

Architecture of a WIS

Group: L3-D01

Repository: <https://github.com/adrrf/acme-life-long-learning>

Student #1:

UVUS: manvazmar3

Name: Vázquez Martín, Manuel

Roles: Developer

Email: manvazmar3@alum.us.es

Student #2:

UVUS: angmunpri

Name: Muñoz Prieto, Ángel

Roles: Developer, Analyst

Email: angmunpri@alum.us.es

Student #3:

UVUS: adrromflo

Name: Romero Flores, Adrián

Roles: Developer, Manager

Email: adrromflo@alum.us.es

Student #4:

UVUS: jorromlim

Name: Romero Limón, Jorge

Roles: Developer

Email: jorromlim@alum.us.es

Student #5:

UVUS: alematcap

Name: Mateo Capilla, Alejandro

Roles: Developer

Email: alematcap@alum.us.es

Date: Sevilla a 16 de febrero de 2023

INTRODUCCIÓN	3
CONTENIDO	3
ARQUITECTURAS POR CAPAS	3
ARQUITECTURA MVC	4
ARQUITECTURA MICROSERVICIOS	5
CONCLUSIONES	6
BIBLIOGRAFÍA	7

Executive Summary:

En este documento se detalla nuestro conocimiento sobre la arquitectura de aplicaciones web. Vemos que hay varios acercamientos y tipos de arquitecturas que nos facilitan el desarrollo de la aplicación. Estas arquitecturas son: por capas, MVC y microservicios.

Introducción

Las aplicaciones web son una gran herramienta para los desarrolladores de software. Nos permiten que nuestras aplicaciones estén disponibles y accesibles desde internet. Pero a la hora de diseñar estas aplicaciones tenemos una gran variedad de arquitecturas para elegir. En este documento se detallarán las arquitecturas de las que el grupo de desarrollo tiene conocimiento.

Contenido

Arquitecturas por capas

Estas arquitecturas deciden dividir nuestra aplicación en varias capas cuyas responsabilidades están bien distinguidas. Cada una de estas capas ofrecerá un servicio a sus capas inferiores y superiores y recibirá el servicio de otra capa. El principal beneficio de esta arquitectura es el bajo acoplamiento y alta cohesión debido a la separación distinguida de responsabilidades de cada una de las capas. Pero, uno de los problemas que puede llegar a tener esta arquitectura es que suele ser poco eficiente.

Hay varias estructuras de capas que se usan en la industria, pero las más comunes son Cliente (*Front-End*)-Servidor (*Back-End*) y Presentación-Lógica de Negocio-Base de datos.

Otro beneficio es que una gran mayoría de las capas de la aplicación (por ejemplo las capas de Servidor o Lógica de Negocio y Base de datos) las podemos tener en servidores propios, ahorrando así que el cliente tenga que gastar recursos en hacer todo el trabajo interno y simplemente que el cliente interactúa con los datos que nuestras capas le envían a la capa de Cliente o Presentación.

Inclusive, ya hay frameworks que permite que el renderizado de los elementos de la capa de Cliente o Presentación lo haga nuestro servidor que solo enviaría páginas de HTML puro y trozos de código Javascript, ahorrando también en los recursos del cliente.

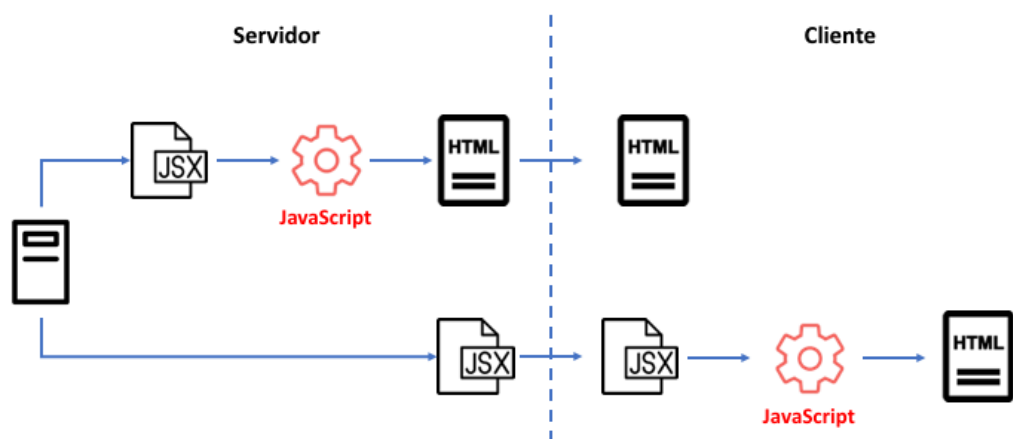


Ilustración 1: (Diez, Diagrama SSR 2018)

Arquitectura MVC

La arquitectura MVC se podría considerar un estilo de arquitectura por capas, pero esta tiene una distinta organización de los elementos. Sigue teniendo la misma filosofía que la arquitectura por capas, reutilización y separación de responsabilidades entre elementos. Estos elementos son: Modelo, que representa la información del sistema, Controlador, que gestiona los cambios y solicitudes de información, y Vista, que presenta al usuario los datos del Modelo.

La principal diferencia con el modelo por capas es como los elementos interactúan entre sí. En el modelo por capas como decíamos cada una de estas ofrece un servicio a sus capas inferiores y superiores, mientras que en esta arquitectura se puede decir que el Controlador manipula el Modelo, el Modelo es Representado por la Vista y la Vista es utilizada para llamar al Controlador.

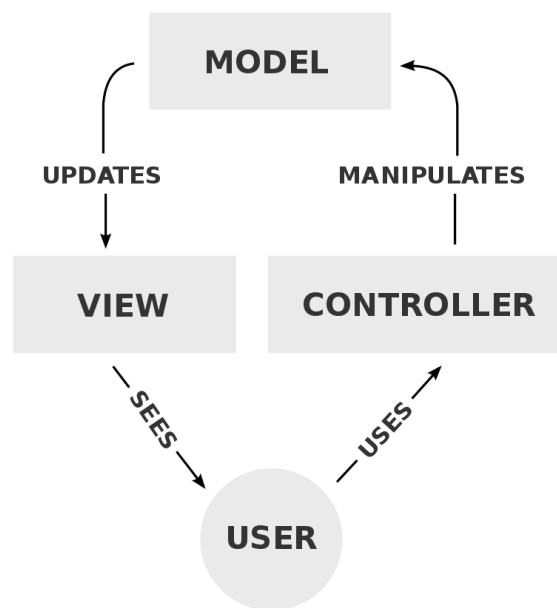


Ilustración 2: (Una típica colaboración entre los componentes de un MVC 2021).

Algunos de los beneficios de esta arquitectura, aparte de los mencionados en la arquitectura por capas, son que podemos presentar múltiples vistas utilizando el mismo modelo y controlador.

La principal desventaja de estos modelos, aparte de las mencionadas en la arquitectura por capas, es que suele requerir de una curva de aprendizaje es mucho más alta que con otras arquitecturas más simples.

Arquitectura Microservicios

Esta arquitectura tiene un enfoque más expansivo a diferencia que las otras dos arquitecturas mencionadas anteriormente. El principal enfoque es el mismo, tener una separación de responsabilidades bien distinguida. En este caso lo que hacemos es realizar una “aplicación” a pequeña escala que solo ofrezca un servicio atómico de nuestra aplicación web, obteniendo así múltiples aplicaciones (que llamaremos microservicios) cuya responsabilidad está bien definida y separada.

Una de las mayores ventajas de esta arquitectura es que podemos adaptar el microservicio a los requerimientos de cada uno de estos. Podemos utilizar distintas tecnologías en cada microservicio, dar más rendimiento a algunos microservicios, etc.

La gran desventaja de esta arquitectura es el manejo, gestión y despliegue de los microservicios, aparte de que la curva de aprendizaje puede ser aun más alta que la arquitectura MVC. Además, esta arquitectura solo tiene sentido si nuestra aplicación es muy grande, ya que no rentaría consumir muchos recursos por cada microservicio cuando una aplicación monolítica puede consumir muchos menos.

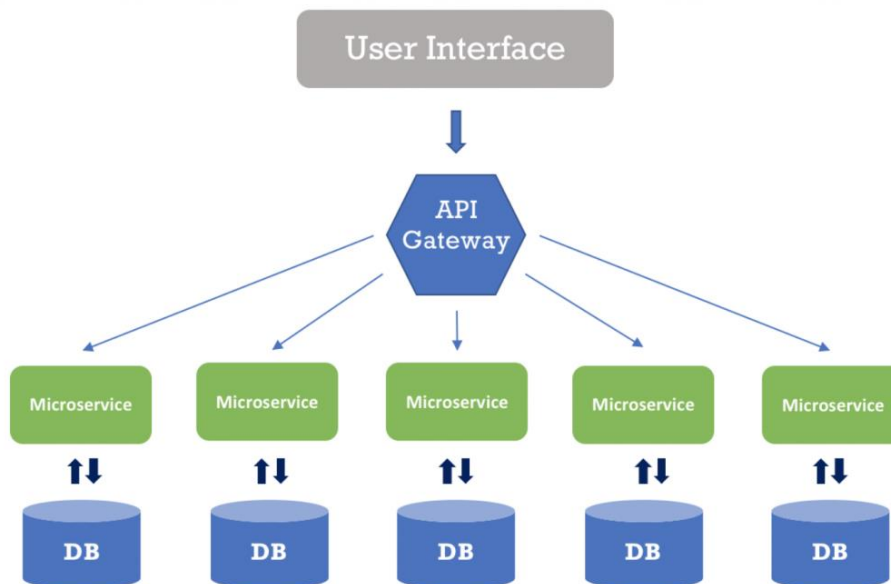


Ilustración 3: (Rodríguez, Arquitectura Microservicios 2019).

Conclusiones

Como resultado final, este documento refleja nuestro conocimiento acerca de varias arquitecturas que podemos aplicar al desarrollo de una aplicación web. También hemos visto algunas ventajas y desventajas de estas arquitecturas y en que casos es conveniente usar cada una de ellas.

Bibliografía

- Diez, B. (2018). Diagrama Ssr. Server Side Rendering I - Conceptos. Retrieved February 17, 2023, from <https://lemoncode.net/lemoncode-blog/2018/5/13/server-side-rendering-i-conceptos>.
- RegisFrey. (2021). Una típica colaboración entre los componentes de un Mvc. Modelo–vista–controlador. Retrieved February 17, 2023, from <https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador>.
- Rodríguez, M. (2019). Arquitectura Microservicios. CloudMasters. Retrieved February 17, 2023, from <https://www.cloudmasters.es/sabias-que-que-son-los-microservicios-y-por-que-son-el-futuro-en-la-arquitectura-de-software/>.