Que debe coincidir con los requisitos funcionales incluidos en el apartado 2.

# 6. Comparación de las dos implementaciones

Se trata de dar valores a los criterios de comparación definidos en el apartado 3 sobre la implementación de cada uno de los prototipos.

## 6.1 Evaluación de los criterios en la implementación usando la tecnología Apigee

| **CRITERIO** | **EVALUACIÓN** |
| --- | --- |
| Dificultad del proceso de creación y publicación | La dificultad en el proceso de creación y publicación de nuestras APIs ha sido relativamente sencilla, exceptuando la creación del proxy, que dentro de este proceso es la parte que implica mayor impedimento.  Valoración:3/5 |
| Horas empleadas en la construcción del prototipo | La construcción de este prototipo nos llevó aproximadamente 5 horas, principalmente, como se ha explicado anteriormente, debido a la dificultad de la creación el proxy, una vez sobrepasado este paso se requiere menos tiempo.  Valoración: 5 horas |
| Recursos necesarios | Para la creación del prototipo basado en la tecnología APIGEE es necesario el uso de un servidor Proxy para ejecutar las peticiones del usuario y cURL, una herramienta y librería la cual se aplica sobre la consola del sistema, la cual nos permite transferir datos a través de URL’s  Valoración: servidor Proxy y cURL |
| Configuración del servidor Proxy | Una vez iniciemos sesión en la plataforma podemos crear y configurar nuestro propio proxy en el que podremos indicarle el nombre, la ruta, descripción, seguridad, etc. Y una vez creado podremos testearlo y hacer modificaciones en el futuro.  Valoración: 4/5 |
| Grado de personalización | La plataforma de Google nos permite personalizar nuestras APIs de tal forma que se realiza a través del código fuente de la misma API para que cumpla la funcionalidad que nosotros queramos.  Valoración: 5/5 |
| Calidad del soporte | Por parte de Apigee no hemos recibido respuesta de la empresa a los correos enviados, por lo que su nivel de soporte en este apartado de atención es bastante escaso, por lo menos desde nuestra experiencia. A pesar de ello, el nivel de documentación de la herramienta y su facilidad mejorar la puntuación de este apartado  Valoración: 2/5 |
| Calidad de la documentación | La calidad de la documentación prestada por APIGEE es muy buena debido a que podemos encontrar una gran cantidad de información sobre la mayoría de apartados de la herramienta y, además, dicha información es muy detallada y precisa, cuenta paso a paso los pasos a seguir para los diferentes procesos de la aplicación.  Valoración: 5/5 |
| Facilidad de uso | APIGEE es un entorno muy intuitivo y enfocado a la comodidad de uso por parte del usuario, además, debido a la gran cantidad de documentación y tutoriales podemos encontrar explicaciones de cada apartado de una forma visual, sabiendo qué se puede realizar dentro de cada uno.  Valoración: 5/5 |
| Implementación de seguridad y control de concurrencia | A través del uso de Keys, Apigee permite implementar un método de seguridad la cuál da acceso a diferentes métodos/apartados de las API’s según la key que tenga el usuario.  Además, se pueden controlar el número de usuarios que pueden acceder simultáneamente a las API’s.  Valoración: 4/5 |
| Profundidad del análisis | APIGEE permite hacer diferentes pruebas de nuestras APIs, según los parámetros que nosotros escojamos, como puede ser el número de visitas o él número de errores encontrados. El problema lo encontramos debido a que dichos análisis sólo permite realizarlos si contamos con la versión de pago de Apigee.  Valoración: 4/5 |
| Capacidad de prueba | Todas las APIs generadas en esta plataforma pueden ser testeadas para conocer su funcionalidad y claridad.  Valoración: Sí |

Y algunos comentarios aclaratorios sobre aquellos criterios cuyo valor indicado en la tabla no sea suficiente para entenderlo.

## 6.2 Evaluación de los criterios en la implementación usando la tecnología WSO2

| **CRITERIO** | **EVALUACIÓN** |
| --- | --- |
| Dificultad del proceso de creación y publicación | Para poder publicar y crear una API en WSO2 es conveniente leer la documentación sobre ello con anterioridad pues sin saber puede resultar una tarea complicada.  Valoración:4/5 |
| Horas empleadas en la construcción del prototipo | Dado que es necesario documentarse e informarse con anterioridad sobre el proceso de creación, incluiremos dicho tiempo en la creación del prototipo por lo que es una duración aproximada de 4:30.  Valoración: 4:30 horas |
| Recursos necesarios | Puesto que estamos utilizando la versión de prueba en la nube de WSO2, únicamente es necesario el acceso a un navegador web.  Valoración: Conexión a internet |
| Configuración del servidor Proxy | La nube de WSO2 cuenta con un proxy propio ya configurado.  Valoración: 1/5 |
| Grado de personalización | WSO2 cuenta con la función de personalización de una API a gusto del creador, modificación de imagen, llamadas por segundo, seguridad y acceso a esta.  Valoración: 5/5 |
| Calidad del soporte | Resulta sencillo ponerse en contacto con el área de soporte de WSO2, además brinda una rápida atención y respuesta.  Valoración: 4/5 |
| Calidad de la documentación | La documentación que ofrece WSO2 es completa, pero en algunos momentos es complicada de entender, además de encontrarse por escrito también existen videotutoriales.  Valoración: 3.5/5 |
| Facilidad de uso | El trabajo con la nube de WSO2 no es especialmente fácil, pues es complicada de entender y de encontrar ciertas funciones, aun que por otro lado su interfaz de usuario si es amigable.  Valoración:3.5/5 |
| Implementación de seguridad y control de concurrencia | WSO2 permite integrar cierta seguridad en tu API a través de la creación de roles y controlar el acceso simultaneo a la API por parte del creador.  Valoración: 4/5 |
| Profundidad del análisis | La plataforma de WSO2 cuenta con una amplia disposición de herramientas de análisis para una API trabajando a través de una serie de parámetros. Sus resultados son de gran ayuda  Valoración: 5 |
| Capacidad de prueba | WSO2 permite realizar una prueba del código de una API ejecutando los métodos de esta.  Valoración: Si |

* Configuración del servidor proxy (WSO2): se valora como 1 dado que no existe dificultad alguna pues ya viene integrado en la plataforma.
* Dificultad del proceso de creación (WSO2): Entendemos 5 como una muy alta dificultad y 1 como muy baja dificultad.
* En los apartados que se mida la dificultad la entenderemos como se ha comentado en el punto anterior.

# 7. Comparación de la implementación de las tecnologías

Debe incluir al menos una tabla resumen, en sección de página horizontal, cruzando los criterios y los valores de cada tecnología. Con una columna de comentarios sobre la comparación

| **CRITERIOS** | **TECNOLOGÍA A** | **TECNOLOGÍA B** | **COMENTARIOS** |
| --- | --- | --- | --- |
| 1 |  |  |  |
| 2 |  |  |  |
| … |  |  |  |
| N |  |  |  |

# 8. Conclusiones

A partir de la información incluida en el apartado 7 y de la experiencia al realizar el trabajo, el grupo debe estar en condiciones de manifestar su opinión sobre la implementación del sistema utilizando ambas tecnologías, y debe plasmarla en este apartado, indicando las ventajas e inconvenientes más relevantes de utilizar una u otra tecnología para implementar el sistema.

---------------------------

(Hay que cumplir la estructura básica indicada de secciones. Pero si se desea se pueden añadir otras secciones como anexos. Por ejemplo, alguna encuesta de opinión realizada sobre las tecnologías, etc.)