

## Ecosistemas de APIs web como un sistema socio-técnico: Un caso de estudio

**Sandra Casas**

[sicasas@uarg.unpa.edu.ar](mailto:sicasas@uarg.unpa.edu.ar)

<https://orcid.org/0000-0002-8289-6132>

Instituto de Tecnología Aplicada,  
Universidad Nacional de la Patagonia Austral  
Río Gallegos – Argentina

**Marcela Constanzo**

[mconstanzo@uarg.unpa.edu.ar](mailto:mconstanzo@uarg.unpa.edu.ar)

<https://orcid.org/0000-0001-8628-1509>

Instituto de Tecnología Aplicada,  
Universidad Nacional de la Patagonia Austral  
Río Gallegos - Argentina

**Graciela Vidal**

[gvidal@uarg.unpa.edu.ar](mailto:gvidal@uarg.unpa.edu.ar)

<https://orcid.org/0000-0003-4670-7944>

Instituto de Tecnología Aplicada,  
Universidad Nacional de la Patagonia Austral  
Río Gallegos - Argentina

**Diana Cruz**

[dcruz@uart.unpa.edu.ar](mailto:dcruz@uart.unpa.edu.ar)

<https://orcid.org/0000-0002-5793-8548>

Instituto de Tecnología Aplicada,  
Universidad Nacional de la Patagonia Austral  
Río Turbio - Argentina

Correspondencia: [sicasas@uarg.unpa.edu.ar](mailto:sicasas@uarg.unpa.edu.ar)

Artículo recibido 25 enero 2023 Aceptado para publicación: 25 febrero 2023

Conflictos de Interés: Ninguna que declarar

Todo el contenido de **Ciencia Latina Revista Científica Multidisciplinar**, publicados en este sitio están disponibles bajo

Licencia [Creative Commons](https://creativecommons.org/licenses/by/4.0/) 

Cómo citar: Casas, S., Constanzo, M., Vidal, G., & Cruz, D. (2023). Ecosistemas de APIs web como un sistema socio-técnico: Un caso de estudio. *Ciencia Latina Revista Científica Multidisciplinar*, 7(1), 9095-9120. [https://doi.org/10.37811/cl\\_rcm.v7i1.5039](https://doi.org/10.37811/cl_rcm.v7i1.5039)

## **RESUMEN**

En torno a las APIs web se ha generado un ecosistema digital dinámico, que incluye personas, empresas, herramientas, nuevas actividades y procesos de desarrollo, todo ello impulsado por la API Economy. Los ecosistemas de API web (EAW) son redes complejas de empresas interdependientes que se benefician colectivamente de los efectos de red basados en la cooperación y la competencia entre esas empresas. Una forma de analizar y comprender los sistemas complejos es el enfoque socio-técnico (S-T). De acuerdo con las revisiones realizadas para este trabajo, no se registran estudios S-T en relación con las APIs web. Proponemos analizar los EAW desde una perspectiva S-T, mediante un caso de estudio exploratorio y descriptivo ejecutado en tres EAW actuales. Hemos identificado los componentes de las dimensiones S-T que tienen en común, varias diferencias, en los roles de los actores, la infraestructura y las normas, las fortalezas de cada EAW, como los niveles de relación con el contexto. Finalmente, dejamos planteadas nuevas preguntas sobre este campo de conocimiento que ayuden a construir una agenda de investigación.

***Palabras clave:** API web; sistema socio – técnico; economía api; plataforma digital; gestión de API*

## Web API ecosystem as a socio-technical system: a case study

### ABSTRACT

Around the web APIs, a dynamic digital ecosystem has been generated, which includes people, companies, tools, new activities and development processes, all driven by the API Economy. Web API ecosystems (EAW) are complex networks of interdependent businesses that collectively benefit from network effects based on cooperation and competition among those businesses. One approach to analyzing and understanding complex systems is socio-technical (S-T). According to the reviews carried out for this research, there are no S-T studies in relation to web APIs. We propose to analyze WAEs from an S-T perspective, through an exploratory and descriptive study case executed in three current WEA. We have identified the components of the S-T dimensions that have in common various differences in the roles of the actors, infrastructure, norms, and the strengths of each WEA, as well as levels of relationship with the context. Finally, we raise new questions in this field of knowledge that help build an updated research agenda.

*Keywords: Web API; socio-technical system; api economy; digital platform; api management*

## INTRODUCCIÓN

Una API (Application Programming Interface) es un conjunto de definiciones y protocolos que se usa para diseñar e integrar software de las aplicaciones web o móviles. Las APIs web conectan aplicaciones de software usando la red, permitiendo a las organizaciones compartir datos, lógica y funcionalidades diversas, constituyéndose el pilar del software moderno. Desde un simple web mashup para visualizar datos de diversas fuentes externas (HousingMaps), hasta complejas aplicaciones software basadas en microservicios (Netflix), usan APIs web generando nuevo valor.

La creación de la arquitectura REST en la tesis doctoral de Fielding (2000), marcó un punto de inflexión en el paradigma de servicios web. Varios autores señalan que la implementación de servicios web de transferencia de estado representacional (REST), como la tecnología más usada, para que los consumidores utilicen sus servicios (Bülthoff y Maleshkova, 2014; Kopecký et. al, 2014; Maleshkova et. al, 2010; Renzel et. al, 2012). Esto además está impulsado por las principales compañías IT como Google, Facebook o Amazon, que han implementado servicios REST para proporcionar un fácil acceso a sus valiosos recursos de datos, mientras promueven sus negocios (Maleshkova et. al, 2010). Alrededor de las APIs web, se ha generado un dinámico y pujante ecosistema, como demuestra el reporte de estado de mercado de API Landscape (Apidays, 2022), que abarca personas, empresas, herramientas, nuevas actividades, todo impulsado fuertemente por una real perspectiva de negocio, denominada Economía API (Tan et. al, 2016). Como otros ecosistemas digitales, los Ecosistemas de API Web (EAW) son redes complejas de empresas interdependientes que se benefician colectivamente de los efectos de la red basados en la cooperación y la competencia entre dichas empresas (Beltagui et. al, 2020). Los EAW se soportan o despliegan en plataformas digitales (Bogers, et. al, 2019), así quedan enmarcados como Ecosistemas de Plataformas (EP) (Costa et. al, 2020).

Varios artículos de revisión abordan diversos temas de relación a APIs web. La usabilidad (Mosqueira-Rey et. al, 2018), evolución (Koci et. al, 2019), documentación (Nybom et. al, 2018), gestión (Mathijssen et. al, 2020), especificación (Casas et. al, 2021), dan cuenta del estado del arte. En particular, en la revisión de Ofoeda et. al (2019) señalan que los temas dominantes en la investigación pertenecen al diseño y la usabilidad, y que hacen

foco al dominio tecnológico. El estudio recomienda que la investigación de API también debe considerarse desde una perspectiva socio-técnica (S-T).

El concepto de sistemas S-T se desarrolló para capturar las interacciones complejas entre humanos, máquinas y entornos organizaciones (Emery y Trist, 1960). Los estudios S-T no son habituales en la Ingeniería de Software como indican Baxter y Sommerville (2011). Este trabajo aborda el vacío que se plantea en lo señalado por Baxter y Sommerville (2011) y que es reforzado por Ofoeda et. al (2019). Proponemos abrir la discusión y analizar el ecosistema de APIs web desde una mirada S-T, de cara a comprender mejor estas dinámicas. Con este propósito, ejecutamos un caso de estudio exploratorio y descriptivo sobre tres EP actuales para responder la siguiente pregunta, ¿Cómo se estructuran y componen los ecosistemas de API web? El caso de estudio se apoyó en un framework que formulamos para la observación, el cual nos permitió ordenar, estructurar y categorizar los datos recolectados desde un enfoque S-T.

### **SISTEMAS SOCIO-TÉCNICO (S-T)**

Los sistemas S-T son sistemas en los que los humanos interactúan con la tecnología (hardware o software) para lograr un objetivo, el cual se usa ampliamente para describir diversos sistemas complejos. Debido a la presencia del comportamiento humano y al constante cambio y evolución de la tecnología, tales sistemas cambian constantemente y son difíciles de definir.

Un enfoque clásico de análisis S-T es el propuesto por Kingdon (1995), afirma que las organizaciones están constituidas de tres subsistemas o elementos principales: (i) Sistema técnico o de tareas, que involucra el flujo de trabajo, la tecnología empleada, las actividades requeridas por la tarea. (ii) Sistema gerencial o administrativo, que involucra la estructura organizacional, las políticas, los procedimientos y las normas, el sistema de incentivos y de sanciones, la toma de decisiones y el empleo de elementos para facilitar los procesos administrativos. (iii) Sistema Social, que involucra la cultura organizacional, con los valores, las normas y la satisfacción de las necesidades personales, tales como el nivel motivacional de los colaboradores y sus actitudes individuales.

Según Badham et. al (2000), se establecen claves de sistemas S-T abiertos: los sistemas deben tener partes interdependientes; los sistemas deben adaptarse y perseguir objetivos en entornos externos; los sistemas tienen un entorno interno que comprende subsistemas técnicos y sociales separados pero interdependientes; los sistemas tienen

equifinalidad (los objetivos de los sistemas se pueden lograr por más de un medio); el desempeño del sistema se basa en la optimización conjunta de los subsistemas técnico y social.

Existen diversos enfoques para analizar y comprender los comportamientos de los sistemas S-T; sin embargo, varios de estos enfoques analizan los S-T desde la cosmovisión, el contexto del problema y el propósito del sistema de una determinada disciplina (Elatlassi y Narwankar, 2016).

Baxter y Sommerville (2011) analizan las razones por las cuales los enfoques S-T no se aplican al desarrollo de software o en el área de Ingeniería de Software, e identifican algunos de los problemas con los métodos de diseño socio técnicos más conocidos. Analizan los enfoques cubiertos por la extensa revisión de Mumford (2006), también identifican otros enfoques (metodología de sistemas soft, análisis cognitivo del trabajo, método S-T para el diseño de sistemas de trabajo, análisis etnográfico del lugar de trabajo, diseño contextual, ingeniería de sistemas cognitivos, y diseño centrado en el usuario), que abarcan ideas socio-técnicas. Consideran que estos otros enfoques pueden ayudar a informar el desarrollo de sistemas S-T. Identifican un conjunto de problemas que deben ser superados para que los sistemas S-T, sean aplicados al ámbito de la ingeniería de software. En función de ello, proponen un framework pragmático para los sistemas S-T y la ingeniería de software. Empero, este framework está enfocado a software que involucra solo a una organización. Es por ello, que los autores en la agenda de investigación argumentan que prácticamente todos los enfoques existentes para diseño de sistemas S-T se basan en la suposición de que los sistemas están ubicados dentro de una única organización coherente donde las partes interesadas del sistema tienen valores y suposiciones culturales similares. Cuestión que no cubre la tendencia creciente a crear sistemas globales, que pueden involucrar a varias organizaciones dispares que se encuentran en todo el mundo. De manera similar, los equipos involucrados en proyectos de ingeniería de sistemas complejos están distribuidos geográficamente a través de zonas horarias y culturas. Los temas de investigación en esta área de sistemas globales y la globalización de la ingeniería de sistemas incluyen diversas preguntas que, según los autores, aún no se han contestado.

La revisión de Kapoor et. al (2021) analiza 70 artículos (1999-2019) con relación a las PE desde un enfoque S-T. Se conceptualizan conceptos claves como ecosistema y

plataforma. Usaron para el análisis un framework, que es una adaptación de Lyytinen y Newman (2008), basado en cuatro dimensiones, aspectos técnicos, tareas, actores y estructura. A la vez, cada uno de estos ejes se abre en dos o más sub-dimensiones de análisis. Identifican hallazgos, brechas de conocimiento en relación a las dimensiones analizadas y las superposiciones (relaciones o intersecciones) entre las mismas. Señala como uno de los resultados más relevantes, que los estudios existentes están interesados principalmente en el aspecto técnico de las plataformas y otros aspectos tangibles (componentes y recursos) y menos en el aspecto social. La revisión se enmarca en disciplinas relacionadas al ámbito empresarial, de negocios y comercio. Las PE que fueron objeto de los artículos analizados (juegos, tiendas de app móviles, viajes, hotelería, transporte, sectores industriales del metal, compañías IT), difieren de los ecosistemas de APIs web.

## **METODOLOGÍA**

Este trabajo se llevó a cabo como un caso de estudio exploratorio y descriptivo que estudió tres ecosistemas de APIs Web actuales, usando las directrices de Creswell (2014), Fabbri (2020) y Runeson et. al (2012). La Figura 1 esquematiza el diseño metodológico aplicado. La selección de los ecosistemas RapidAPI, APILayer y Nubentus obedeció a que muestran rasgos bien diferenciadas, en cuanto a, permanencia en el mercado, posición en el mercado (nivel de madurez y consolidación), cantidad de usuarios (consumidores y proveedores), cantidad de APIs y dominio.

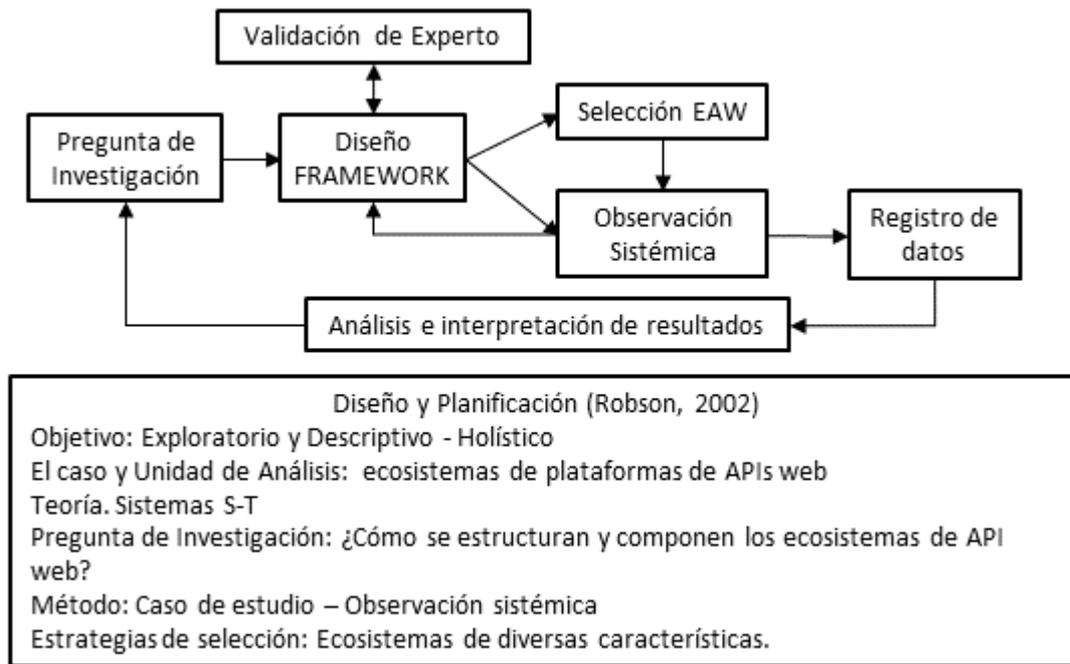
Los datos se recopilaron principalmente mediante la técnica de observación sistemática u objetivamente estructurada (Cresswell, 2014; Fabbri, 2020) de las plataformas en las que los EAW están emplazados usando un framework que diseñamos y más adelante describimos. Este framework nos permitió ordenar, estructurar, categorizar los datos recogidos, desde un enfoque S-T.

Los datos se recopilaron principalmente mediante la técnica de observación sistemática u objetivamente estructurada (Cresswell, 2014; Fabbri, 2020) de las plataformas en las que los EAW están emplazados usando un framework que diseñamos y más adelante describimos. Este framework nos permitió ordenar, estructurar, categorizar los datos recogidos, desde un enfoque S-T.

Los datos se recopilaron principalmente mediante la técnica de observación sistemática u objetivamente estructurada (Cresswell, 2014; Fabbri, 2020) de las plataformas en las

que los EAW están emplazados usando un framework que diseñamos y más adelante describimos. Este framework nos permitió ordenar, estructurar, categorizar los datos recogidos, desde un enfoque S-T.

**Figura 1.** Diseño metodológico



El proceso de recogida de datos se realizó a partir de una indagación en las plataformas, la cual implicó, la inspección de las mismas, mediante el análisis de los distintos contenidos (documentación, videos, formularios, plantillas, blogs, etc.), participación de las comunidades y recursos externos recomendados desde las plataformas. Se realizaron consultas a los mails y cuentas en las redes sociales, se efectuaron pruebas simples de los asistentes, herramientas en línea de soportes, búsquedas, etc. A partir de estas experiencias, se determinó la información requerida por el framework. Para ello, se realizaron registros mediante un formulario diseñado a tal fin, el cual se estructuró en dos niveles, primero según las dimensiones del framework y para cada dimensión según los componentes. Para cada componente previamente se generó una especificación y caracterización del mismo (por ejemplo, SDK y CLI en herramientas). Los registros obtenidos de las observaciones y experiencias fueron revisados en cada iteración por al menos dos de las autoras. Es válido mencionar que esta aplicación del framework a los ecosistemas mencionados, produjo una retroalimentación en cuanto a la definición original del framework, en los primeros momentos de la indagación. El proceso de

aplicación del framework a los EAW mencionados se realizó en el primer semestre del año 2022.

**Framework S-T.** La construcción del Framework S-T para analizar los ecosistemas de APIs web, es una adaptación de Kingdon (1995) en cuanto a que define tres dimensiones con características similares. Toma la identificación de actividades principales y sus requerimientos de Baxter y Sommerville (2011), y la inclusión de la identificación de contexto de Elatlassi y Narwankar (2016). El framework se plantea en torno a premisas, actividades y requerimientos, dimensiones y contexto.

Premisas. Conjunto de características distintivas que un sistema o PE debe cumplir para ser considerado un ecosistema de API web. (i) Global. No se establece en torno a un espacio físico y/o geográfico único, sino que se establece en torno a una plataforma digital. (ii) Multi-organizacional. El componente social-humano está compuesto por personas, equipos o grupos que pertenecen a diferentes y diversas organizaciones (empresas, gobiernos, etc.). (iii) Abierto. Pueden participar del ecosistema cualquier persona, equipo, u organización en tanto acepte y cumpla las normas establecidas.

Actividades y Requerimientos. Las dos actividades principales son, consumir APIs web y producir APIs web, las que además permiten identificar a los actores del ecosistema y sus respectivos roles. Los desarrolladores de aplicaciones necesitan APIs web para conectar sus aplicaciones con fuentes de datos externas o incorporar diversas funcionalidades, y así añadir valor al software que producen. Los proveedores de APIs, ofrecen sus APIs para obtener beneficios económicos directos o indirectos. Es posible identificar más actividades en el ecosistema, pero todas están ligadas a cumplir las actividades definidas de consumir y/o proveer. Estas actividades son independientes del tipo de API (REST, RESTful, etc.) e incluso el ámbito de uso y visibilidad (interna, externa, asociada, privada, pública, etc.).

Para consumir una API web un desarrollador requiere que el ecosistema proporcione, acceso a la plataforma, herramientas, documentación, un catálogo de APIs web, información técnica del desempeño (analíticas) y garantías técnicas de la APIs (SLA), información de los costos, dispositivos para resolver problemas, facilidades para comunicarse con otros consumidores y los proveedores de la API, estímulos para continuar en el ecosistema, etc.

Los proveedores de APIs web requieren del ecosistema, acceso a la plataforma, herramientas, soporte a la gestión del ciclo de vida de la API, un gateway (puerta de enlace) con diversas capacidades, medios y mecanismos para obtener beneficios, estímulos para continuar en el ecosistema, información del consumo de la API (analíticas), medios para comunicarse con los consumidores, etc.

Agrupando por función y objetivo estos requerimientos y adaptando el esquema propuesto por Kingdon (1995), en esta instancia establecemos tres dimensiones o sistemas principales del EAW.

**Dimensión Social:** personas o grupos de personas que participan o son parte del ecosistema. En principio los desarrolladores (consumidores) en forma individual, y otros actores. Por otro lado, los proveedores (productores) de las APIs, en general su presencia en el ecosistema es en forma corporativa. La comunidad del ecosistema, con frecuencia es el conjunto de desarrolladores que comparten problemas, soluciones, consejos, tips y experiencias en foros creados a tal fin en la plataforma del ecosistema. Además, incluimos en esta dimensión, las acciones de estímulo y/o incentivación, que desde la plataforma se proponen y que buscan, lograr fidelidad de los individuos o equipos con el ecosistema, y/o atraer potenciales o nuevos consumidores y/o productores a que se sumen al ecosistema (evangelización). Un tercer actor, que actúa en el ecosistema, pero de una manera menos explícita y visible corresponde a aquellas personas o grupos que se ocupan de la gestión de la plataforma u otras tareas que tienen el propósito de dar soporte a consumidores y productores de APIs. Este actor es responsable del establecimiento de la mayoría de las normas y políticas del ecosistema.

**Dimensión Infraestructura:** conjunto de recursos y servicios ofrecidos desde la plataforma para consumir o proveer APIs web. Entre estos, servicio de registro, acceso, credenciales y claves API (gestión de clave API); herramientas y aplicaciones (editores en línea, plantillas o formularios de configuración o diseño, SDK (Software Development Kit), CLI (Command-Line Interface), generadores de pruebas, gestión de cambios, generadores de documentación, gestores de tickets, etc.); documentación y capacitación (especificaciones, tutoriales, guías, cursos, artículos, videos, ejemplos, rebinar o meeting, etc.); catálogos y buscadores de APIs; analíticas; información del uso o desempeño de la API; mecanismos de seguridad (autenticación, autorización, privacidad de datos, certificados digitales, gestión de claves y certificados, protección denegación de servicio;

servicios de gateway (gestión del tráfico, traducción de interfaz, almacenamiento en cache, enrutamiento de servicios, orquestamiento de servicios).

Dimensión Normas: conjunto de reglas, normas, pautas y políticas que se imponen para garantizar el armonioso ambiente en el ecosistema. Refieren a los compromisos que asumen los consumidores y proveedores, como la infraestructura. Forman parte de esta dimensión, las políticas y términos de uso de la plataforma donde reside el ecosistema, el modelo de monetización, los acuerdos a nivel de servicio (SLA), entre otros.

Contexto: El ecosistema API web se despliega básicamente en una determinada plataforma pero se extiende en el contexto para aumentar las capacidades del ecosistema. La comunidad (dimensión social) se extiende a las redes sociales si desde la plataforma se promueven la interacción en cuentas oficiales o se establece un espacio de discusión en Stack Overflow. La infraestructura se extiende cuando por ejemplo se depositan o comparten recursos (herramientas, documentación, código u otro) en repositorios externos como GitHub. Por otro lado, la plataforma en la que se asienta el ecosistema comúnmente está montada sobre la nube de alguna empresa que le proporciona diversos servicios y recursos, que actúa como un proveedor del ecosistema. Por último y no menor, las empresas u organizaciones empleadoras tanto los desarrolladores (consumidores) o los proveedores de APIs, son parte de este contexto del ecosistema. Todas estas entidades del contexto del EAW tienen sus propias normas y políticas, las cuales son compatibilizadas por el ecosistema.

## RESULTADOS

A continuación, se presenta una reseña de los tres ecosistemas de APIs web analizados con el framework y en la Tabla I, un resumen comparativo de los mismos.

**RapidAPI.** Se presenta como una plataforma de API de última generación e innovación, y la más grande del mundo, con más de 3.000.000 de desarrolladores y 35.000 APIs. Ofrece una serie de herramientas propias como productos, los cuales asisten a los consumidores y proveedores, en las actividades principales cubriendo los diversos requerimientos que se plantean. RapidAPI enfatiza la usabilidad de estos productos, en tanto garantiza que los mismos mejoran la experiencia de los usuarios al contar con IU (Interfaz de Usuario) intuitivas y diversas capacidades que facilitan el uso y mejoran la productividad (admisión de distintos tipos y categorías de APIs, integración con otras herramientas y plataformas,

generación de código en los lenguajes más populares, importación/exportación de especificaciones, manejo de esquemas de seguridad, etc.).

### **Dimensión Social.**

Se reconocen los siguientes actores, Consumidores a nivel desarrollador individual, equipos y empresa; Proveedores, productores de las APIs; Gestores de la plataforma, un equipo de profesionales que brindan ayuda a través de los diferentes medios de contacto, para facilitar, tanto a proveedores como consumidores, el uso de las aplicaciones que ofrecen.

### **Dimensión Infraestructura.**

Las herramientas RapidAPI están destinadas a cubrir diferentes necesidades de consumidores y productores. RapidAPI Hub está dirigida a desarrolladores que requieren consumir e incluso integrar APIs, incluye diversos servicios, búsqueda y catálogo de APIs, acceso a la documentación, pruebas, monitoreo con una única clave de API y desde el mismo tablero o espacio de trabajo. RapidAPI Client está dirigida a productores, proporciona todas las funciones para diseñar, describir, depurar, probar e interactuar con las APIs. Acepta cualquier tipo (REST, SOAP, GraphQL, etc.) y categoría (interna, externa, asociada, etc.) de API, y admite distintos esquemas de autenticación, importa y exporta desde estándares (OpenAPI) y generar fragmentos de código para lenguajes populares, entre otras funcionalidades. RapidAPI for Teams, es una herramienta destinada a equipos que están haciendo la transición de sus arquitecturas a microservicios, para ello, se ofrece un espacio de trabajo privado que favorece la colaboración y en el cual pueden publicar, compartir e integrar las APIs internas y externas. RapidAPI Enterprise Hub es una herramienta para empresas que permite crear una plataforma personalizada para administrar y gobernar todo tipo y categoría de APIs, conexión con otras plataformas o gateways. Las plataformas creadas tendrán diversas funcionalidades para exponer, crear, integrar, administrar y monetizar APIs, entre otras. Estos productos además se integran desde la herramienta RapidAPI Studio. RapidAPI Testing permite realizar pruebas funcionales de las APIs y las integraciones creadas con los demás productos RapidAPI. Ofrece tres opciones para generar pruebas (generación de pruebas visual, automatizada o basada en código). RapidAPI Testing se integra con las principales herramientas de gestión de incidentes y versiones como GitHub. Otra funcionalidad de este producto es el Monitoreo Global, permite a los desarrolladores, equipos y empresas supervisar las

pruebas para garantizar el rendimiento de las apps, APIs y microservicios, genera alertas e informes de ejecución detallados entre otras posibilidades.

La documentación se compone de guías y tutoriales, blogs para comenzar a trabajar con RapidAPI, Get Started (dirigida a los consumidores). Además, breves videos explicativos y muy didácticos, que están separados en secciones y clasificados según sean destinados a desarrolladores o proveedores. También se incluyen guías, tutoriales, webinars, ebooks, hojas de datos, reportes y white papers.

El Catálogo de RapidAPI está organizado en varias categorías que a la vez se descomponen en subcategorías, según ciertos criterios tales como popularidad, tendencia y destacadas. Permite la búsqueda por la denominación de la app.

Las analíticas se representan por tres métricas que se muestran en cada lista de API en RapidAPI Hub, popularidad, latencia promedio, nivel de servicio.

#### **Dimensión Normas.**

RapidAPI presenta cuatro niveles de normas. Términos de servicio dirigido a proveedores y consumidores. Política de derechos de autor, derechos y obligaciones aplicable al contenido de RapidAPI como el de los usuarios. Políticas de privacidad (de datos), como RapidAPI recopila, utiliza y divulga la información y se aplica a la información recopilada cuando interactúa con RapidAPI a través del sitio web o alguno de los servicios en línea.

#### **Cookies y uso de cookies.**

El modelo de monetización es por producto (RapidAPI Hub, RapidAPI Client, etc.). En cada caso presenta diferentes planes que se pueden definir como gratuitos, freemium o pagos (con cuotas y límites). La suscripción puede ser mensual o planes de pago por uso. Ofrece planes privados que están disponibles solo por invitación.

Contexto. Desde la plataforma se difunden las cuentas oficiales a través de las redes sociales, Facebook, Twitter, YouTube y LinkedIn. DocuSign para la suscripción electrónica de contratos. RapidAPI Testing se integra con las principales herramientas de gestión de incidentes y desarrolladores, como GitHub, PagerDuty, Slack y Twilio.

**APILayer.** Se presenta como una marketplace líder del mercado de SaaS, en el que se ofrecen APIs propias de APILayer (35%) y APIs de terceros. Es una plataforma abierta que publica solo APIs curadas, para ello se lleva a cabo una cuidadosa selección de las mismas con el fin de no ofrecer repetidas en una misma categoría y así facilitar el acceso a los

consumidores. En la actualidad es posible acceder a más de 80 APIs y cuenta con más de un millón de usuarios.

#### **Dimensión Social:**

En APILayer los productores son, los miembros del equipo de APILayer, los desarrolladores independientes (externos) y empresas que publican sus APIs para ser comercializadas. Los consumidores de las mismas son los suscriptores a las APIs. La gestión de la plataforma la lleva a cabo los miembros de APILayer. La interacción de los mismos se presenta en diferentes formas, el equipo de soporte técnico de APILayer responde a las consultas realizadas por Chat Live o tickets específicos sobre una API y tópico particular. Otra interacción se presenta cuando los usuarios envían un recurso para ser publicado conjuntamente. El equipo de editores de APILayer realiza las observaciones y como recompensa los autores reciben créditos para ser utilizados en suscripciones a las APIs del Marketplace. Los productores de las APIs externas proveen asistencia a los productores y consumidores desde sus plataformas a las cuales se accede desde la de APILayer.

#### **Dimensión Infraestructura:**

El catálogo ordena las APIs en 18 categorías, accesibles desde el buscador del marketplace. Los consumidores de APIs deben registrarse, este perfil los habilita a gestionar sus contraseñas, suscripciones, métodos de pago y las APIs key requeridas en las solicitudes a las APIs. Además, tienen acceso al servicio Live Demo que permite probar las APIs, desde el navegador, sin necesidad de instalar ni codificar nada. Para esto deben cargar los parámetros, entonces pueden realizar pruebas utilizando diferentes lenguajes (JavaScript, Python, PHP, Swift entre otros). Un servicio para generar (Run Code) código a partir de los datos ingresados, el que puede ser copiado y utilizado para iniciar un proyecto.

Cada API cuenta con su descripción y documentación, lo cual puede estar disponible en la plataforma de APILayer o en el sitio del proveedor. En APILayer las APIs se describen mediante su funcionalidad y a través de ejemplos. El consumidor puede consultar diversas métricas de las APIs en uso, como límite de solicitudes por mes, solicitudes restantes, límite de solicitudes por día y solicitud limite restante en el día.

Los productores deben proveer la documentación en formato OpenAPI, para esto tienen acceso a API provider Postman (una interfaz para documentar la descripción de la API).

Cuando se desarrolla una nueva versión de una API, se debe actualizar la documentación, la que debe ser probada y verificada antes de su lanzamiento.

APILayer emplea distintos mecanismos de seguridad y privacidad (SHA256, Reglamento general de protección de datos, etc.).

APILayer ofrece distintas opciones de alojamiento de APIs externas basadas en las nubes de terceros (Amazon Web Services, Google Cloud Plataform, etc.) o en la propia. La infraestructura de APILayer se completa con diversos servicios de terceros para mejorar su funcionalidad, entre los cuales se encuentran las herramientas Klaviyo, Referral, Stripe, Google Analytics, Hotjar, Hubspot Docker, y Datadog.

#### **Dimensión Normas:**

APILayer establece normas legales, términos y condiciones, políticas de privacidad (GDPR), políticas de cookies y además cuenta con un acuerdo de servicios. Las normas son las definidas por Idera Inc.

El modelo de monetización se basa en cinco planes (free, starter, pro, enterprise y custom). Estos varían de acuerdo con la cantidad de solicitudes diarias y mensuales, cantidad de endpoints, soporte técnico, tipos de uso (comercial/personal), soporte de mail y otras funcionalidades, pero los precios de cada plan de las APIs externas son establecidos por los propios proveedores, aunque APILayer realice sugerencias.

Contexto: El ecosistema de APILayer se extiende a sitios de terceros (proveedores) a través de enlaces y también por medio de las redes sociales Twitter, Facebook, Instagrams, LinkedIn las cuales generan espacios en los cuales la comunidad accede a diferentes tópicos sobre los cuales puede interiorizarse, debatir, compartir y comentar. También cuenta con una cuenta en GitHub la que permite acceder a diversos recursos en repositorios, proyectos y paquetes. Como se mencionó anteriormente APILayer utiliza herramientas y servicios de terceros (Klaviyo, Referral, etc.) y ofrece alojamiento en la nube de distintos proveedores. Es posible acceder desde APILayer al portal del proveedor de APIs y entonces obtener la descripción, documentación, recursos, planes, comunidad, aspectos legales, condiciones de servicios y políticas específicas. Las normas definidas por Idera Inc.

**Tabla 1.** Resumen del análisis S-T de RapidAPI, APILayer y Nubentos

	RapidAPI	APILayer	Nubentos
<b>Premisas</b>	Global – Abierto – Multi-organizacional		
<b>Actividad</b>	Consumir y proveer APIs de terceros	Consumir y proveer APIs de terceros y propias	Consumir y proveer APIs de terceros de salud
<b>Social</b>	Actores: consumidores de APIs web, proveedores de APIs web, gestores de la PE		
	Mecanismos de interacción basado en la usabilidad y experiencia de usuarios	El gestor de la PE es proveedor	Mecanismos de interacción dinámicos y directos
		Mecanismos de interacción dinámicos	
Incentivos			
<b>Infraestructura</b>	Herramientas específicas que cubren diversos requerimientos de consumo y producción de APIs Web. Abundante documentación. Analíticas a nivel API	Pruebas y generadores en línea. Documentación básica	SDK para diversas tecnologías. Documentación básica. Analíticas a nivel plataforma
<b>Normas</b>	Términos de uso y servicio, política de seguridad, cookies y derecho de autor Modelo de monetización basado en planes		
	Planes basados en uso de los productos con opciones de suscripción. Planes privados por invitación	Planes por API. Planes personalizados	Planes combinados
<b>Contexto</b>	Redes Sociales		
	Contratos. Plataformas de gestión de incidentes y control de versiones	Proveedores de APIs web. Plataformas, herramientas y servicios (repositorios, prueba, alojamiento en la nube, pagos, monitoreo, analíticas, mensajería, etc.)	Proveedores de APIs web

**Nubentos.** Es una startup tecnológica que comenzó a funcionar a principio de 2019. Es un ecosistema orientado al dominio de la salud. Se presenta como una opción que permite a los consumidores ahorrar ya que pagan solo por el uso que se haga de las APIs. El modelo de monetización combinado y atención personalizada a consumidores parecen ser la principal estrategia para posicionarse en este nicho.

**Dimensión Social:** Se reconocen como actores, los desarrolladores de aplicaciones de la sanidad (clientes), los proveedores de APIs, y también el equipo técnico de la API Store. Las dudas sobre el uso del API Store, son evacuadas por el equipo de expertos, mientras que las preguntas sobre el funcionamiento o el uso de la API, son evacuadas por los

proveedores con su servicio de soporte. Los gestores ofrecen realizar reuniones periódicas con los consumidores para mejorar las prestaciones ofrecidas.

**Dimensión Infraestructura:** Para usar las APIs del API Store, solo se necesita un usuario registrado para poder suscribirse a cualquiera de los planes de consumo que publica cada API. Cada suscripción está asociada a una Aplicación, que le permite gestionar los recursos de la suscripción como tokens y claves de acceso. Una vez suscrito a un plan de consumo de una API, ya se puede interactuar con ella desde la consola online, e integrarla en los proyectos. Ofrece un servicio que consiste en realizar el diseño de interfaces para realizar las integraciones de APIs específicas a demanda del consumidor.

Ofrece analíticas que permiten analizar el uso real de las integraciones realizadas por los usuarios finales para controlar el nivel de consumo y las preferencias de uso en las aplicaciones; identificando áreas de mejora en los proyectos y así mantener bajos los costos. También ofrece un registro constante de la actividad, porcentaje de actividad del Gateway, Marketplace y Publisher, entre otros, durante un lapso de 30 días.

En relación con la Seguridad, ofrece transacciones cifradas de extremo a extremo, protocolo OAuth2 para todas las APIs, estandarización de la capa tecnológica y la forma de integrar soluciones de terceros en los proyectos de software.

Nubentos cuenta con un blog organizado por categorías y búsquedas, la documentación incluye guías, ebooks y videos. El directorio publica APIs para la sanidad, permitiendo que los desarrolladores las prueben y las integren autónomamente desde la nube a los proyectos. Se brindan diferentes maneras de ayudar a los desarrolladores, reserva de demo para mostrar cómo usar Nubentos, interactuar con expertos a través del chat (APIENS) y encontrar respuestas en el centro de ayuda.

Ofrece 10 SDK para cada API para facilitar la integración con el API Gateway, cubriendo las principales plataformas y tecnologías desarrollo (JavaScript, PHP, etc.).

**Dimensión Normas:** Nubentos tiene cuatro políticas, términos y condiciones de uso que rigen la relación comercial (acceso a servicios, obligaciones, responsabilidades y garantías, baja de servicios, confidencialidad, cambios, etc.); políticas de privacidad (responsabilidad, uso, cesión de los datos y derechos de los usuarios); políticas de cookies y política de precios (acceso, prueba, integración y consumo de APIs, publicación de APIs y modificaciones de precios).

El modelo de monetización es combinado. El consumidor debe suscribir primero a un plan de usuario (free, starter, basic, pro y business) y luego al plan de consumo de cada API. El plan de usuario da acceso a todas las funcionalidades del portal del desarrollador, y el pago de la tarifa correspondiente al plan de consumo, solo se hace efectiva cuando las integraciones de los consumidores son ejecutadas por los usuarios finales. Cada uno de los planes de consumo podrán ser free, freemium o paid, con tarifas por cantidad de llamadas a las API o tarifa plana.

Contexto. Desde la plataforma se difunden las cuentas oficiales a través de las redes sociales, Facebook, Twitter, YouTube y LinkedIn. Desde Nubentos, los consumidores pueden acceder a los portales de los proveedores de APIs, para hacer uso de los recursos y servicios que estos ofrecen.

## **DISCUSIÓN**

Los ecosistemas de APIs web, ofrecen APIs web para ser consumidas o provistas, en torno a una PE. Los EAW incluyen los componentes de un sistema S-T y cumplen las características señaladas por Badham et. al (2000). Además, intrínsecamente son globales, abiertos y multi-organizacionales. En consecuencia, los EAW desde una perspectiva S-T son entramados de componentes sociales, técnicos (o tecnológicos) y normas que configuran relaciones y dependencias complejas. A partir del caso de estudio exploratorio y descriptivo realizado, los datos recogidos y nuestra interpretación de los mismos, observamos que,

(i) Los roles, consumidor, proveedor y gestor que componen la dimensión Social están claros y diferenciados. Sin embargo, ninguno de los EAW en la PE trata a los consumidores y proveedores como una comunidad, disponiendo en la infraestructura la clásica herramienta de los foros. Lo cual difiere en parte con lo expuesto por De (2017) que incluye a la comunidad e indica que los foros de debate y los blogs que describen las experiencias de los desarrolladores ayudan a crear una comunidad de desarrolladores comprometida. Reafirmado por Mathijssen et. al (2020), que identifican como una de las capacidades de la Gestión de APIs a la administración de la Comunidad,

A los desarrolladores de aplicaciones a menudo les gusta conocer las opiniones de otros desarrolladores de la comunidad. Es posible que deseen colaborar y compartir sus aprendizajes y experiencias sobre el

uso de la API entre sí. Los blogs y foros forman una parte importante de la gestión de la colaboración y la comunidad. (p. 17),

y las prácticas asociadas de Soporte al foro y/o blogs de la Comunidad,

Una comunidad de desarrolladores puede consistir en un foro y/o blog para apoyar a los desarrolladores y ayudarlos a colaborar. (p. 21).

La comunidad se desarrolla y está vigente en el contexto, en las redes sociales, donde los tres EAW tienen presencia.

(ii) En relación con acciones de incentivación, solo APILayer despliega una acción directa (al compensar con créditos para el uso de APIs a quienes aporten artículos).

(iii) La facilidad de uso de la infraestructura tiene tres características, a) basada en la usabilidad y experiencia UX de los productos y/o herramientas que se ofrecen para consumir o producir APIs, y abundantes recursos de información y documentación, es la estrategia de RapidAPI, b) basada en mecanismos y dispositivos de interacción dinámicos y/o directos, a partir de asistentes interactivos, generadores de tickets, incluso reuniones personales, siendo las estrategias de APILayer y Nubentos; c) disponer de dispositivos que posibilitan la integración, con los proyectos de los consumidores y proveedores. Para ello, se proporcionan generadores de código en diversos lenguajes, SDK, uso (importación/exportación) de especificación OpenAPI, integración con otras plataformas de desarrollo, etc. Esta última es soportada en distintos niveles por los tres EAW.

(iv) Todos los EAW proporcionan las herramientas o servicios necesarios para consumir y proveer APIs. No obstante, RapidAPI y Nubentos son los desarrolladores de las herramientas que ofrecen, lo que les permite generar soluciones a medida, mientras que APILayer se apoya en herramientas de terceros (Postman, etc.) y dos herramientas que dispone en línea, lo que podría tener como ventaja, primero que los consumidores y proveedores quizás ya conozcan esas herramientas y luego que APILayer no es responsable del desarrollo de las mismas.

(v) Los recursos de documentación difieren en cantidad y tipo. RapidAPI proporciona una importante cantidad de documentación en diversos formatos cuyo objetivo es que los usuarios aprendan a usar las herramientas que se ofrecen y a partir de ello, consumir o proveer APIs, mientras que la documentación en Nubentos y APILayer es menor tanto en cantidad como en variedad, y refiere principalmente a las APIs.

(vi) El catálogo y servicio de búsqueda de las APIs en todos los casos se presenta ordenado por categorías, pero RapidAPI lo incluye como una función u operación más de las herramientas, mientras que APILayer y Nubentos lo incluyen como una funcionalidad destacada en el portal de la plataforma.

(vii) El modelo de monetización es el componente que mejor refleja las capacidades, estrategias y diferencias entre los EAW. En todos los casos, presenta variantes y opciones basado en el formato de planes. Los planes pueden incluir diversos ítems, además del consumo específico de las APIs (por cantidad de llamadas u otros parámetros), también suelen abarcar, el uso de las herramientas y/o servicios disponibles, el nivel de soporte o asistencia, tipo de alojamiento de APIs (en el caso de proveedores), etc. Sin embargo, hay inflexiones, RapidAPI y APILayer proponen opciones por fuera de los planes clásicos (plan por invitación y plan personalizado). APILayer, los planes son por API y el precio de cada plan lo establece el proveedor de la misma. Nubentos propone un plan basado en la combinación de dos tipos de planes.

(viii) La SLA en términos de parámetros de QoS (latencia, tiempo de respuesta, etc.), es dispar, no están claramente definidos como es el caso de Nubentos y APILayer, a diferencia de RapidAPI. En algunos casos, solo se proporcionan ciertas medidas de rendimiento de algunos servicios del gateway.

(ix) El contexto es importante. Los tres EAW se extienden para fortalecer o completar las capacidades de alguna de las dimensiones S-T. Como ya se ha mencionado, la comunidad como componente de la dimensión social, se desarrolla por completo en las redes sociales en los tres casos observados. La atención a consumidores, también se divide con el contexto, en el caso de APILayer y Nubentos, con los proveedores de las APIs. Se usan otras plataformas para ofrecer más recursos que completen la infraestructura, como GitHub, las nubes de diversos proveedores para alojar las APIs de los proveedores, o una herramienta para los contratos que se suscriben en la dimensión normas. APILayer es el único EAW que explícitamente enuncia una relación con los gobiernos de los países, al indicar que en el pago a los proveedores se descontarán las tasas impositivas correspondiente a los países de origen.

(x) El EAW hace visible su fortaleza. Cada EAW se presenta con una fortaleza como estrategia para captar más consumidores y proveedores de APIs, o distinguirse de otros EAW. La fortaleza de RapidAPI está en los productos que ofrece y la cantidad de APIs,

consumidores y proveedores que forman parte del ecosistema; APILayer, indica tres aspectos para ser elegido, ofrecer solo APIs curadas, inicio gratis, tiempo de actividad estricto (SLA bajo demanda disponible) y Nubentos, además de anunciar que ofrece el mayor catálogo del mundo de APIs para Salud digital, simplifica el proceso de integración a consumidores a la vez que les permite ahorrar costos. En los tres casos, la fortaleza radica en algún rasgo técnico, y luego en otros aspectos.

Las relaciones e interacciones entre los actores de la dimensión social (gestor-consumidor, gestor-proveedor y consumidor-proveedor) son en cierta forma influenciadas o determinadas por dos esquemas, la forma en que se establezca de atención al consumidor y el modelo de monetización. Estos esquemas deben ser estudiados con mayor detalle a efectos de comprender mejor el modelo de interacción entre los actores mencionados.

**Amenazas de la validez.** La validez descriptiva (interna) en este estudio buscó garantizar la precisión de las descripciones recogidas, para ello se diseñó de un formulario con los datos de las dimensiones del framework. La versión final del framework usado fue validado por un experto (cientista social del GIDIC – UTN FRER - Argentina), con el objeto específico de realizar un estudio exploratorio y descriptivo. En cuanto a la recogida de datos fue revisada y controlada varias veces por distintas autoras, minimizando el sesgo de interpretación. Sin embargo, podrían estar incompletas, si la información no estaba publicada en la plataforma o no pudo ser recabada por medios como las redes sociales. La validez externa, está dada en la medida que el framework puede representar todos los EAW. Hemos seleccionado solo tres EAW de un universo que sin dudas es mayor. Sin embargo, dado que tienen rasgos bien diferenciados, podemos asumir que es representativo. RapidAPI es líder y marcador de tendencia en la industria, y sus volúmenes de APIs y usuarios se cuentan por miles y millones. Mientras que Nubentos es un EAW de dominio específico (salud) y de reciente aparición, lo que lo posiciona en un mercado bastante más acotado. APILayer sería el EAW que se ubica en una posición un tanto intermedia, puesto que a pesar estar consolidado sus volúmenes son menores. En función de estas particularidades, consideramos que el framework capta la mayoría de los rasgos S-T de los EAW.

## CONCLUSIONES

En este trabajamos hemos expuesto un caso de estudio exploratorio y descriptivo sobre los EAW RapidAPI, APILayer y Nubentos. Hemos caracterizado a los mismos en términos S-T, identificado los componentes comunes de las dimensiones social, infraestructura y normas, y relaciones con el contexto. Al responder la pregunta que nos planteamos, hemos detectado también diversas diferencias, tales como los roles de los actores, capacidades de la infraestructura y las normas, distintos niveles de relación con el contexto y sus fortalezas.

La Economía API y la transformación digital están impulsando que los ámbitos y formas de desarrollo de software tradicionales sean atravesados por nuevas modalidades. Cada vez, es más frecuente que los desarrolladores de software desempeñen sus tareas usando en alguna medida estas plataformas, sumándose así a ecosistemas, como clientes (consumidores) o proveedores de APIs web. A la vez, atender y satisfacer las necesidades de quienes usan las plataformas, también es un desafío para el desarrollo de software, que debe amalgamar una diversidad de requisitos, componentes y recursos, que herramientas clásicas (lenguajes y frameworks de desarrollo, editores, depuradores, etc.) requieren y los métodos de desarrollo de software no contemplan claramente. Consideramos que la influencia e impacto en el trabajo de desarrollar software son razones válidas para abordar los ecosistemas de APIs web desde un enfoque S-T. El estudio continuará, en el análisis de la adaptación de los actores (dimensión social) a estos nuevos entornos y requerimientos de trabajo y los conjugan con el trabajo propio. A partir del análisis de los casos estudio entendemos que es necesario recabar más evidencia empírica que conteste las siguientes preguntas:

- (i) Correspondencia de roles y responsabilidades. En los EAW los roles de los desarrolladores consumidores y proveedores se reducen y generalizan (cliente, equipo, desarrollador) y no es claro quien hace que. ¿cómo se administra la separación de funciones de los proyectos de los consumidores y/o proveedores, entre las diferentes actividades que se realizan en el ecosistema? ¿aparecen nuevos roles y funciones?
- (ii) Dinámicas de interacción y/o comunicación. Las dinámicas de interacción y/o comunicación entre los actores del ecosistema son variadas y múltiples, ¿cómo influyen en la producción de los equipos de desarrollo de los consumidores y proveedores?

(iii) Los EAW ofrecen diversas herramientas y/o facilidades para el consumo o producción de las APIs (especificación OpenAPI, integración con otras herramientas y plataformas, generación de código en los lenguajes más populares, importación/exportación de especificaciones, SDK que cubren las principales plataformas y tecnologías desarrollo, etc.) para que lo desarrollado en la PE se traslade a los proyectos de los consumidores y/o proveedores. ¿Qué costo y esfuerzos implican estas tareas? ¿Cómo se planifica? ¿qué riesgos se corren?

(iv) Información de los EAW. Los EAW ofrecen diversas analíticas e información de consumo de APIs, de desempeño de los gateway, etc., ¿qué decisiones de los consumidores y proveedores requieren esta información?, ¿es suficiente?

(v) Los cambios en la EP. ¿Qué tipo de cambios de la plataforma impactan en los proyectos de los consumidores y proveedores? y ¿cómo se adaptan a estos cambios?

(vi) Gestión de la PE. Los gestores de la plataforma deben manejar diversas situaciones (incidentes, errores, actualización de la plataforma, soporte de ayuda y asistencia, etc.) ¿cómo es la organización gerencial, técnica y operativa de estos equipos?

(vii) La evolución y/o adaptación del ecosistema. ¿cómo se sucede?, ¿qué relaciones entre las dimensiones S-T entran en este tipo de tensiones?

#### **LISTA DE REFERENCIAS**

Apidays. Platformable (2022). API Landscape State of the Market 2022. En The API Landscape. Recuperado 16 de julio de 2022, de <https://apilandscape.apiscene.io/>

Badham, R., Clegg, C. & Wall, T. (2000). Socio-technical theory. En W. Karwowski (Ed.), Handbook of Ergonomics. John Wiley.

Baxter, G. & Sommerville, I. (2011). Socio-technical systems: From design methods to systems engineering. *Interacting with Computers*, 23(1), 4-17. <https://doi.org/10.1016/j.intcom.2010.07.003>

Beltagui, A., Rosli, A. & Candi, M. (2020). Exaptation in a digital innovation ecosystem: The disruptive impacts of 3D printing. *Research Policy*, 49(1), Article 103833. <https://doi.org/10.1016/j.respol.2019.103833>

Bogers, M., Sims, J. & West, J. (2019). What Is an Ecosystem? Incorporating 25 Years of Ecosystem Research. SSRN Electronic Journal. <https://doi.org/10.2139/ssrn.3437014>

- Bülthoff, F. & Maleshkova, M. (2014). RESTful or RESTless – Current State of Today’s Top Web APIs. *The Semantic Web: ESWC 2014 Satellite Events, Lecture Notes in Computer Science*, 8798, 64-74. [https://doi.org/10.1007/978-3-319-11955-7\\_6](https://doi.org/10.1007/978-3-319-11955-7_6)
- Casas, S., Cruz, D., Vidal, G. & Constanzo, M. (2021). Uses and applications of the OpenAPI/Swagger specification: a systematic mapping of the literature. 2021 40th International Conference of the Chilean Computer Science Society (SCCC). <https://doi.org/10.1109/sccc54552.2021.9650408>
- Costa, E., Soares, A. L. & de Sousa, J. P. (2020). Industrial business associations improving the internationalisation of SMEs with digital platforms: A design science research approach. *International Journal of Information Management*, 53, 102070. <https://doi.org/10.1016/j.ijinfomgt.2020.102070>
- Creswell, J. W. (2014). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. SAGE Publishing.
- Cummaudo, A., Vasa, R. & Grundy, J. (2019). What should I document? A preliminary systematic mapping study into API documentation knowledge. 2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM). <https://doi.org/10.1109/esem.2019.8870148>
- De, B. (2017). *API Management: An Architect’s Guide to Developing and Managing APIs for Your Organization* (1st ed.). Apress.
- Elatlassi, R. & Narwankar, C. (2006). A Categorization of Socio-Technical Systems Approaches based on Context and Purpose. *Proceedings of the 60th Annual Meeting of the International Society for the Systems Sciences (ISSS)*, 1(1).
- Emery, F. E. & Trist, E. L. (1960). Socio-technical systems. En C. W. Churchman & M. Verhulst (Eds.), *Management sciences, models and techniques* (Vol. 2). Pergamon Press.
- Fabbri, M. S. (2020). Las técnicas de investigación: la observación. *Institutocienciashumanas.com*. Recuperado 13 de marzo de 2022, de <https://bit.ly/3FKTt7E>
- Fielding, R. T. (2000). *Architectural styles and the design of network-based software architectures* [Disertación doctoral]. University of California, Irvine. <https://bit.ly/3CbeaHJ>

- Kapoor, K., Ziaee Bigdeli, A., Dwivedi, Y. K., Schroeder, A., Beltagui, A. & Baines, T. (2021). A socio-technical view of platform ecosystems: Systematic review and research agenda. *Journal of Business Research*, 128, 94-108. <https://doi.org/10.1016/j.jbusres.2021.01.060>
- Kingdon, J. W. (1995). *Agendas, Alternatives, and Public Policies* (2.a ed.). HarperCollins College Publishers.
- Koci, R., Franch, X., Jovanovic, P. & Abello, A. (2019). Classification of Changes in API Evolution. 2019 IEEE 23rd International Enterprise Distributed Object Computing Conference (EDOC), 243-249. <https://doi.org/10.1109/edoc.2019.00037>
- Kopecký, J., Fremantle, P. & Boakes, R. (2014). A history and future of Web APIs. *Information Technology*, 56(3), 90-97. <https://doi.org/10.1515/itit-2013-1035>
- Lyytinen, K. & Newman, M. (2008). Explaining information systems change: a punctuated socio-technical change model. *European Journal of Information Systems*, 17(6), 589-613. <https://doi.org/10.1057/ejis.2008.50>
- Maleshkova, M., Pedrinaci, C. & Domingue, J. (2010). Investigating Web APIs on the World Wide Web. 2010 Eighth IEEE European Conference on Web Services, 107-114. <https://doi.org/10.1109/ecows.2010.9>
- Mathijssen, M., Overeem, M. & Jansen, S. (2020). Identification of Practices and Capabilities in API Management: A Systematic Literature Review. *arXiv: Software Engineering*. <https://doi.org/10.48550/arXiv.2006.10481>
- 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. <https://doi.org/10.1016/j.infsof.2017.12.010>
- Mumford, E. (2006). The story of socio-technical design: reflections on its successes, failures and potential. *Information Systems Journal*, 16(4), 317-342. <https://doi.org/10.1111/j.1365-2575.2006.00221.x>
- Nybom, K., Ashraf, A. & Porres, I. (2018). A Systematic Mapping Study on API Documentation Generation Approaches. 2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA). <https://doi.org/10.1109/seaa.2018.00081>

- Ofoeda, J., Boateng, R. & Effah, J. (2019). Application Programming Interface (API) Research. *International Journal of Enterprise Information Systems*, 15(3), 76-95. <https://doi.org/10.4018/ijeis.2019070105>
- Renzel, D., Schlebusch, P. & Klamma, R. (2012). Today's Top "RESTful" Services and Why They Are Not RESTful. *Web Information Systems Engineering, Lecture Notes in Computer Science*, 7651, 354-367. [https://doi.org/10.1007/978-3-642-35063-4\\_26](https://doi.org/10.1007/978-3-642-35063-4_26)
- Robson, C. (2002). *Real World Research: A Resource for Social Scientists and Practitioner-Researchers* (2.a ed.). Wiley-Blackwell.
- Runeson, P., Host, M., Rainer, A. & Regnell, B. (2012). *Case Study Research in Software Engineering: Guidelines and Examples*. Wiley.
- 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. <https://doi.org/10.1109/mic.2016.74>