

Evaluación de la usabilidad de APIs web

Ariel Machini¹ and Sandra Casas²

¹ Centro de Investigaciones y Transferencias Santa Cruz (CONICET/UNPA/UTN)
amachini@conicet.gov.ar

² Instituto de Tecnología Aplicada (ITA) - Universidad Nacional de la Patagonia
Austral (UNPA UARG)

Abstract. En los últimos años las APIs web se han convertido en componentes clave para agilizar el proceso de construcción de aplicaciones. Debido a esto, resulta importante que estas sean fáciles de aprender y utilizar para no reducir la productividad de los programadores ni obstaculizar el proceso de desarrollo. La usabilidad de las APIs es entonces considerada un factor fundamental para su correcta adopción, y si bien existen estudios que proponen soluciones para evaluarla y mejorarla, estos abarcan mayormente APIs locales; están más centrados sobre la documentación; suelen contemplar una limitada cantidad de características de usabilidad y, además, los resultados de investigaciones sobre usabilidad de APIs en general aún son insuficientes. Por estas razones, a través del presente trabajo se propone resolver algunos de los problemas recién mencionados mediante el estudio de la especificación OpenAPI y la construcción de un framework de evaluación de la usabilidad de APIs web.

Keywords: APIs Web · Métricas · Usabilidad · Framework · OpenAPI

1 Motivaciones, problemas y objetivos

En la última década, las APIs web se han convertido en uno de los pilares del desarrollo de aplicaciones modernas [1]. Además de permitir la reutilización de software y el acceso a servicios y recursos *de y entre* organizaciones, han producido una nueva y real perspectiva de negocio, la **Economía API** [2]. El mercado de APIs web es muy competitivo, por lo que se considera clave la facilidad de uso de las APIs para los desarrolladores potenciales; lo cual va en línea con [3], que resuelve que el valor de una API se identifica con su usabilidad. Por lo tanto, cualquier buena API debería ser fácil de aprender y usar y traducirse en la productividad de los desarrolladores [3]. Según la ISO 9241-11:1998, se define como "la medida en que un producto puede ser utilizado por usuarios específicos para lograr objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso específico". [5] afirman que "los diseñadores de API deben agregar la usabilidad como un criterio explícito de diseño y evaluación para no crear APIs inutilizables sin darse cuenta". Así, existen diversos estudios que proponen métodos y modelos de análisis, mejora y evaluación de la usabilidad de APIs, pero se observa: (i) están enfocados mayormente en APIs locales, (ii) están centrados principalmente en la documentación de las APIs, (iii) en cuanto a los enfoques

métricos, el problema común es que los trabajos están "algo" asociados con un modelo de usabilidad, pero las métricas derivadas no están relacionadas con una característica de usabilidad o están asociadas solo con algunas de estas características de usabilidad [6] y (iv) los resultados de investigaciones en usabilidad de APIs en general, y web en particular aún son insuficientes [6]. En la búsqueda de estrategias que sirvan tanto a proveedores y a consumidores de APIs web, se propone resolver algunos de los problemas planteados. Concretamente, se estudiará la especificación OpenAPI para el análisis y evaluación de la usabilidad de APIs web. ¿Qué aspectos de la usabilidad pueden medirse desde este artefacto?, ¿cómo hacerlo?, y ¿qué métricas de usabilidad se pueden obtener? son algunas preguntas que este trabajo pretende responder. El **objetivo general** de este trabajo es proporcionar soluciones consistentes, integrales y automáticas de análisis y evaluación de usabilidad de APIs web, y los **objetivos específicos** son (i) estudiar, identificar e integrar, métodos y heurísticas de análisis y evaluación de la usabilidad de APIs; (ii) formular teórica y conceptualmente un framework para el análisis y evaluación de la usabilidad de APIs web, aplicando el paradigma GQM (Goals-Questions-Metrics) y (iii) desarrollar una herramienta software que de soporte de automatización al framework propuesto.

2 Antecedentes

APIs como interfaces de librerías de códigos, frameworks o fuentes de datos, para liberarse de las tareas de programación de bajo nivel y/o acelerar el desarrollo [8]. A diferencia de las API locales, una nueva generación de APIs, las denominadas APIs de servicios web, ofrecen un enfoque sistemático y extensible para integrar servicios en aplicaciones [9,10]. La cantidad de servicios y recursos que proporcionan las APIs web disponibles al público ha aumentado rápidamente [11]. Varios estudios señalan que los desarrolladores han pasado de SOAP o RPC a la implementación de servicios web de transferencia de estado representacional (REST) como medio para que los consumidores utilicen sus servicios [11–14]. Esto puede corroborarse con compañías IT como Google, Facebook o Amazon, las cuales implementaron servicios REST para proporcionar un fácil acceso a sus recursos de datos mientras promueven sus negocios [11]. La arquitectura REST fue introducida en el año 2000 y se basa en los principios que sustentan la WWW [15]. A pesar de esto, REST es simplemente un estilo arquitectónico sin especificaciones estándar. Los desarrolladores de APIs web deben decidir y definir cómo se debe exponer una API, qué características debe poseer o cómo se proporciona la documentación. Este último factor ha sido problemático, ya que sin medios estándares para documentar las APIs, la tendencia fue utilizar texto para describir la API (muchas veces en lenguaje natural), que puede ser diverso en forma, estructura o profundidad y es obviamente un problema para sus consumidores [9]. Para superar este obstáculo, se propusieron varias especificaciones de la interfaz de las APIs web (WADL, API Blueprint, RAML), pero la que se ha consolidado como un estándar fue la especificación OpenAPI. Esta

es abierta, portátil, neutral e independiente del lenguaje, de definir los recursos y las operaciones de una API REST, ya sea en JSON o YAML.

3 Usabilidad y APIs web

Ejecutar tareas con efectividad, eficiencia y satisfacción [16] son factores determinantes en el éxito de las aplicaciones de software, por ello, la usabilidad adquiere una importancia cada vez mayor en el desarrollo de software [17]. Varias disciplinas y métodos proponen un diseño centrado en el usuario para asegurar que se cumplan los principios básicos de la usabilidad en una interfaz: facilidad de aprendizaje, facilidad de uso, flexibilidad y robustez. La usabilidad de un software se evalúa realizando pruebas para recopilar datos y hallar debilidades relacionadas con el uso de la misma. Existen cuatro formas básicas de evaluación: automática, empírica, formal e informal. Este proceso de evaluación implica actividades acordes al método empleado, sin embargo, estos comparten [18]: Captura, análisis y crítica.

La primera investigación en usabilidad de APIs [19] afirma que los programadores son usuarios de las APIs en algunos casos y, por ende, necesitan librerías que sean fáciles de aprender y usar, y proponen una lista de atributos a considerar al analizar la usabilidad de las APIs. Otros autores siguen un enfoque metodológico para la usabilidad de APIs, como [6]. Clarke [20] basó su trabajo en la adaptación del framework de *13 dimensiones cognitivas* de [21]. También propone seguir un enfoque centrado en el usuario para diseñar APIs usables que empleen escenarios, para así garantizar que reflejen las tareas que los usuarios desean realizar en lugar de sus detalles de implementación; y establece perfiles de desarrollador, que afectan la forma de analizar la usabilidad de la API. Sin embargo, el trabajo de Clarke tiende a ser demasiado abstracto para poder ser aplicado por programadores [22] y excluye aspectos importantes de usabilidad. Hay aportes como [3, 23] más enfocados en las necesidades de los programadores, que abordan problemas específicos o elecciones que deben tomarse en el diseño de API y proponen soluciones y cursos de acción. Otros trabajos, como [3], ofrecen una comprensión de qué son las APIs y por qué son utilizadas por desarrolladores. Establecen que el valor de una API se identifica en gran medida por su poder y, más importante, su usabilidad. Para que una API sea usable, debe ser eficiente, satisfactoria, fácil de aprender y memorizar y tener errores mínimos [4]. Otro tipo de contribuciones, refiere al desarrollo de directrices, lineamientos o guías para ayudar a los programadores a desarrollar APIs usables [24–28]. En una línea diferente, el problema de la usabilidad de las APIs se aborda desde el punto de vista de las métricas de software. [29] propuso 12 métricas estrictamente de complejidad. [30] propusieron medir la usabilidad de APIs como una función de su complejidad. [31] propusieron 9 métricas estructurales que se crearon específicamente para medir la usabilidad de la API. [32] definieron el *API Concepts Framework*, un marco extensible para medir la complejidad de la interfaz. Si bien el trabajo ignora algunos aspectos de usabilidad por la dificultad que representa el medirlos automáticamente, los otros aspectos de la usabilidad

están bien medidos. [33] publicaron una revisión del estado del arte de los estudios de usabilidad de API y afirman que "los diseñadores de API deben agregar la usabilidad como un criterio explícito de diseño y evaluación para no crear una API inutilizable sin darse cuenta" y promueven el uso de métodos centrados en el ser humano para mejorar la API.

4 Metodología y actividades

Como marco metodológico general se usará el enfoque de investigación DSR [34]. Siguiendo este enfoque, las actividades principales y técnicas que se aplicaran son: **Recopilación y análisis bibliográfico**. Se realizará una revisión sistemática para identificar e integrar el corpus relacionado a usabilidad de APIs (en general) y particularmente a APIs web. El resultado de esta actividad es la sistematización integrada y compatibilizada del vocabulario que es heterogéneo, modelos y métodos de análisis y evaluación, enfoques métricos, guías, directrices y pautas; **Formulación del framework de usabilidad de APIs web**. Para esta actividad se aplicará el paradigma GQM [7]. Como resultado se obtiene un conjunto de atributos/factores de usabilidad relevantes a APIs web y métricas asociadas; **Validación del framework y ajustes**. El framework obtenido en la actividad anterior será validado mediante una técnica empírica como las entrevistas a desarrolladores (proveedores y consumidores de APIs) y/o expertos en el tema. Los resultados obtenidos se usarán para mejorar el framework; **Desarrollo de una herramienta** que de soporte de automatización al framework propuesto; **Validación**. Se realizará una evaluación de un conjunto de al menos 100 APIs web reales con la herramienta y los resultados se analizarán estadísticamente.

References

1. Raemaekers, S., Van Deursen, A., & Visser, J. (2012). *Measuring software library stability through historical version analysis*. 28th IEEE International Conference on Software Maintenance (pp. 378-387). IEEE.
2. Tan, W., Fan, Y., Ghoneim, A., Hossain, M. A., & Dustdar, S. (2016). *From the service-oriented architecture to the web API economy*. IEEE Internet Computing, 20(4), 64-68.
3. Stylos, J., & Myers, B. (2007). *Mapping the space of API design decisions*. IEEE Symposium on Visual Languages and Human-Centric Computing (pp. 50-60). IEEE.
4. Nielsen, J. (1992). *The usability engineering life cycle*. IEEE Computer, 25(3), 12-22.
5. Myers, B. A., & Stylos, J. (2016). *Improving API usability*. Communications of the ACM, 59(6), 62-69.
6. Mosqueira-Rey, E., Alonso-Ríos, D., Moret-Bonillo, V., Fernández-Varela, I., & Álvarez-Estévez, D. (2018). *A systematic approach to API usability: Taxonomy-derived criteria and a case study*. Information and Software Technology, 97, 46-63.
7. Van Solingen, R., Basili, V., Caldiera, G., & Rombach, H. D. (2002). *Goal Question Metric (GQM) approach*. Encyclopedia of software engineering.

8. Dagenais, B., & Robillard, M. P. (2011). *Recommending adaptive changes for framework evolution*. ACM Transactions on Software Engineering and Methodology, 20(4), 1-35.
9. Curbera, F., Duftler, M., Khalaf, R., Nagy, W., Mukhi, N., & Weerawarana, S. (2002). *Unraveling the web services Web: an introduction to SOAP, WSDL, and UDDI*. IEEE Internet Computing, 6(2), 86-93.
10. Vinoski, S. (2008). *RESTful web services development checklist*. IEEE Internet Computing, 12(6), 96-95.
11. Maleshkova, M., Pedrinaci, C., & Domingue, J. (2010). *Investigating web APIs on the World Wide Web*. 8th IEEE European Conference on web Services (pp. 107-114). IEEE.
12. Renzel, D., Schlebusch, P., & Klamma, R. (2012). *Today's top "RESTful" services and why they are not RESTful*. International Conference on web Information Systems Engineering (pp. 354-367). Springer, Berlin.
13. Bülthoff, F., & Maleshkova, M. (2014). *RESTful or RESTless—current state of today's top web APIs*. European Semantic web Conference (pp. 64-74). Springer, Cham.
14. Kopecký, J., Fremantle, P., & Boakes, R. (2014). *A history and future of web APIs*. IT-Information Technology, 56(3), 90-97.
15. Fielding, R. T. (2000). *Architectural styles and the design of network-based software architectures*. Universidad de California, Irvine.
16. Harms, P., & Grabowski, J. (2014). *Usage-based automatic detection of usability smells*. International Conference on Human-Centred Software Engineering (pp. 217-234). Springer, Berlin.
17. Grau, X. F. (2000). *Principios Básicos de Usabilidad para Ingenieros Software*. JISBD (pp. 39-46).
18. Ivory, M. Y., & Hearst, M. A. (2001). *The state of the art in automating usability evaluation of user interfaces*. ACM Computing Surveys, 33(4), 470-516.
19. McLellan, S. G., Roesler, A. W., Tempest, J. T., & Spinuzzi, C. I. (1998). *Building more usable APIs*. IEEE Software, 15(3), 78-86.
20. Clarke, S. (2005). *Describing and measuring API usability with the cognitive dimensions*. Cognitive Dimensions of Notations 10th Anniversary Workshop (p. 131).
21. Green, T. R. G., & Petre, M. (1996). *Usability analysis of visual programming environments: A 'cognitive dimensions' framework*. Journal of Visual Languages & Computing, 7(2), 131-174.
22. Bore, C., & Bore, S. (2005). *Profiling software API usability for consumer electronics*. 2005 Digest of Technical Papers. International Conference on Consumer Electronics, 2005. (pp. 155-156). IEEE.
23. Ellis, B., Stylos, J., & Myers, B. (2007). *The factory pattern in API design: A usability evaluation*. 29th International Conference on Software Engineering (ICSE'07) (pp. 302-312). IEEE.
24. Jacques, M. (2004). *API Usability: Guidelines to improve your code ease of use*. Recuperado en Julio del 2021, de <http://www.codeproject.com/Articles/8707/APIUsability-Guidelines-to-improve-your-code-ease>
25. Henning, M. (2007). *API: Design Matters: Why changing APIs might become a criminal offense*. Queue, 5(4), 24-36.
26. Zibran, M. (2008). *What makes APIs difficult to use*. International Journal of Computer Science and Network Security (IJCSNS), 8(4), 255-261.

27. Zibran, M. F., Eishita, F. Z., & Roy, C. K. (2011). *Useful, but usable? Factors affecting the usability of APIs*. 18th Working Conference on Reverse Engineering (pp. 151-155). IEEE.
28. Grill, T., Polacek, O., & Tscheligi, M. (2012). *Methods towards API usability: A structural analysis of usability problem categories*. International conference on human-centred software engineering (pp. 164-180). Springer, Berlin.
29. Doucette, A. (2008). *On API usability: An analysis and an evaluation tool*. CMPT816-Software Engineering, Canada. Universidad de Saskatchewan.
30. de Souza, C. R., & Bentolila, D. L. (2009). *Automatic evaluation of API usability using complexity metrics and visualizations*. 31st International Conference on Software Engineering-Companion Volume (pp. 299-302). IEEE.
31. Rama, G. M., & Kak, A. (2015). *Some structural measures of API usability*. Software: Practice and Experience, 45(1), 75-110.
32. Scheller, T., & Kühn, E. (2015). *Automated measurement of API usability: The API concepts framework*. Information and Software Technology, 61, 145-162.
33. Daughtry, J. M., Farooq, U., Myers, B. A., & Stylos, J. (2009). *API usability: Report on special interest group at CHI*. ACM SIGSOFT Software Engineering Notes, 34(4), 27-29.
34. Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). *A Design Science Research methodology for information systems research*. Journal of Management Information Systems, 24(3), 45-77.