

# Análisis de la capacidad y determinación del plan de precios

El presente documento contiene la justificación y razonamiento aportado por el equipo 'FIS-G4' frente a las siguientes tareas:

- Análisis de la capacidad del Customer Agreement asumiendo que esto no tiene implicaciones en el plan de precios.
- Determinar el coste de cada plan del Customer Agreement de forma justificada en base al análisis de capacidad.

## Análisis de la capacidad

### *Estimación de usuarios*

En esta sección se plantea una evolución estimada del número de usuarios que tendrá nuestra aplicación, con idea de que sea utilizada como base para el cálculo de los costes asociados a las integraciones de herramientas de terceros para cada microservicio.

Teniendo en cuenta que, durante los primeros meses tras el lanzamiento de la aplicación, se realizan tareas clave para el desarrollo del producto, como:

- Una investigación del mercado continua
- Estrategias de marketing
- Colaboraciones con otras organizaciones
- Recogida y aplicación del feedback de los usuarios de manera continua

Es posible estimar una evolución del número de usuarios para nuestra aplicación basándonos en la trayectoria de plataformas similares durante los primeros 12 meses. En concreto:

1. Durante el **primer mes**, dependiendo de la estrategia de lanzamiento y marketing que hayamos seguido, podemos estimar entre **1000 y 5000** usuarios.
2. A partir de este punto, podemos considerar un **crecimiento mensual de un 20 o un 30 %** de usuarios hasta llegar a tener entre **5000 y 20.000 usuarios al final del tercer mes**.
3. Antes del sexto mes, manteniendo una estrategia continua y aplicando ajustes en la aplicación según el feedback de los usuarios, podríamos ver un crecimiento más rápido, llegando a **20.000 - 50.000 usuarios al final del sexto mes**.
4. Finalmente, si la plataforma sigue atrayendo nuevos usuarios y manteniendo los existentes, es posible alcanzar en torno a **50.000 - 100.000 usuarios, o más, al final del primer año**.

En base a esta estimación, se considerará el siguiente volumen de usuarios para cada mes:

Mes	Número de usuarios
1	3.000
2	5.000
3	8.000
4	12.000
5	18.000
6	25.000
7	35.000
8	50.000
9	70.000
10	90.000
11	110.000
12	130.000

## Descripción de XaaS utilizados

En esta sección presentaremos las herramientas utilizadas por los microservicios de FIS-G4. Algunas de ellas son utilizadas por varios microservicios, mientras que otras sólo en uno.

### *Google Cloud Storage*

Servicio de almacenamiento en la nube ofrecido por Google Cloud Platform. Su utilización implica la creación de *buckets* que actúan como contenedores para diversos tipos de archivos de gran tamaño, que contienen información no estructurada, e.g videos, fotos, etc. La API de servicio facilita su interacción con otras aplicaciones, permitiendo a los desarrolladores cargar, descargar y gestionar archivos de manera programática.

Dentro de nuestro proyecto, la hemos utilizado en el microservicio de usuarios y materiales para almacenar las imágenes de perfil, contenido multimedia de los cursos, etc.

### *Google Cloud Functions*

Google Cloud Functions es un servicio de cómputo sin servidor que permite a los desarrolladores ejecutar funciones individuales en respuesta a eventos específicos sin preocuparse por la gestión de la infraestructura subyacente. Las funciones se ejecutan en entornos aislados y escalan automáticamente según la demanda, lo que proporciona flexibilidad y eficiencia en el consumo de recursos.

Dentro de nuestro proyecto las hemos utilizado para realizar la validación y generación de tokens JWT de manera centralizada para todos los microservicios.

## *Brevo*

Brevo es una plataforma de marketing para empresas que, entre sus productos, ofrece una API para enviar correos electrónicos de manera programática, simplificando y mejorando la gestión del correo electrónico transaccional. Ofrece características como seguimiento de entregas, estadísticas detalladas, gestión de listas de suscriptores, plantillas personalizables y capacidades de prueba.

Dentro de nuestro proyecto la utilizamos durante el proceso de recuperación de contraseña de los usuarios.

## *Stripe*

Stripe es herramienta que ofrece una API para integrar capacidades de pago y servicios financieros en aplicaciones y sitios web. La herramienta implementa un sistema de límites de tasa adaptable, que varía según el patrón de uso, lo que permite incrementar el número de llamadas permitidas en momentos de baja demanda y disminuirlo en épocas de alta actividad.

En nuestro proyecto la consume el microservicio de gestión de pagos para ofrecer una plataforma unificada y de confianza.

## *Github*

GitHub es una plataforma de desarrollo colaborativo que permite el control de versiones, almacenamiento de código en repositorios, colaboración en equipos, gestión de proyectos y código, hacer wikis y documentación, integración continua y despliegue.

En nuestro proyecto la hemos utilizado, además de como sistema de control de versiones para todos los microservicios y el frontend, como herramienta colaborativa para gestionar a los equipos y sus tareas, automatizar workflows de integración continua y despliegue, etc.

## *Google Cloud Platform*

Google Cloud Platform (GCP) es una plataforma integral de servicios de computación en la nube ofrecida por Google. Esta plataforma proporciona una serie de servicios modulares que incluyen computación, almacenamiento de datos, análisis de datos y aprendizaje automático, junto con un conjunto de herramientas de gestión.

En FIS-G4 la utilizaremos para desplegar nuestra arquitectura backend de cada microservicio, exponiendo a internet un único punto de entrada común a ellos gestionado por un API Gateway.

## *Vercel*

Vercel es una plataforma de desarrollo web enfocada en la eficiencia y facilidad de uso para los desarrolladores frontend. Una de sus principales ventajas es su capacidad para escalar automáticamente según las necesidades del proyecto, lo que es esencial para manejar aumentos de tráfico y agregar nuevas características. Además, la plataforma garantiza tiempos de carga rápidos mediante su red de entrega de contenido global y técnicas de

optimización, mejorando así la experiencia del usuario final. También permite extraer insights de la página que ayudan a evaluar su rendimiento.

Posiblemente su mejor característica es que está diseñado para integrarse con un repositorio Git (GitHub, GitLab, etc), y desplegar una nueva versión del proyecto por cada commit a la rama principal.

En FIS-G4 la utilizaremos para desplegar de manera continua el frontend de la aplicación.

### *Microsoft Teams*

Microsoft Teams es una plataforma integral que reúne chat, videoconferencias, colaboración en documentos y aplicaciones integradas en un solo entorno, mejorando la eficiencia y la colaboración en equipos empresariales.

En nuestro proyecto, esta plataforma se ha utilizado para organizar reuniones y compartir los documentos del proyecto.

## Determinación del coste

### *Github*

Hemos escogido el plan “Team” que nos ofrece mayor seguridad, atención al cliente, protección de ramas y permiso para utilizar numerosas funcionalidades de GitHub en repositorios privados.

El plan tiene un coste 4\$/mes por usuario, que por los 9 desarrolladores que somos, hace un total de **36\$/mes**.

### *Microsoft Teams*

Respecto a Teams hemos preferido el plan “Business Basic”, que nos permite tener grabaciones y transcripciones además de todo el paquete de Microsoft 365. Este plan tiene un coste de 6\$/mes por usuario que, de nuevo, por 9 usuarios hace un total de **54\$/mes**.

### *Vercel*

Para el propósito de este proyecto, el plan gratuito de Vercel es suficiente para garantizar que el frontend de FIS-G4 está disponible en todo momento y se despliega de manera automática al hacer un push a main. En caso de trasladar toda nuestra arquitectura de despliegue a Vercel, sí barajaríamos la posibilidad de mejorar al plan Pro, pero por ahora no es necesario.

## Google Cloud Storage

Google Cloud no ofrece planes de precio específicos, si no pago por uso de sus servicios. Esto es muy común en los IaaS. En el caso del servicio de Cloud Storage Estándar, el coste por GB almacenado al mes tiene un coste asociado de 0.026\$, más 0,01\$ por cada 1000 peticiones de lectura o actualización de un recurso.

Para realizar una estimación del costo asociado al uso de Google Cloud Storage, necesitamos tener en cuenta la cantidad de datos almacenados por uno de los tres tipos de usuario. En concreto:

- Los usuarios con Plan Gratuito (Free Tier) pueden almacenar hasta 20 GB de materiales y cursos de forma gratuita.
- Los usuarios con Plan Avanzado tienen asignados 50 GB de almacenamiento para materiales y cursos.
- Los usuarios con Plan Pro pueden almacenar hasta 100 GB de materiales y cursos.

Además, consideraremos que el 75% de los usuarios de nuestra plataforma también consumen cursos y, por tanto, realizan una media de 15 peticiones de materiales por curso realizado por día. También realizarán una media de 5 peticiones al servicio para obtener su foto de perfil.

Con esta información, podemos estimar el coste más pesimista asociado a este recurso, que se consigue al considerar que todos los usuarios utilizan el máximo de almacenamiento disponible en su plan. Podemos modelar este coste de la siguiente manera:

$$CCS_{dia} = N_{basico} * 20 * 0.026 + N_{avanzado} * 50 * 0.026 + N_{pro} * 100 * 0.026 \\ + \frac{N_{usuarios} * 0.75 * 15}{1000} * 0.01 + N_{usuarios} * 5 * 0.01$$

Simplificando la expresión, obtenemos la siguiente forma de calcular aproximadamente (con un error de  $10^{-4}$ ) el precio asociado por día de uso de Cloud Storage:

$$CCS_{dia} = 0.26 * (2N_{basico} + 5N_{avanzado} + 10N_{pro}) + 0.05N_{usuarios}$$

Por tanto, el coste mensual:

$$CCS_{mes} = CCS_{dia} \cdot 30$$

## Google Cloud Functions

En Cloud Functions cobran en función de dos métricas: el **número de invocaciones** a las funciones, y el **tiempo de procesamiento** dependiendo de la memoria utilizada. En nuestro caso haremos uso de una memoria de **256 MB**.

Para realizar los cálculos, consideraremos que un usuario hace una media de **30 peticiones** a las APIs de nuestros microservicios por sesión, las cuáles hacen uso de los mecanismos de autenticación con JWT implementadas con este servicio. A continuación, se detalla la distribución de llamadas:

- 1 llamada de autenticación.
- 10 llamadas al microservicio de cursos en concepto de búsqueda.
- 10 llamadas al microservicio de materiales, bien para consumirlos o para subirlos.
- 2 llamadas al microservicio de pagos, una para hacer una transacción y otra para consultar su historial de pago.
- 7 llamadas al microservicio de reseñas, presente en los detalles de cada curso. Para este último se ha aplicado una reducción con respecto al número de llamadas del microservicio de cursos, presuponiendo que un usuario que busca cursos no siempre accede a los detalles de estos.

Con cada petición, pueden hacerse una o dos llamadas a funciones de autenticación con JWT. Este número depende de si el método solicitado necesita extraer información del token incluido en la cabecera de la petición o no.

Es por ello que **consideraremos que el 50% de las peticiones realizadas por un usuario sí necesitarán extraer información del token**, y el resto no.

Por último, según los resultados de nuestras pruebas de rendimiento, cada petición tarda una media de **50ms en ejecutarse**.

Con todo esto, podemos definir el coste medio de uso del servicio de Cloud Functions con la siguiente ecuación:

$$CCF_{dia} = 0,4 \cdot \frac{30 \cdot N_{usuarios}}{2.000.000} + 4,63 \cdot 10^{-6} \cdot (7,5 + 15) \cdot N_{usuarios}$$

Al tardar 50ms en ejecutarse, las 15 llamadas que requieren usar dos veces Cloud Functions de JWT se han contabilizado como 1, y las otras 15, al usar solo una llamada, hacen el equivalente de 7,5 para cubrir 100ms. Simplificando:

$$CCF_{dia} = \frac{3 \cdot N_{usuarios}}{500.000} + 1,04 \cdot 10^{-4} \cdot N_{usuarios}$$

Por tanto, el coste mensual:

$$CCF_{mes} = CCF_{dia} \cdot 30$$

## *Google Cloud Platform*

En base

### *SendGrid*

En base a nuestra estimación de usuarios, durante los tres primeros meses podemos hacer uso del **plan Free** de SendGrid, que incluye un máximo de 100 e-mails al día, es decir, 3000 al mes si respetamos la limitación. Para llegar a esta conclusión, hemos considerado que uno de cada tres usuarios olvida su contraseña y necesita hacer uso del servicio, lo que resulta en que podemos dar **soporte con este plan a 9000 usuarios**. Además, durante los primeros meses del lanzamiento es muy probable que el porcentaje de olvidos sea menor.

A partir del cuarto mes, pasaríamos al plan **Essentials**, que nos permite realizar hasta 50.000 envíos de e-mails al mes. Su precio es de **\$19,95** y, en base a nuestro a nuestra estimación del 33 % de usuarios que olvida la contraseña, nos permitiría dar **soporte hasta a 150.000 usuarios**, ofreciendo cobertura durante el primer año de servicio.

### *Stripe*

Stripe actualmente aplica un cargo del 1,5% más 0,25€ por transacción para tarjetas estándar del Espacio Económico Europeo. Por otro lado, las transacciones con tarjetas del Reino Unido incurren en un coste del 2,5% más 0,25€ por operación.

## [Resumen del análisis](#)

### ***Cálculo del OpEX***

### ***Justificación del precio***

### ***Conclusiones generales***