



Escuela Técnica Superior de
Ingeniería Informática

TRABAJO FIN DE GRADO

Título del trabajo

Realizado por
Javier García Aguilar

Para la obtención del título de
Grado en Ingeniería Informática - Ingeniería del Software

Dirigido por
José Antonio Parejo Maestre
Antonio Ruiz Cortés

En el departamento de
Lenguajes y Sistemas Informáticos

Convocatoria de junio, curso 2023/24

Aquí la dedicatoria del trabajo

Agradecimientos

Quiero agradecer a X por...

También quiero agradecer a Y por...

Resumen

Incluya aquí un resumen de los aspectos generales de su trabajo, en español.

Palabras clave: Palabra clave 1, palabra clave 2, ..., palabra clave N

Abstract

This section should contain an English version of the Spanish abstract.

Keywords: Keyword 1, keyword 2, ..., keyword N

Índice general

1	Introducción	1
1.1.	Presentación del problema	1
1.2.	Motivación	3
1.3.	Objetivos del proyecto	4
1.3.1.	Objetivos técnicos	4
1.3.2.	Objetivos docentes	4
1.4.	Estructura de la memoria	4
2	Estudio previo	5
2.1.	Metodología	5
2.2.	Política de código	5
2.2.1.	Ramificación	5
2.2.2.	Commits	5
2.2.3.	Revisiones	5
2.3.	Planificación	5
2.3.1.	Product Backlog del proyecto	5
2.3.2.	Sprint Backlog	5
3	Análisis tecnológico	6
3.1.	Lenguajes, librerías y herramientas	6
3.1.1.	Lenguajes	6
3.1.2.	Librerías	6
3.1.3.	Herramientas	6
4	Diseño de la solución	7
4.1.	Diagramas	7
5	Recreación	8
5.1.	Lista de APIs	8
5.2.	Modelado SLA4OAI	8
5.3.	Clúster HPC	8
5.4.	Librería	8
5.5.	Conclusión de resultados	8
6	Conclusiones	9
	Bibliografía	10

Índice de figuras

1.1. Este modelo visual representa una API web, sus componentes y su funcionamiento (Fuente: 1)	1
1.2. Estado del mercado de APIs hasta el 2010 (Fuente: 2)	2
1.3. Representa el crecimiento de directorios API en ProgrammableWeb (Fuente: 3)	2
1.4. Estadísticas de Postman (Fuente: 4)	3

Índice de tablas

Índice de extractos de código

1. Introducción

1.1. Presentación del problema

Las API (Application Programming Interface) es una pieza de código que permite a diferentes aplicaciones comunicarse entre sí y compartir información y funcionalidades. Actúa como un intermediario entre sistemas, permitiendo la comunicación y el intercambio de datos y acciones específicas[5]. En el contexto web, estas interfaces procesan solicitudes HTTP y suelen devolver una respuesta en formato JSON, aunque su implementación puede variar. En la figura 1.1, podemos ver un ejemplo sencillo de una reserva y como pasa por las diferentes capas hasta devolver una respuesta.

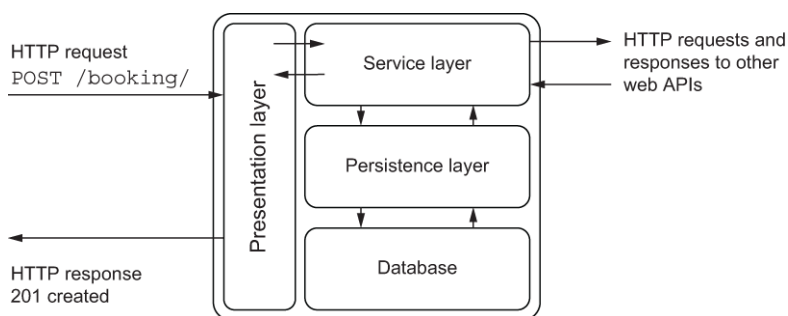


Figura 1.1: Este modelo visual representa una API web, sus componentes y su funcionamiento (Fuente: 1)

A continuación, exploraremos la evolución del mercado de las API desde el año 2000 hasta el 2019, dividiendo en tres intervalos temporales: 2000-2008, 2008-2010, y 2010-2019. Utilizaremos figuras (1.2, 1.3) visuales para respaldar nuestro análisis, detallando el crecimiento y la relevancia de las API en cada periodo.

Durante el primer intervalo, que abarca desde el año 2000 hasta el 2008, se observa un crecimiento gradual pero modesto en la adopción de API, con apenas mil interfaces registradas en el mercado.

El periodo entre los años 2008 y 2010 marca un cambio significativo, con un aumento notable en la cantidad de API, que alcanza las dos mil interfaces, demostrando un interés creciente y una mayor adopción en el ámbito web.

Finalmente, en el intervalo que abarca desde el 2010 hasta el 2019, presenciamos un aumento exponencial en el número de API registradas, llegando a la impresionante cifra de 22000 interfaces, lo que refleja su consolidación como una parte fundamental del ecosistema digital.

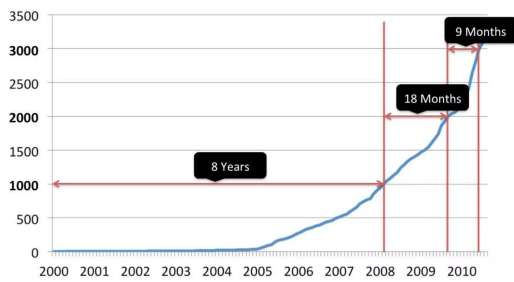


Figura 1.2: Estado del mercado de APIs hasta el 2010 (Fuente: 2)

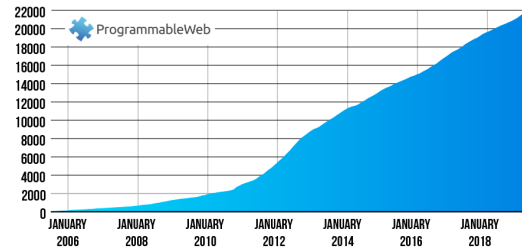


Figura 1.3: Representa el crecimiento de directorios API en ProgrammableWeb (Fuente: 3)

El notable crecimiento del mercado de las API, como se ha ilustrado en las figuras 1.2 y 1.3, refleja su creciente relevancia y adopción en el ámbito digital. Sin embargo, este aumento en la disponibilidad y variedad de API, también ha planteado nuevos desafíos, especialmente en lo que respecta a garantizar su funcionalidad y fiabilidad. A medida que las API se han vuelto más omnipresentes en el ecosistema digital, la necesidad de un riguroso proceso de testing se ha vuelto aún más crucial. En este sentido, el análisis del mercado de las API no solo nos proporciona una visión del panorama actual, sino que también nos señala la importancia de abordar la calidad y el rendimiento de estas interfaces. A continuación, exploraremos una herramienta de testing de API muy bien conocida, Postman¹.

Postman es una plataforma de construcción y uso de API[6], en la cual puedes crear colecciones donde se organizan solicitudes API agrupadas y donde se ejecutan fácilmente para probar API. Además, Postman tiene una sección de blog donde publica, entre otras cosas, estadísticas anuales sobre sus usuarios y colecciones. Las cuales vamos a ver en la figura 1.4.

La figura 1.4 muestra que hay más de 25 millones de usuarios, aproximadamente 121 millones de colecciones y aproximadamente 1.29 mil millones. Viendo estas cifras se puede apreciar fácilmente qué tan importante es el testing de API.

Ya Alberto Martín López en su artículo *REStest: automated black-box testing of RESTful web APIs*[7], comentó la importancia de probar a fondo una API RESTful en la integración de software. Un año después realizó un estudio empírico sobre el uso de métodos automatizados de generación de casos de prueba[8], en ese estudio probó 13 API durante 15 días y analizó los errores encontrados.

Tras esos estudios realizados por Alberto Martín, surgió la duda de si sería buena idea establecer un limitador a las llamadas de las API durante el testing, de forma que se evitara el bloqueo temporal o **cooldown**, hablando en códigos de estado HTTP suelen ser 429. De esta forma surgió la idea de PlanifyAPI², una librería en Python que establece un limitador a las llamadas basándose en el *rate limit* o las cuotas de los planes de precio de las API.

¹<https://www.postman.com/>

²<https://github.com/PlanifyAPI/US-TFG-PlanifyAPI-Python>

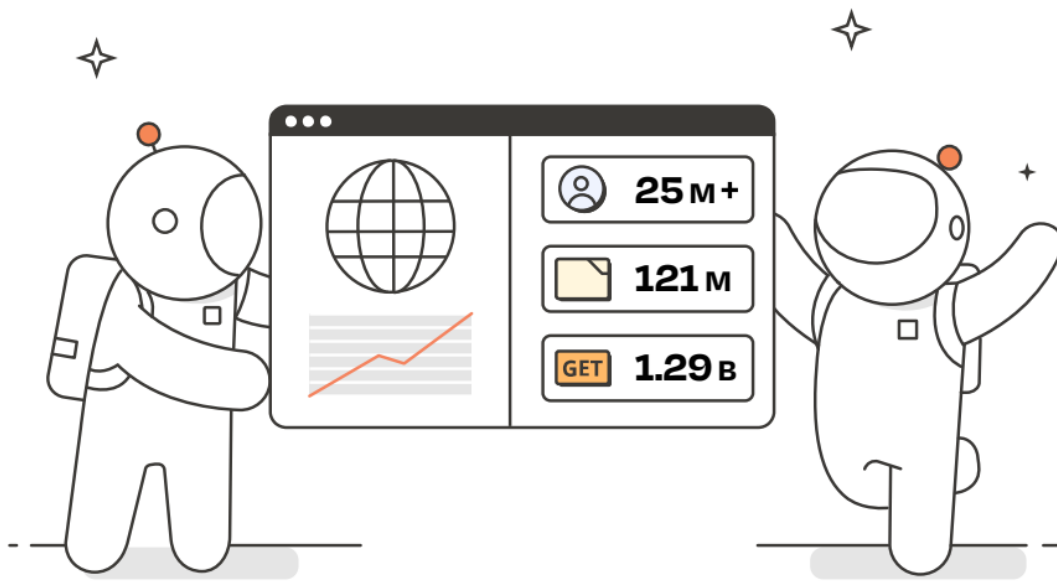


Figura 1.4: Estadísticas de Postman (Fuente: [4](#))

1.2. Motivación

La motivación principal surge del profundo interés del autor por el campo de la investigación, lo que le ha llevado a poner a prueba su pasión en este ámbito. Es así como surge la idea de este trabajo de fin de grado, el cual se inspira en el experimento realizado por Alberto Martín López[8].

Este trabajo de fin de grado se fundamenta en dos aspectos motivacionales: uno de carácter investigador y otro de índole académica. En lo que concierne a la investigación, se ha tratado de recrear el experimento de Alberto Martín López[8] y analizar detenidamente las implicaciones que surgirían al establecer un límite en las llamadas a las API durante las pruebas. Por otro lado, desde una perspectiva académica, el proyecto representa un desafío de considerable envergadura en comparación con las tareas habituales realizadas durante el grado de Ingeniería Informática - Ingeniería del Software. Además, supone la oportunidad de familiarizarse con herramientas utilizadas en entornos profesionales, como GitHub y Grafana, entre otras.

1.3. Objetivos del proyecto

1.3.1. Objetivos técnicos

1.3.2. Objetivos docentes

1.4. Estructura de la memoria

2. Estudio previo

2.1. Metodología

2.2. Política de código

2.2.1. Ramificación

2.2.2. Commits

2.2.3. Revisiones

2.3. Planificación

2.3.1. Product Backlog del proyecto

2.3.2. Sprint Backlog

Sprint 1

Sprint 2

Sprint 3

Sprint 4

Sprint 5

3. Análisis tecnológico

3.1. Lenguajes, librerías y herramientas

3.1.1. Lenguajes

3.1.2. Librerías

3.1.3. Herramientas

4. Diseño de la solución

4.1. Diagramas

5. Recreación

5.1. Lista de APIs

5.2. Modelado SLA4OAI

5.3. Clúster HPC

5.4. Librería

5.5. Conclusión de resultados

6. Conclusiones

Bibliografía

- [1] Mark Winteringham. Testing web apis, 2022. 1st edition. Shelter Island, New York: Manning Publications Co. ISBN 1-63835-153-8.
- [2] Open APIs - State of the Market 2011, May 2011. URL <https://es.slideshare.net/jmusser/open-apis-state-of-the-market-2011>. [Online; accessed 22. Apr. 2024].
- [3] APIs show Faster Growth Rate in 2019 than Previous Years, July 2019. URL <https://web.archive.org/web/20191212210454/https://www.programmableweb.com/news/apis-show-faster-growth-rate-2019-previous-years/research/2019/07/17>. [Online; accessed 22. Apr. 2024].
- [4] 2023 State of the API Report | Brought to You by Postman, March 2024. URL <https://www.postman.com/state-of-api/api-global-growth/#api-global-growth>. [Online; accessed 21. Apr. 2024].
- [5] Colaboradores de los proyectos Wikimedia. API - Wikipedia, la enciclopedia libre, February 2024. URL <https://es.wikipedia.org/w/index.php?title=API&oldid=158385052>. [Online; accessed 22. Apr. 2024].
- [6] Postman api platform — sign up for free. <https://www.postman.com/>. (Accessed on 04/24/2024).
- [7] Alberto Martin-Lopez, Sergio Segura, and Antonio Ruiz-Cortés. Restest: automated black-box testing of restful web apis. In *Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis, ISSTA 2021*, page 682–685, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450384599. doi: 10.1145/3460319.3469082. URL <https://doi.org/10.1145/3460319.3469082>.
- [8] Alberto Martin-Lopez, Sergio Segura, and Antonio Ruiz-Cortés. Online testing of restful apis: promises and challenges. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/FSE 2022*, page 408–420, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450394130. doi: 10.1145/3540250.3549144. URL <https://doi.org/10.1145/3540250.3549144>.