**Universidad de las Fuerzas Armadas - ESPE**

**Departamento**

Ciencias de la Computación

**Carrera**

Ingeniería En Tecnologías de la Información

**Tipo de trabajo**

Investigación arquitectura de software en microservicios

**Asignatura**

Ingeniería de Software II

**Grupo Nª 5**

**Autores:** Narcisa Nataly Pila Iza

Luis Alfredo Carpio Celi

Geovany Nelson Toaquiza Puco

Aline Maribel Carrasco Pinto

**Tutor:** Ing. Jorge Edison Lascano

**Fecha:** Quito 20 de noviembre de 2022

**SANGOLQUI, 2022**

Introducción

Dentro del campo de la ingeniería de software existe una serie de conceptos y roles necesarios para el diseño correcto de un programa, uno de estos es la arquitectura de software que se encarga de establecer, descubrir y desarrollar nuevos modelos y estandarizará la relación entre cada una de las capas de una aplicación, de forma tal que se establece una guía para del desarrollo, funciones y herramientas a usar para alcanzar la funcionalidad esperada por el usuario del software. En este trabajo se sintetiza la arquitectura basada en microservicios, que en la actualidad es la más popular y varias empresas orientadas a la tecnología emplean debido a su gran escalabilidad.

Objetivos

* Sintetizar los conceptos respecto a los componentes y funcionalidades las arquitecturas de software de dos sistemas comerciales.
* Establecer diferencias y similitudes entre las arquitecturas que se presentan en este trabajo.

Desarrollo

1. **Arquitectura de Software**

Se trata de un conjunto de principio que ayudan y proporciona los cimientos sobre los cuales se puede construir el software. Este fin se alcanza con el uso de modelos de sistemas, patrones y atracciones teóricas. Este es el paso previo para cualquier implementación a realizar. [[[1]](#footnote-1)]

1. **Arquitectura basada en microservicios**

“*Es un método de desarrollo de aplicaciones software que funciona como un conjunto de pequeños servicios que se ejecutan de manera independiente y autónoma, proporcionando una funcionalidad de negocio completa*”. [ [[2]](#footnote-2)]

En otras palabras la arquitectura de microservicios es la descomposición de capas como la intermedia, y así conseguir dividir el sistema en muchas aplicaciones pequeñas llamadas microservicios. Una ventaja del uso de los microservicios es la posibilidad de implementarlos en diferentes servidores o contenedores en el mismo servidor.

1. **Arquitectura monolítica**

Las aplicaciones monolíticas son más fáciles de implementar y probar En las primeras etapas del proyecto, las aplicaciones monolíticas pueden funcionar bien. Sin embargo, a medida que la demanda sigue aumentando, cada vez más personas se unen al equipo de desarrollo y la base de código se expande rápidamente. Lentamente, las aplicaciones individuales se vuelven cada vez más infladas, la capacidad de mantenimiento y la flexibilidad se reducen gradualmente y los costos de mantenimiento son cada vez más altos.

* 1. Componentes de la arquitectura

De forma general la arquitectura de microservicios se divide en:

* + 1. Cliente

Se trata del conjunto de mecanismo necesarios para interactuar con el usuario, un ejemplo son las interfaces de usuario web, móviles o de escritorio (front-ed) que se encarga de presentar la información de forma atractiva al usuario y recoger los datos generados por el usuario para su posterior manejo en el back-end.

* + 1. Integración

Este elemento es el encargado de gestionar las peticiones del cliente hacia los microservicio y definir los parámetros de la comunicación, además de diferenciar e identificar la procedencia de los datos y hacer los ajustes necesarios para que la aplicación y funcionalidades del sistema se puedan ejecutar en el dispositivo del emplea el cliente.

* + 1. Comunicación

En esta capa se hace uso de REST’s o API’s para llamar a cada uno de los servicios según se requiera por los clientes mediante las puertas de enlace de API, que reenvía la llamada a los servicios apropiados en el back-end.

* + 1. Microservicios

En esta área se define las funcionalidades del software separándolas una de otra, de manera tal que sean independientes y en el caso de un servicio fallara el resto de los servicios no se vea afectado, en esta capa se define funcionalidades, estructuras y los paradigmas con los cuales se manejaran los datos.

* + 1. Datos

En esta área se define las características de las tablas donde se almacenarán los datos recolectados bajo el formato establecido en el área de microservicios, idealmente cada microservicio cuenta con su propia base de datos para que los casos de problemas en el futuro datos de otros microservicios no se vean comprometidos. [ [[3]](#footnote-3)]

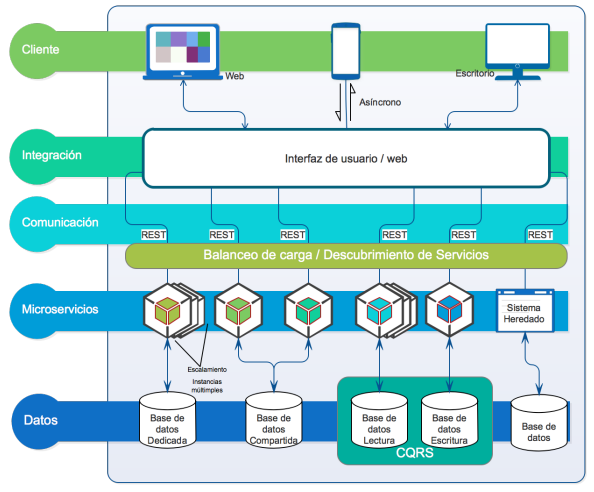


Figura 1. Esquema de genérico de una aplicación bajo la arquitectura de microservicios **Fuente:** [3]

1. **Arquitectura de Software de Instagram**

El diseño de esta aplicación está basado en la arquitectura de microservicios donde cada funcionalidad se vuelve un microservicio tal es el caso de los sistemas de “Me gusta”, subida de fotos, cometarios, notificaciones, etc. Cabe mencionar que en Instagram se hace usos de diferentes tecnologías y lenguajes con el fin de darle una escalabilidad y facilitar el mantenimiento del sistema. Un ejemplo de ello es el uso de Amazon s3 únicamente para el alojamiento de la base de datos de las imágenes.

* 1. Funcionalidad de cada elemento

Como ya se mencionó con anterioridad la arquitectura se divide en 5 elementos, en el casos de Instagram se hace una diferenciación entre el usuario móvil y web ya que en ambos entregan diferente tipo de datos. De ahí en adelante cada elemento funciona de la forma ya antes descrita.

* 1. Funcionalidad de conjunto

En conjunto cada elemento es necesario y trabajan a la vez, donde los clientes realizaran peticiones a través del API con cada interacción que tengan con el sistema, cada tipo de interacción con las diferentes funcionalidades de Instagram dándole un identificador para que la información vaya a donde sea requerida y ya dentro del microservicio sea trabajada de forma tal que se almacene en la base de datos y de ser el caso otros micro servicios usen el contenido de la base de datos para presentar nueva información al usuario. [ [[4]](#footnote-4)]

Diagrama

Descripción generada automáticamente

Figura 2. Bosquejo genérico de la arquitectura basada en microservicios de Instagram. **Fuente:** Elaborado por los autores en Draw.io

1. **Arquitectura de Software de la Asamblea Nacional del Ecuador**

Este sistema se lo implemento el 2017 con el fin de actualizar el sistema de que se tenía en ese entonces pasando de una arquitectura monolítica a un de microservicios.

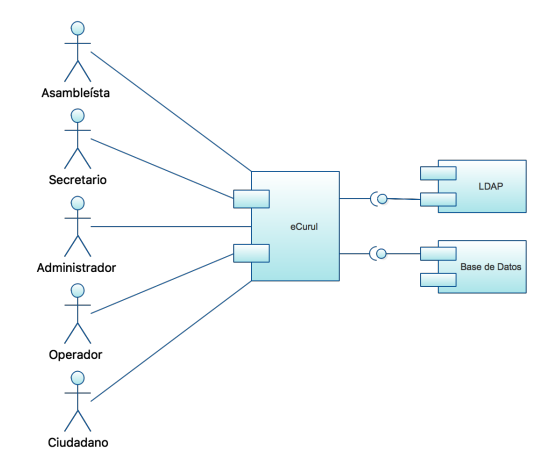


Figura 3. Bosquejo de la arquitectura genérico basado en microservicios de la Asamblea Nacional. **Fuente:** [ [[5]](#footnote-5)]

Para este caso se identifica varios clientes que interactúan con el sistema y la curul se presenta como la representación del microservicio que identificara al usuario y enviara la respuesta que recolecta a una base de datos para almacenar los dato de sesiones, órdenes del día (temas a tratar), registros de asistencia, quórum y votaciones de propuestas de ley; ademas de almacenar en un sistema directivo activo para obtener los datos (LDAP) de los usuarios que acceden al sistema. [5]

* 1. Funcionalidad de cada elemento

En cuanto a los elementos, en el área de datos se usa los motores de base de datos PostgreSQL 9.x ó MySQL 5.x para hacer uso de patrones: dedicados para alojar los datos exclusivos de cada microservicio, en la compartición, almacenamiento de los datos para uno o más microservicios. En cuanto a los microservicios las tecnologías a usar de acuerdo con el análisis que realice el área de desarrollo para cada caso y se sugiere el empleo de las tecnologías y lenguajes de mayor uso por parte del Área de Desarrollo de la CGTIC fueron java como plataforma y Node.js para la activación de los microservicios.

En la comunicación prevalece los servicios web REST, y para el caso de aplicaciones heredadas las que cada sistema defina como: Servicios web basados en SOAP, Invocación de métodos remotos (RMI) Conectividad abierta a base de datos ODBC/JDBC, entre otros. [ [[6]](#footnote-6)]

En cuanto a la integración los microservicios pueden ser integrados a través de la delegación de la secuenciación de los microservicios existentes a un tercer microservicio que tendrá la función de llamar al resto y coordinar una respuesta al usuario.

Cabe mencionar que esta arquitectura trabajo junto al sistema eCurul que resulta estar diseñado bajo una arquitectura monolítica que se divide en:

* + 1. Capa de servicios de autenticación

Contiene módulos que proporcionan los datos de usuarios y su validación para acceso al sistema.

* + 1. Capa de acceso a datos

Contiene módulos que almacenan los datos generados por los módulos de la capa de procesos legislativos.

* + 1. Capa de procesos legislativos

Contiene módulos para efectuar procesos legislativos.

* + 1. Conector local de directorio activo

El módulo se encarga de proporcionar acceso al directorio activo institucional para validar usuarios con acceso al sistema.

* + 1. Conector de base de datos

Este módulo proporciona acceso a la base de datos para el registro/consulta de datos generados por los módulos de procesos legislativo.

* + 1. Autenticación

Este módulo de administración de accesos de usuario al sistema. Interactúa con los datos de directorio activo.

* + 1. Administrador de quórum

Módulo de administración de asistencia/quórum de asambleístas. Interactúa con el módulo de autenticación para su validación, registrando los datos del proceso en la base de datos local.

* + 1. Administrador de votación

Módulo de administración de consignación de votos de asambleístas en sobre aprobación de una moción de proyecto de ley y los almacena en la base de datos local.

* + 1. Administrador de intervenciones

Módulo de administración de intervenciones sobre un tema activo para debate por parte de los asambleístas y los almacena en la base de datos local.

* 1. Funcionalidad de conjunto

En conjunto se trabaja bajo un modelo hexagonal con una composición en la funcionalidad con el fin de cumplir el principio de responsabilidad única, en otras palabras independencia. Por otra parte, los actores relacionados con el sistema eCurul, basado en una arquitectura monolítica, interactúan con las funcionalidades acordes a su rol, el sistema depende de un directorio activo (LDAP) para obtener los datos de los usuarios que acceden al sistema. La base de datos almacena los datos de sesiones, órdenes del día (temas a tratar), registros de asistencia, quórum y votaciones de propuestas de ley. [ [[7]](#footnote-7)]

1. **Comparación entre las arquitecturas presentadas**

|  |  |
| --- | --- |
| **Similitudes** | **Diferencias** |
| En ambos casos se hace uso de la arquitectura basada en microservicios para el diseño de software. | Si bien ambos cuentan con la misma arquitectura, la manera en que se comunican la API con el microservicio cambia entre sí. |
| En el caso de la Asamblea Nacional no existe un microservicio que luego del almacenaje en la base de datos requiera o solicite la información contenida. Todo lo contrario a Instagram que realimenta sus microservicios con bases datos obtenidas a partir de otros microservicios. |
| En ambas arquitecturas se hace una diferenciación de capas entre funcionalidades, tan solo que los microservicios se aíslan aún mas de forma tal que si una componente del sistema falla no influirá en el resto | Las arquitecturas monolíticas mezclan todas las capas bajo una misma tecnología, lo facilita el seguir construyen la aplicación y no es un problema cuando se prevé que el sistema no escale. En cambio las arquitecturas de microservicio permiten separar y distribuir las funcionalidades y permite que nuevas funciones se han agregadas, que es necesario sobre todo en sistemas que crecen y tienen alta generación de información. |

Conclusiones

* Las arquitecturas monolíticas soportan sistemas que no requieran mucha demanda y que no se prevea un gran crecimiento a futuro, su implementación es más barata pero que pude terminar teniendo un alto costo el manteamiento si el flujo de datos crece muy rápido. En cambio las arquitecturas de microservicios permiten una gran escalabilidad y es ideal para sistemas que registren rápidos cambios y crecimiento en el flujo de sus datos.
* No necesariamente un sistema debe contar con una única arquitectura, puede contar con mas de una siempre y cuando el manteamiento y desarrollo no se vea comprometido, en esos casos depende del diseñador medir los parámetros en juego para fijar la arquitectura a emplear.

Bibliografía

Daniel, L. H. (2017). *Universidad Técnica del Norte.* Obtenido de Core: https://core.ac.uk/download/pdf/200329168.pdf

*Decide Soluciones*. (03 de 09 de 2019). Obtenido de Decide Soluciones: https://decidesoluciones.es/arquitectura-de-microservicios/

Maida, E. G. (Diciembre de 2015). *Repositorio de Universidad Catolica de Argentina.* Obtenido de Repositorio de Universidad Catolica de Argentina: https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf

四种软件架构，看看你属于哪个层次. (n.d.). Retrieved November 19, 2022, from https://www.jianshu.com/p/e7b992a82dc0

Lien, A., Randem, L., Taralrud, H., & Edalati, M. (2022, June 14). OSN dashboard tool for sentiment analysis. Retrieved November 20, 2022, from https://arxiv.org/abs/2206.06935

Whitepapers. (n.d.). Retrieved November 19, 2022, from https://docs.aws.amazon.com/whitepapers/latest/microservices-on-aws/microservices-on-aws.html

1. 四种软件架构，看看你属于哪个层次. (n.d.). Retrieved November 19, 2022, from https://www.jianshu.com/p/e7b992a82dc0 [↑](#footnote-ref-1)
2. *Decide Soluciones*. (03 de 09 de 2019). Obtenido de Decide Soluciones: https://decidesoluciones.es/arquitectura-de-microservicios/ [↑](#footnote-ref-2)
3. Daniel, L. H. (2017). *Universidad Técnica del Norte.* Obtenido de Core: https://core.ac.uk/download/pdf/200329168.pdf [↑](#footnote-ref-3)
4. Diseño de Arquitectura para instagram | #microservicios. (2022, January 19). Retrieved November 19, 2022, from https://www.youtube.com/watch?v=4HlpOuycER0 [↑](#footnote-ref-4)
5. Daniel, L. H. (2017). *Universidad Técnica del Norte.* Obtenido de Core: https://core.ac.uk/download/pdf/200329168.pdf [↑](#footnote-ref-5)
6. Daniel, L. H. (2017). *Universidad Técnica del Norte.* Obtenido de Core: https://core.ac.uk/download/pdf/200329168.pdf [↑](#footnote-ref-6)
7. Daniel, L. H. (2017). *Universidad Técnica del Norte.* Obtenido de Core: https://core.ac.uk/download/pdf/200329168.pdf [↑](#footnote-ref-7)