Universidad Católica del Uruguay

Arquitectura en capas - cliente y servidor

Análisis y Diseño de Aplicaciones

Franco Barlocco

Nicolás Lorenzo

Lucas Benítez

Entrega 27 de junio del 2024

Indice

Introducción

La arquitectura en capas representa un enfoque fundamental en el diseño de sistemas de software, utilizada extensamente para estructurar aplicaciones complejas y sistemas informáticos robustos. Al dividir un sistema en múltiples capas o niveles interrelacionados, cada una con responsabilidades bien definidas y funcionalidades específicas, esta metodología proporciona una serie de beneficios clave que van desde la modularidad y la escalabilidad hasta la claridad en la separación de preocupaciones.

Cada capa dentro de esta arquitectura se concibe como un componente independiente y coherente, diseñado para ejecutar una porción del procesamiento total del sistema. Esta estructura no solo facilita una mejor organización del código y una gestión más eficiente de recursos, sino que también promueve la reutilización de código y la facilidad en las actualizaciones y modificaciones, esenciales en entornos de desarrollo ágiles y adaptables.

Además, la arquitectura en capas promueve la utilización de buenas prácticas de diseño, como la separación de intereses y la cohesión, contribuyendo a un código más mantenible y comprensible. Cada capa puede enfocarse en tareas específicas, desde la presentación de la interfaz de usuario hasta la lógica de negocio y el acceso a datos, garantizando una estructura clara y eficiente que optimiza tanto el desarrollo inicial como las futuras expansiones del sistema.

En general, la arquitectura en capas no solo ofrece un marco robusto para el desarrollo de software, sino que también proporciona un enfoque sistemático para abordar la complejidad inherente a los sistemas informáticos modernos. Al dividir y conquistar, esta metodología permite a los equipos de desarrollo construir sistemas que sean tanto funcionales como adaptables, respondiendo de manera efectiva a las demandas cambiantes del mercado y las necesidades del usuario final.

Desarrollo

Hay tres componentes que trabajan en conjunto para estructurar y organizar un sistema de software en capas, donde cada capa tiene responsabilidades específicas y se comunica de manera definida con las capas adyacentes. Esta organización mejora la modularidad, la escalabilidad y la mantenibilidad del sistema, permitiendo un desarrollo más ordenado y estructurado. Estos componentes son las interfaces, los servicios y las funciones.

Interfaces

Una interfaz es un conjunto de definiciones de métodos y propiedades que proporciona una forma estandarizada de comunicación entre dos capas adyacentes. Las interfaces se utilizan para determinar cómo las capas interactúan entre sí y se comunican entre ellas. En general, una interfaz define un conjunto de operaciones que una capa puede realizar o solicitar a otra. Las operaciones de la interfaz pueden incluir la lectura y escritura de datos, la ejecución de operaciones o la obtención de información de estado. Las interfaces también pueden precisar propiedades que se utilizan para configurar o controlar el comportamiento de una capa.

Servicios

Los servicios en la arquitectura en capas sirven como unidades lógicas de funcionalidad que encapsulan la lógica de negocio o de aplicación de manera coherente y accesible a través de una interfaz bien definida. Los servicios encapsulan y centralizan la lógica compleja y específica de la aplicación o del negocio. Esto significa que todas las operaciones y decisiones críticas están contenidas dentro del servicio, facilitando así la comprensión y el mantenimiento del código.

Funciones

Son unidades lógicas de código que se utilizan para realizar una tarea específica dentro de una capa determinada. Las funciones se usan para implementar la lógica de negocio o de aplicación dentro de una capa, y suelen ser invocadas por otras partes del sistema a través de una interfaz definida. Las funciones son importantes en la arquitectura en capas porque permiten la modularización del código y la separación de responsabilidades. Cada función se encarga de realizar una tarea específica.

Arquitectura de dos capas

La arquitectura de dos capas, también conocida como Cliente-Servidor, es fundamental en sistemas informáticos donde un cliente solicita recursos y un servidor responde directamente a esas solicitudes utilizando sus propios recursos. En este modelo, el servidor no depende de otras aplicaciones para proporcionar el servicio requerido por el cliente.

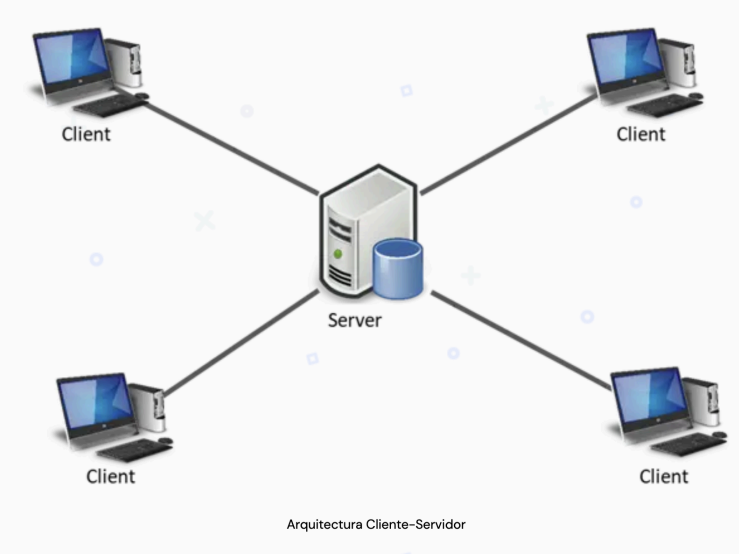
Capa de cliente: La capa de cliente está orientada principalmente a la interfaz de usuario y la interacción directa con el usuario final. Es responsable de presentar la información de manera comprensible y permitir que el usuario interactúe con la aplicación.

Capa de servidor: La capa de servidor se centra en la lógica de negocio y el manejo de datos. Es responsable de procesar las solicitudes recibidas desde los clientes, ejecutar la lógica de aplicación y devolver los resultados adecuados.

En una arquitectura Cliente-Servidor, múltiples clientes se conectan a un servidor para obtener los recursos necesarios para su funcionamiento. El cliente actúa principalmente como una interfaz para representar datos y realizar acciones que modifican el estado del servidor, mientras que el servidor lleva a cabo el procesamiento pesado y la gestión de datos.

La comunicación entre el cliente y el servidor se realiza típicamente a través de TCP/IP, lo que permite una conexión continua y bidireccional. Esta comunicación es esencial para que el cliente envíe solicitudes al servidor y reciba respuestas, asegurando así la interactividad y el intercambio de datos entre ambas partes.

En esta arquitectura, la disponibilidad del servidor es crucial para el funcionamiento del cliente, ya que sin el servidor en funcionamiento, el cliente no puede realizar sus operaciones. Del mismo modo, el servidor no tendría propósito sin clientes que lo utilicen. Esta interdependencia resalta la naturaleza distribuida de la arquitectura Cliente-Servidor, donde el servidor y los clientes pueden estar ubicados en diferentes equipos físicos o incluso en la misma máquina, comunicándose a través de una red, como Internet.



Arquitectura Cliente-Servidor

El Cliente y el servidor son desarrollados como aplicaciones separadas, lo que permite su desarrollo independiente en diferentes tecnologías, siempre y cuando respeten el protocolo de comunicación establecido. Esta separación facilita la centralización de la información y la lógica de negocio en el servidor, protegiendo así los datos y gestionando eficientemente los recursos.

El servidor, al ser el custodio de los datos y realizar operaciones complejas, generalmente se despliega en equipos con mayores recursos, capaces de atender simultáneamente a múltiples clientes. Por otro lado, el cliente se instala en dispositivos con recursos más limitados, ya que su función principal es mostrar información y facilitar la interacción del usuario con el sistema.

Arquitectura en tres capas

La arquitectura en tres capas, también conocida como arquitectura de tres niveles, es un modelo organizativo fundamental en el diseño de sistemas de software que permite la gestión eficiente de la complejidad y la modularidad. Este enfoque divide la aplicación en tres capas lógicas distintas, cada una con roles específicos y bien definidos, lo cual facilita la escalabilidad, el mantenimiento y la adaptabilidad del sistema.

Capa de Presentación: La primera capa, conocida como capa de presentación, se encarga de interactuar directamente con el usuario. Es aquí donde se desarrolla la interfaz gráfica de usuario (GUI) que permite la presentación de la información al usuario final de manera comprensible y la recepción de datos ingresados por el usuario. Esta capa se comunica exclusivamente con la capa intermedia o de negocio para solicitar datos y presentar resultados.

Capa de Negocio: La capa intermedia, o capa de negocio, contiene la lógica de aplicación o de negocio de la aplicación. Aquí residen las funciones y reglas de negocio que procesan los datos recibidos desde la capa de presentación o que son solicitados por ella. Esta capa es responsable de manejar las solicitudes del usuario, procesar la información según las reglas del negocio y enviar respuestas procesadas de vuelta a la capa de presentación. También se comunica con la capa de acceso a datos para almacenar o recuperar datos según sea necesario.

Capa de Acceso a Datos: La tercera capa, conocida como capa de acceso a datos, se encarga de manejar el almacenamiento y recuperación de datos desde y hacia la base de datos o cualquier otro sistema de almacenamiento persistente. Esta capa proporciona una abstracción entre la lógica de negocio y los detalles específicos de cómo se almacenan y recuperan los datos. Es la única capa que tiene acceso directo a los sistemas gestores de bases de datos y se comunica con la capa de negocio para cumplir con las solicitudes de almacenamiento o recuperación de datos.

Características y Beneficios: La arquitectura en tres capas ofrece varias ventajas significativas:

- **Separación de Responsabilidades:** Cada capa tiene un conjunto específico de responsabilidades, lo que facilita el mantenimiento, la depuración y la evolución del sistema.
- **Escalabilidad y Flexibilidad:** Permite escalar cada capa por separado según sea necesario, adaptándose a cambios en los requisitos o en la carga de trabajo.

- **Reutilización del Código:** La separación de la lógica de negocio de la interfaz de usuario permite reutilizar la lógica subyacente con diferentes interfaces de usuario, mejorando así la eficiencia del desarrollo.

Distribución Física: Aunque las tres capas pueden residir en un mismo servidor en aplicaciones menos complejas, en sistemas más robustos y distribuidos, cada capa puede estar ubicada en servidores diferentes para optimizar el rendimiento y la disponibilidad. Esto se logra manteniendo la comunicación entre las capas a través de protocolos bien definidos y estándares de comunicación.

En conclusión, la arquitectura en tres capas proporciona un marco sólido para el desarrollo de sistemas de software que facilita la gestión de la complejidad, mejora la mantenibilidad y permite la adaptación a los cambios del entorno tecnológico y de negocio, todo ello manteniendo un alto nivel de eficiencia y escalabilidad.

Arquitectura de n capas

Las aplicaciones N-Capas, son un enfoque arquitectónico en el desarrollo de software en el que una aplicación se divide en múltiples capas o niveles.
Cada capa tiene un conjunto específico de responsabilidades y se comunica con las capas adyacentes a través de interfaces bien definidas. El término "N" en "N-Capas" significa que puede haber múltiples capas, y el número específico de capas puede variar según los requisitos y la complejidad de la aplicación.

Características de las Aplicaciones N-Capas

Separación de Preocupaciones: Cada capa se enfoca en una preocupación específica, como la presentación de datos, la lógica de negocio o el acceso a datos. Esta separación facilita la comprensión y el mantenimiento del código.
Modularidad: Las capas se dividen en módulos o componentes independientes, lo que permite la reutilización y la fácil sustitución de partes del sistema.
Interfaz Bien Definida: Cada capa se comunica con las capas vecinas a través de interfaces bien definidas, lo que garantiza una comunicación ordenada y coherente.
Escalabilidad: Las aplicaciones N-Capas son escalables, lo que significa que se pueden agregar nuevas capas o módulos según sea necesario sin afectar otras partes del sistema.

Usabilidad de las Aplicaciones N-Capas

Las aplicaciones N-Capas son adecuadas para una variedad de escenarios, incluyendo:

Aplicaciones Empresariales: Para desarrollar aplicaciones empresariales complejas que requieren una separación clara de la presentación, la lógica de negocio y el acceso a datos.
Aplicaciones Web: En el desarrollo de aplicaciones web, donde la capa de presentación se encarga de la interfaz de usuario, la capa de lógica de negocio gestiona la funcionalidad y la capa de acceso a datos interactúa con la base de datos.
Sistemas de Gestión: En sistemas de gestión como sistemas de gestión de relaciones con el cliente (CRM), sistemas de gestión de recursos empresariales (ERP) y sistemas de gestión de contenido (CMS).
Funcionamiento de las Aplicaciones N-Capas

El funcionamiento de las aplicaciones N-Capas implica una clara separación de las responsabilidades de cada capa y una comunicación organizada entre ellas.

Capas fundamentales:

Capa de Presentación (Interfaz de Usuario): Esta capa se encarga de la presentación de la aplicación y la interacción con el usuario. Aquí se generan las páginas web, se gestionan las interacciones del usuario y se presentan los datos.
Capa de Lógica de Negocio: La capa de lógica de negocio contiene la lógica central de la aplicación. Aquí se procesan las solicitudes del usuario, se aplican reglas comerciales, se realizan cálculos y se toman decisiones.
Capa de Acceso a Datos: Esta capa se comunica con la base de datos subyacente. Se encarga de realizar operaciones de lectura y escritura en la base de datos, lo que incluye recuperar información, guardar datos y gestionar transacciones.

En una aplicación N-Capas, además de las capas básicas (presentación, lógica de negocio y acceso a datos), podríamos tener:

Capa de Servicios: Para la exposición de servicios RESTful, gestión de sesiones, etc.

Capa de Seguridad: Encargada de autenticación, autorización y gestión de permisos.

Capa de Integración: Para integraciones con sistemas externos o servicios de terceros.

Capa de Persistencia: Abstracción adicional sobre la capa de acceso a datos para manejar múltiples fuentes de datos.

Ventajas de la arquitectura en capas

Modularidad y Flexibilidad
Una de las principales ventajas de la arquitectura en capas es su modularidad. Cada capa funciona como una unidad independiente, lo que facilita la adición, modificación o eliminación de capas según sea necesario. Esto permite a los desarrolladores trabajar en una capa sin afectar las demás, lo que resulta en un desarrollo más rápido y eficiente.

Mantenibilidad Simplificada
La separación de responsabilidades en diferentes capas hace que el mantenimiento sea más sencillo. Si surge un problema en una capa específica, puedes abordarlo sin alterar las demás capas. Esto reduce el riesgo de introducir nuevos errores durante el proceso de corrección.

Mejora de la Escalabilidad
La arquitectura en capas también mejora la escalabilidad del sistema. Si la demanda aumenta, puedes escalar cada capa de manera independiente. Por ejemplo, si una red social experimenta un aumento en la cantidad de usuarios, se puede escalar la capa de acceso a datos para manejar la carga adicional sin afectar las otras capas.

Desventajas de arquitectura en capas

Estas arquitecturas que se dividen en capas, tienen algunas desventajas como:

Complejidad
aumenta la complejidad del sistema debido a la necesidad de interfaces y comunicación entre las capas.

Tiempo y esfuerzo de desarrollo
requiere más tiempo y esfuerzo de desarrollo debido a la necesidad de diseñar, desarrollar y probar cada capa por separado.

Consumo de recursos
aumenta el consumo de recursos del sistema, como el uso de memoria y CPU, debido a la necesidad de comunicación entre las capas.

Costos
eleva los costos del desarrollo debido a la necesidad de diseñar y desarrollar varias capas separadas.

Rendimiento
disminuye el rendimiento del sistema debido a la necesidad de comunicación entre las capas.

Dependencia entre capas
si alguna de las capas falla, puede afectar la funcionalidad completa de la aplicación.

DEMO

Desarrollamos una aplicación de Lista de Tareas (To-Do List) es una herramienta desarrollada para ayudar a los usuarios a gestionar sus actividades diarias de manera eficiente. Utilizando tecnologías modernas, esta aplicación ofrece una interfaz amigable y funcionalidades robustas que permiten agregar, editar, completar y eliminar tareas.

1. Capa de Presentación (Frontend)
Tecnologías Utilizadas: HTML, CSS, JavaScript.
Descripción: Esta capa es la interfaz de usuario que permite a los usuarios interactuar con la aplicación. Aquí, los usuarios pueden añadir, ver, editar y eliminar tareas.

2. Capa de Lógica de Negocio (Backend)
Tecnologías Utilizadas: Node.js, Express.js, Mongoose (para conectar y realizar operaciones en MongoDB), MongoDB.
Descripción: Esta capa maneja las operaciones de la lógica de negocio, como la creación, actualización, eliminación y obtención de tareas. Además, esta capa maneja la comunicación con la base de datos.

Solicitudes del Cliente:

- Cuando un usuario agrega una nueva tarea, el cliente envía una solicitud POST al servidor.
- Al solicitar la lista de tareas, el cliente envía una solicitud GET al servidor.
- Para actualizar una tarea, se envía una solicitud PUT.
- Para eliminar una tarea, se envía una solicitud DELETE.

Respuestas del Servidor:

- El servidor procesa las solicitudes del cliente, realiza las operaciones necesarias en la base de datos y devuelve la respuesta adecuada.
- Las respuestas incluyen datos en formato JSON que el cliente utiliza para actualizar la interfaz de usuario.

Conclusiones

La arquitectura en capas es esencial para el diseño de sistemas de software complejos, proporcionando beneficios como la modularidad, escalabilidad y mantenibilidad. Dividir un sistema en capas permite una mejor organización del código y facilita la adaptación a cambios y actualizaciones futuras.

En el desarrollo de aplicaciones basadas en arquitectura en capas, se enfatiza la importancia de las interfaces, servicios y funciones. Las interfaces definen cómo interactúan las capas, los servicios encapsulan la lógica de negocio y las funciones realizan tareas específicas dentro de cada capa, promoviendo así la separación de responsabilidades y la reutilización del código.

La arquitectura Cliente-Servidor, un tipo de arquitectura en dos capas, se destaca por la separación clara entre la interfaz de usuario (cliente) y la lógica de negocio y acceso a datos (servidor). Es adecuada para sistemas donde múltiples clientes interactúan con un servidor centralizado a través de una red, promoviendo la distribución de recursos y la eficiencia en el procesamiento.

La arquitectura en tres capas divide la aplicación en capas de presentación, negocio y acceso a datos. Esta estructura permite una gestión más eficiente de la complejidad, facilitando la escalabilidad y adaptación del sistema a medida que crecen las demandas del negocio y del usuario. La separación clara de responsabilidades entre las capas mejora la mantenibilidad y la reutilización del código.

Las aplicaciones N-Capas amplían el concepto a múltiples capas, permitiendo una mayor modularidad y flexibilidad. Además de las capas fundamentales, como la presentación, negocio y acceso a datos, pueden incluirse capas adicionales como servicios, seguridad, integración y persistencia, dependiendo de los requisitos específicos del sistema.

Finalmente se presentó una demostración práctica utilizando una base de datos MongoDB, donde se destacaron las tecnologías y componentes de cada capa (presentación, lógica de negocio y acceso a datos). Esta demostración ilustra cómo se aplican los conceptos teóricos en un entorno real de desarrollo de software.

Bibliografía

https://latamtech-sac.com/que-es-la-arquitectura-en-capas-descubre-sus-ventajas-y-ejemplos/

https://reactiveprogramming.io/blog/es/estilos-arquitectonicos/cliente-servidor

https://www.ibm.com/es-es/topics/three-tier-architecture