

UAICase: Arquitectura diseñada para dar una solución ubicua colaborativa

Flores Fanelli, Francisco Andrés; Battaglia, Nicolas; Neil, Carlos

Universidad Abierta Interamericana

Facultad de Tecnología Informática

CAETI – Centro de Altos Estudios en Tecnología Informática

Buenos Aires, Argentina

francisco.floresfanelli@gmail.com; nicolas.battaglia@uai.edu.ar;

carlos.neil@uai.edu.ar

Resumen

En este trabajo se muestra una arquitectura de software pensada para soluciones ubicuas colaborativas, utilizando como ejemplo el software UAICase.

Este se encuentra integrado múltiples módulos los cuales trabajan en conjunto para lograr la capacidad de una aplicación ubicua, es decir, una aplicación a la cual se puede acceder desde múltiples dispositivos, contando con la capacidad de realizar diagramas UML de modo colaborativo.

Estos diagramas pueden ser evaluados por el docente a cargo de la materia en tiempo real e incluso, ser autoevaluados por los alumnos. Dando la posibilidad de una comunicación fluida entre estos, acelerando y ayudando al proceso de aprendizaje.

Por lo tanto, gracias a estas características, se elimina la necesidad de que el contacto con el profesor sea meramente presencial y solo dentro de los horarios de cursada.

Palabras Clave: Ubicuidad, Colaboración, Servicios, Aprendizaje, WebSockets, Microservicios.

Introducción

Actualmente la necesidad que se tienen de obtener información, desde donde sea y como sea, tanto forma física o digital nos ha llevado a desarrollar tecnologías que nos permita satisfacer este requerimiento.

Gracias a la disponibilidad de las interfaces electrónicas que se encuentran presentes en

nuestra vida diaria, ya sean celulares, computadoras, relojes inteligentes, etc., es posible acceder a esta prácticamente al instante.

Por lo tanto, los dispositivos digitales al estar interconectados, cuentan con la posibilidad de comunicar a múltiples personas, y que estas puedan acceder a la misma fuente o información de manera concurrente.

Teniendo esta habilidad, estos pueden dar lugar a la generación de un entorno colaborativo, en el cual todos pueden compartir sus conocimientos, así como aprender de otros mediante la ayuda mutua. Planteando esta situación, las soluciones informáticas y sistemas de información que se utilizaran deberán plantearse en un contexto diseñado para tal fin y apoyarse en una arquitectura que permita acceder a la información de forma ubicua.

Tal como se indica [1] las soluciones software que sean diseñadas para ser utilizadas de forma colaborativa y requieran acceso ubicuo (UCSCL) [4] o, de forma más específica, aquellas que sean destinadas para utilizar en alguna área de conocimiento específico, como por ejemplo el modelado de software en la educación (UCASE-CL) [5].

CONAIISI 2018

6to Congreso Nacional de Ingeniería
Informática - Sistema de Información

Este no será solo acceso concurrente y en tiempo real para compartir recursos, sino que debe ser un diseño el cual permita que la solución sea independiente de la plataforma para que pueda accederse de forma ubicua.

Esto aplicaría tanto como para una computadora personal como un dispositivo móvil, permitiendo que los usuarios de ambas tecnologías puedan colaborar en un mismo proceso de forma simultánea si así lo desean.

Ubicuidad

El concepto de ubicuidad se refiere a la capacidad de estar en todo momento y en todo lugar.

Para la computación ubicua, término acuñado por Mark Weiser [2], la cual comprende a todos aquellos servicios que permitan al usuario interactuar con los sistemas digitales mediante interfaces naturales.

A esta se la puede definir con los siguientes cuatro principios:

- El propósito de un dispositivo informático es ayudarnos en hacer algo.
- El mejor dispositivo es aquel el que pasa desapercibido.
- Los dispositivos deben extender tu inconsciente.
- La tecnología tiene que aportar calma y bienestar.

Tecnología Colaborativa

Las tecnologías colaborativas se caracterizan por adquirir, distribuir y utilizar la información que generan a partir de la transformación de datos.

Estas permiten el intercambio de información necesaria entre distintas organizaciones, personas o departamentos.

La necesidad de compartir este saber va más allá de las necesidades individuales, aprovechando las facilidades que brinda esta tecnología.

Esta genera ciertos beneficios:

- Facilitan la comunicación.
- Se pueden compartir conocimientos, resolver dudas, realizar tareas de forma virtual.
- Ayudan a desarrollar el pensamiento crítico.
- Permiten una mayor flexibilidad y capacidad de adaptación.
- Cada integrante/participante de la actividad será actor principal así también como espectador.
- Un ambiente que fomenta el intercambio de comentarios o ideas, generando una realimentación inmediata entre los participantes.

UAICase fue diseñado para utilizar **WebSockets** en el aspecto colaborativo, estas tecnologías proporcionan un canal de comunicación bidireccional, o full-duplex, sobre un único socket TCP, manteniendo esta conexión por tiempo indeterminado.

Sistemas Orientados a Servicios

La arquitectura orientada a servicios o SOA, es un marco de trabajo conceptual que establece una estructura de diseño para la integración de aplicaciones o servicios.

También se podría decir que es un concepto que define la utilización de servicio para dar soporte a ciertos requisitos del negocio.

Permitiendo crear sistemas altamente escalables los cuales pueden impulsar el rendimiento, que al mismo tiempo pueden ayudar a reducir los costos, así mejorando la flexibilidad de los procesos de negocio.

Las razones del requerimiento de una arquitectura SOA fueron:

- Integración entre aplicaciones y plataformas es complicada.
- Existencia de sistemas heterogéneos, es decir, diferentes tecnologías.
- Existencia de múltiples soluciones de integración, independientes y ajenas unas de otras.

CONAIISI 2018

6to Congreso Nacional de Ingeniería
Informática - Sistema de Información

- Una arquitectura de desarrollo orientada a servicios.
- Basada en un “bus común estándar de mensajería”.
- Debe ser estándar entre todas las plataformas a utilizar.
- Interoperabilidad entre servicios.
- Basado en especificación estándar a nivel de protocolos.

SOA hace hincapié en que los sistemas se construyen basándose en un conjunto de servicios autónomos. Un “servicio” es simplemente un programa con el cual uno interactúa con éste mediante mensajes, por lo tanto, un conjunto de servicios se lo puede llamar sistema.

Actualmente existen dos tendencias de arquitectura en cuanto a servicios Web:

- SOAP (Simple Object Access Protocol), intercambio por medio de datos XML.
- REST (Representational State Transfer), servicios RESTful.

Los ambientes ubicuos también tienen su representación en Arquitecturas Orientadas a Servicios.

Montejano y Testa explican en su trabajo [3], que en un modelo en donde la coordinación y administración de dispositivos ubicuos en ambientes pervasivos se pueden atacar utilizando esta arquitectura.

UAICase SOA

La arquitectura SOA puede ser implementada por la arquitectura propuestas por REST [12], al igual que los entornos ubicuos posee de forma natural un acceso orientado a recursos URI (Identificador de recursos uniforme). Gracias a esto se puede implementar de manera sencilla en forma de servicios y

aplicaciones que pueden ser utilizadas por medio de cualquier dispositivo con la capacidad de utilizar un cliente que entienda la semántica propuesta por HTTP [1].

Al diseñar UAICase se decidió utilizar esta opción de SOA, ya que, al día de hoy una gran mayoría de los dispositivos de uso diario, ya sean computadoras, celulares, tablets cuentan con un explorador web por default, y todos son compatibles con el protocolo HTTP.

La aplicación utiliza un modelo distribuido con una arquitectura de Cliente-Servidor, de esta manera todos los servicios se desarrollaron para estar alojados en un servidor para ser consumidos por los clientes WEB.

Arquitectura Ubicua Colaborativa UAICase

Como antes ya fue mencionado, la arquitectura utilizada para diseñar UAICase fue la de Cliente-Servidor, por lo tanto, cada usuario (Cliente) podrá acceder a la aplicación y comunicarse con el servidor desde su explorador web, al estar basada en un estándar (HTTP) cuenta con compatibilidad con la mayoría de los exploradores del mercado.

Los distintos módulos del sistema están siendo alojados utilizando Bluemix [6] y la capa subyacente para computación en la nube (Cloud Computing) denominado Softlayer [7] provista por IBM en el marco del acuerdo con UAI, obtenido para implementar el proyecto UAICase utilizando esta infraestructura. Esto lo podemos comprobar por medio del diagrama de la **Figura 1** el cual representa la arquitectura completa del sistema

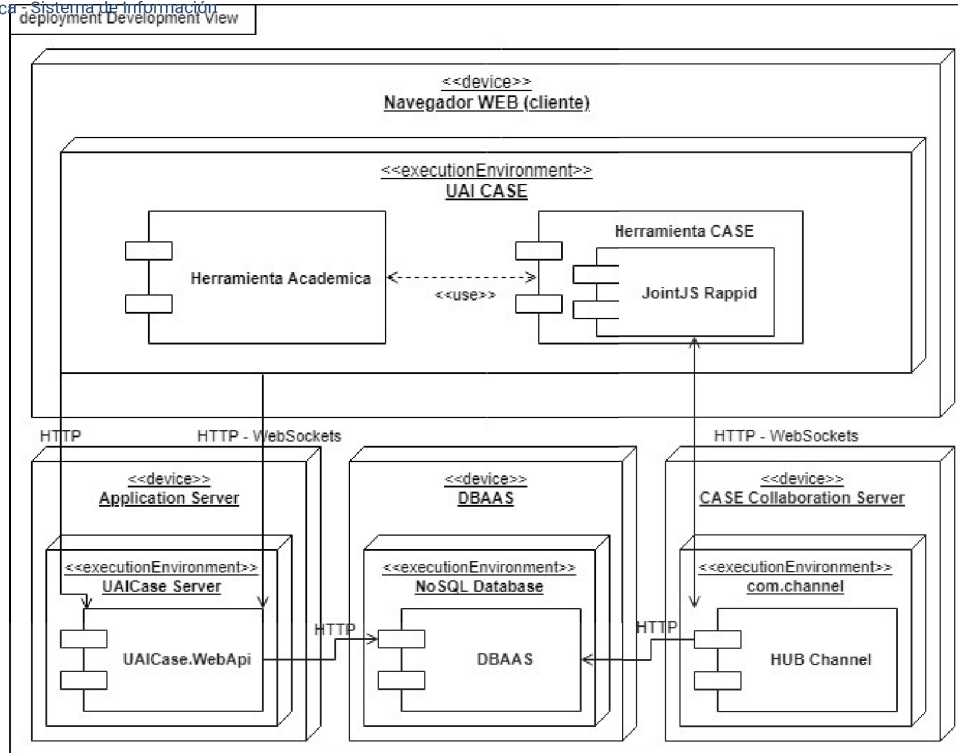


Fig 1. Modelo de componentes

- **Navegador WEB**, este se encargará de mostrar la interfaz de usuario en cada dispositivo, es decir lo que ve el usuario y percibe cómo el sistema en sí. Este módulo completo comprende al Cliente.
- **Application Server (API)**, este tiene como objetivo el manejo de la mayoría de las funciones del sistema, ya sea el manejo de permisos, acciones de CRUD, además se encarga de manejar los pedidos que vienen desde el lado del cliente. Estos llegan a modo de Request HTTP (solicitud/mensajes) al servidor y este regresa una respuesta.
- **DBAAS**, encargado del almacenamiento de datos del sistema, realizados en una base de datos no NoSQL, es decir orientada a documentos. Se decidió utilizar este tipo de BD para tener en cuenta más que nada la velocidad de respuesta ya que es necesario una respuesta inmediata al realizar un sistema de acceso en tiempo real. Un servicio de BD NoSql alternativo a DBASS, es mongoDB ya que con muy pocas configuraciones se lo puede tener funcionando en cualquier servidor, ya que al ser un servicio liviano no necesita de grandes requerimientos.
- **CASE Collaboration Server**, este módulo se encarga de la sincronización de los modelos UML, mientras los múltiples usuarios están conectados. Por lo tanto, este es uno de los módulos más importante ya que es el que se encarga de compartir en tiempo real los modelos entre los usuarios (**ubicuidad**), generando ese ambiente buscado de **colaboración** entre estos.

Seguridad en sistemas ubicuos

Dos aspectos bastante importantes deben tomarse en cuenta al momento de implementar el acceso a sistemas ubicuos, son la autenticación y la autorización.

Una de las características más importantes de la arquitectura REST es que no cuenta con un estado, esto quiere decir que no guarda las sesiones de los usuarios que se conectaron, sino que en cada solicitud que recibe de los clientes esta información viene con esta. Por lo tanto, cada vez que un usuario quiera acceder a algún recurso de la aplicación (URI) incluirá dentro del mensaje sus datos de acceso, ya sea una clave como un dato cifrado.

Autenticación: Esta se refiere a comprobar si el usuario existe dentro del sistema, es decir si cuenta con usuario y contraseña, y si cuenta con los permisos necesarios. En UAICase además de realizar la acción de validar efectivamente que el usuario cuenta con credenciales, se le otorga a cada usuario una componente léxico o TOKEN generada por el servidor con tiempo de expiración. Este servirá para futuros acceso.

Autorización: Este aspecto está ligado a validar si el usuario tiene los permisos necesarios para realizar una acción. Esto se realiza por cada petición que llega del cliente al servidor.

Para lograr implementar estos conceptos en UAICase se utiliza un estándar abierto con el nombre de JWT (Json Web Token) [9], este contiene toda la información requerida de un usuario sin la necesidad de volver a requerir una nueva petición a la BD. Es decir, que al ser compacta y autocontenida, puede almacenar todo lo requerido para agregarlo en cada petición (header request) al sistema, siendo suficiente para que esta puede aprobar como rechazarla, esto significa que puede dar o no acceso a la petición del usuario dependiendo lo que este contenga y como sea validado en la API [11].

Trabajo colaborativo entre los usuarios

Para lograr crear realmente un ambiente en el que todos los usuarios puedan conectarse, sin importar su ubicación, permitiendo una interacción entre estos en tiempo real, es decir, que todos puedan ver lo que está pasando y lo que están haciendo los demás usuarios sin la necesidad de esperar una respuesta.

Por lo tanto, se decidió la utilización de SignalR [8], el cual es framework/librería que proporciona websockets para la comunicación entre clientes/servidor. Para esto se creó un punto de acceso común (HUB) al cual los clientes (Exploradores WEB) se conectan, y mediante estas recibirán los datos relacionados con el entorno en donde se está realizando la actividad.

Entornos colaborativos dentro de UAICase

La aplicación cuenta con varios espacios lo cuales pueden colaborar a generar un entorno adecuado para el trabajo colaborativo, ya sea entre alumnos como con los docentes.

- **Espacios de trabajo para grupos de alumnos:** La herramienta dispone de un listado de “Grupos Activos”, esto le permite al docente saber quiénes conforman sus materias, todas sus actividades dentro de la aplicación, además de conocer los documentos que modificó.

Esto permite llevar un control más personalizado de cada alumno por parte del docente, ya que al saber que actividades ha realizado puede hacer hincapié en las que no ha realizado o en las que debe hacer para poder ser evaluado.

Dentro de este “módulo” se puede además iniciar conversaciones tanto los docentes como los alumnos en la

CONAIISI 2018

6to Congreso Nacional de Ingeniería
Informática - Sistema de Información

sección de chat ayudando a mantener una comunicación más fluida.

- **Espacio de trabajo para grupos docentes:** Al igual que los alumnos, los docentes cuentan con un espacio para interactuar, realizar actividades de control y llevar un seguimiento de cada una de las materias que le corresponde.
- **Espacio de trabajo para un curso:** Sección en la que el alumno verá en qué cursos él se encuentra activo, con cronogramas de cursada, además de tener acceso a todo el material que el docente provea, y pueda estar pendiente de las actividades que puedan realizarse con la herramienta colaborativa.
- **Sesiones de Chat:** Esta particular herramienta contribuye a la comunicación entre el alumno y el profesor, y se encuentra disponible en todos los entornos anteriormente mencionados.
- **Herramienta Case:** Diseñada para llevar la principal actividad de la herramienta. Diseñada para el ambiente Online. Permitiendo el modelado UML, donde se permite el trabajo colaborativo, ya sea entre alumnos como con el docente, posee un espacio de evaluación pedagógica.

Las actividades colaborativas con las que UAICase cuenta son:

- **Modelado UML colaborativo libre:** Cada usuario puede realizar diagramas de manera individual y modificarlo a placer.
- **Modelado UML colaborativo coordinado:** Múltiples usuarios conectados podrán modificar y visualizar diagramas en línea. Así como

también el docente podrá hacer las observaciones necesarias para guiar al grupo.

- Evaluación colaborativa de modelos UML.

Extensión con Microservicios

El paradigma de la arquitectura SOA, nos permite desarrollar sistemas distribuidos, por lo tanto, nos ofrece flexibilidad y facilidad en la integración de funciones en nuestros proyectos. Esto fue aplicado a la arquitectura de UAICase, permitiendo encapsular funciones específicas en componentes separados, es por eso que también se le podría aplicar el estilo de arquitectura de microservicios, a lo que estaríamos pensando en una arquitectura totalmente desacoplada, es decir cada servicio serio independiente del otro.

Podemos llegar a hacer una aproximación de un modelo orientado a microservicios respetando la arquitectura actual de componentes, **Figura 2.**

En la arquitectura actual ya se encuentran separados en componentes varios elementos de la API, pero al pensarlos como microservicios independientes, se los puede separar aún más.

El módulo colaborativo podría tener la necesidad de manejar muchas conexiones concurrentes lo cual impediría que este funcione con fluidez, por lo que podríamos agregar un servicio de manejo de carga.

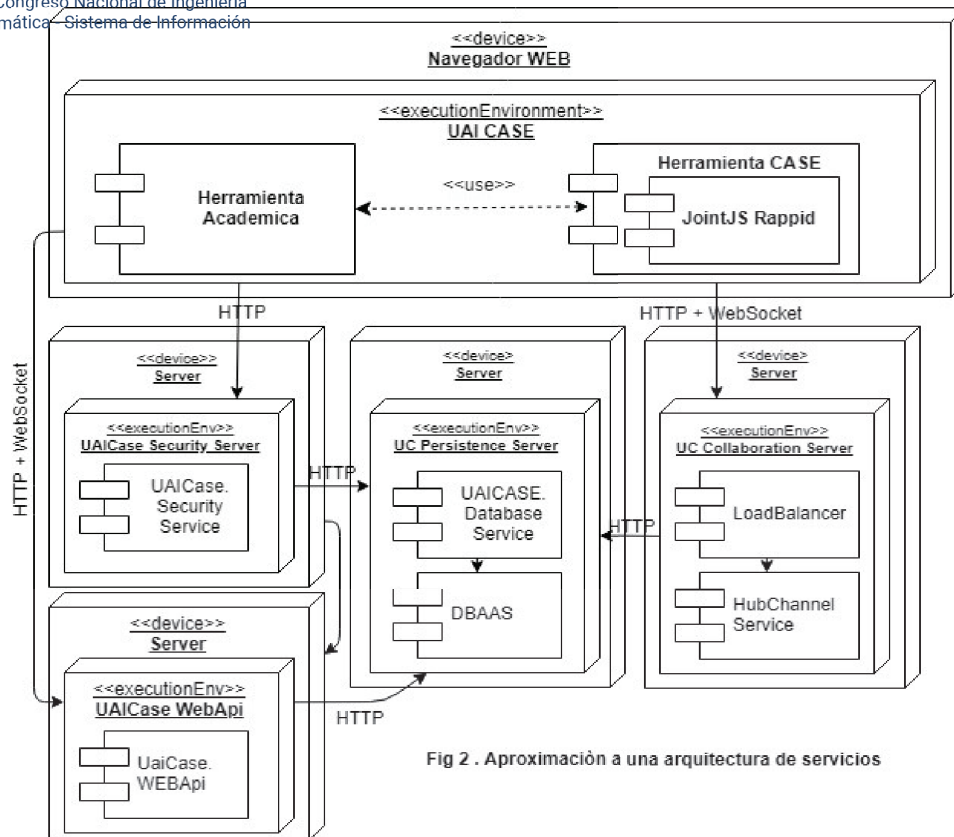


Fig 2 . Aproximación a una arquitectura de servicios

No estaríamos utilizando el HUB de conexiones como un solo módulo sino como ahora como micro servicios, permitiendo que el manejador de carga pueda validar muchas más conexiones en paralelo.

Otra buena práctica sería dividir la sección de seguridad de la Api en otro servicio separado, de esta forma este componente solo se encargaría de las tareas pertenecientes al manejo de alumnos, cursos y profesores.

Y por último se debería poder manejar el servicio de persistencia de manera independiente, permitiendo que pueda existir mas de un servicio de base datos.

Extendiendo esta funcionalidad, podría existir un servicio que se encargue de la persistencia relativa a errores, otra a datos, y otro a diagramas, estos trabajando de forma totalmente independiente, pero al mismo tiempo comunicados entre sí.

Como todos los servicios trabajan de manera independiente, no es necesario que se conozcan entre sí, solo necesitan conocer sus interfaces en caso de requerir de otro.

Conclusión y Futuros Proyectos

Como hemos visto, y probablemente muchos han experimentado, es que en entornos en los cuales uno puede opinar sin problemas, pero al mismo tiempo aceptar las opiniones de los demás, el conocimiento colectivo se encuentra enriquecido, logrando que cada individuo pueda llevarse algo más que solo una idea, sino un aprendizaje. Además, gracias a el avance de las tecnologías actuales, la cual ha logrado insertarse dentro de nuestra vida diaria, y hasta incluso pasar desapercibida, han logrado mantenernos conectados con todo y todos, propiciando estos ambientes colaborativos.

CONAIISI 2018

6to Congreso Nacional de Ingeniería
Informática - Sistema de Información

UCase pretende lograr generar este ambiente al hacer disponible esta conexión entre los alumnos y los profesores, haciéndola más fácil de lograr por medio de los entornos propuestos, en los que no necesariamente tienen que ser el aula.

Permitiendo que incluso sea conectado desde el celular, no solo con la posibilidad de ver los diagramas sino también en la posibilidad de modificarlos, sabiendo que el docente podrá ver su actividad y podrá evaluar, al mismo tiempo que podrá comunicarse con este para resolver dudas.

Pero no solo tendrá una comunicación directa con el docente, sino que también con sus demás compañeros, ayudándose mutuamente.

Esto permite propiciar un ambiente colectivo de conocimientos, participando y colaborando entre todos, haciendo más ameno el proceso de evaluación tanto para el docente como para el o los alumnos.

Como hemos visto que en la arquitectura SOA utiliza estándares ya probados y con funcionalidad comprobada, este se puede adaptar a prácticamente cualquier dispositivo.

En resumen, mientras tengamos un cliente el cual puede interpretar solicitudes HTTP podremos implementar esta arquitectura.

Dados los temas expuestos es trabajo se propone abordar en un futuro:

- Posible desarrollo de aplicaciones móviles propietarias. Esto sería el poder agregar al modelo general un módulo de móviles, ya sea android, ios, etc. Esto no solo se aplicaría a celulares sino también tablets.

Referencias

- [1] Nicolas Battaglia. Integración de una Herramienta CASE en un Entorno Académico Colaborativo para la Enseñanza de Ingeniería de Software.
- [3] Germán Montejano; Oscar Testa; Rubén Pizarro; Darío Segovia; Oscar Dieste; Efrain

Fonseca. Aplicación de principios SOA para la composición de servicios en ambientes ubicuos.

[2] M Weiser. ACM Conference on Computer Science, 1994 - archive.teachfind.com

[4] Coto, M., Collazos, C. A., & Rivera, S. M. (2016). Modelo Colaborativo y Ubicuo para apoyar los procesos de enseñanza-aprendizaje a nivel Iberoamericano. Revista de Educación a Distancia, (48)

[5] UCASE - CL: aprendizaje colaborativo de la ingeniería de software en entornos virtuales ubicuos. Battaglia, Nicolás | Neil, Carlos | De Vincenzi, Marcelo | Martínez, Roxana | González, Dana

[6] IBM Bluemix - Infraestructura de cloud, servicios de plataforma, Watson y más soluciones PaaS. (2017). Ibm.com. Retrieved 8 February 2017, from <https://www.ibm.com/cloud-computing/bluemix/es>

[7] SoftLayer | Cloud Servers, Storage, Big Data, & More IAAS Solutions. (2017). Softlayer.com. Retrieved 9 February 2017, from <http://www.softlayer.com/es>

[8] signalr.net. (2017). Incredibly simple real-time web for .NET. [online] Available at: <http://signalr.net/> [Accessed 1 Feb. 2017].

[9] Neil C, De Vincenzi M, Battaglia. N, Martínez. R (2016). Herramientas Colaborativas Multiplataforma en la Enseñanza de la Ingeniería de Software.

[10] RedHat. ¿Qué son los microservicios? <https://www.redhat.com/es/topics/microservices>

[10b] OpenWebinars. Qué son los microservicios y ejemplos reales de uso. <https://openwebinars.net/blog/microservicios-que-son/>

[11] Que es una API y para qué sirve. ABC. <https://www.abc.es/tecnologia/consultorio/20150216/abci--201502132105.html>

[12] API REST. ¿Qué es y cuáles son sus ventajas? <https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos>