

Middleware

Introdução

Na aula de hoje, sobre Middleware, adentraremos em um campo essencial da arquitetura de software que serve como ponte para a comunicação e integração entre diferentes sistemas e aplicações. Middleware, é um conceito que, embora amplamente utilizado, permanece enigmático para muitos. Desvendar este tema nos permite compreender como softwares podem atuar discretamente, facilitando o fluxo de dados e informações de maneira eficaz entre plataformas distintas. Nesta aula, abordaremos desde a definição e importância do Middleware no cenário tecnológico atual até sua aplicabilidade em variados contextos, como sistemas legados, sensores e APIs. Por meio de exemplos e explicações teóricas, visamos oferecer uma compreensão sólida sobre como Middleware opera, sublinhando sua flexibilidade e papel crucial na manutenção e atualização de sistemas. Além disso, enfatizaremos a importância de abordar soluções já existentes antes de iniciar desenvolvimentos do zero, incentivando a inovação e eficiência no uso destas ferramentas vitais para o desenvolvimento de soluções tecnológicas. Preparados? Então, vamos mergulhar neste mundo intermediário que conecta e transforma a maneira como construímos e interagimos com a tecnologia ao nosso redor.

Definição de Middleware

Na primeira parte de nossa aula sobre Middleware, exploraremos o conceito fundamental deste termo que se tornou tão discutido, mas ainda pouco esclarecido no âmbito tecnológico.

Middleware, oriundo do termo em inglês que significa “o que está no meio”, desempenha um papel crucial na intermediação entre diferentes sistemas, atuando muitas vezes de forma invisível ao usuário. Sua principal característica é ser um

software que se posiciona entre outros softwares ou aplicações, facilitando a comunicação e o fluxo de dados entre eles. Esta capacidade de atuar como intermediário pode incluir funções diversas, como filtragem de dados, e até assumir o papel de um API Gateway, controlando o acesso entre as partes, quando se configura como o único ponto de passagem.

Através de exemplos didáticos, ilustraremos como o Middleware opera na prática, destacando sua flexibilidade em conectar usuários, aplicações web, servidores, e grupos de usuários a outros sistemas, como bancos de dados ou sistemas internos de uma empresa. Essa conexão pode ser crucial para a conversão de dados entre sistemas com diferentes requisitos ou padrões, exemplificado no contexto financeiro, onde dados externos precisam ser adaptados e validados conforme normativas específicas de cada país.

Além disso, o Middleware se mostra essencial na manutenção e atualização de sistemas legados, que, por suas limitações, não conseguem se comunicar diretamente com novas aplicações ou tecnologias. Ao invés de reconstruir sistemas antigos, fundamentais para o funcionamento de uma empresa, o Middleware oferece uma solução eficaz, agindo como um adaptador que permite a integração e comunicação entre diferentes gerações de tecnologia.

Neste módulo, também serão compartilhadas experiências de mercado, destacando a importância do Middleware no cenário atual da tecnologia da informação. Ao final desta parte da aula, espera-se que os alunos tenham uma compreensão sólida do conceito de Middleware, suas funções e aplicabilidades, despertando a curiosidade e o interesse em explorar mais sobre como trabalhar com Middleware em aplicações práticas. No decorrer do material, aprofundaremos nas características específicas, variações e exemplos práticos de Middleware, enriquecendo ainda mais nosso entendimento e habilidade em aplicar esse conhecimento no desenvolvimento de soluções tecnológicas.

Exemplos e aplicações Middleware

Prosseguindo, aprofundaremos nosso entendimento por meio de exemplos práticos e aplicações de Middleware no mundo tecnológico. Por exemplo, considere o cenário em que um aplicativo de e-commerce precisa integrar-se com um sistema de pagamentos externo. Aqui, um Middleware pode atuar como um conector, traduzindo chamadas e dados entre o aplicativo e o sistema de pagamentos, garantindo uma transação segura e eficiente sem que o usuário final perceba essa complexa interação nos bastidores.

Outro exemplo prático é o uso de Middleware em sistemas de monitoramento de saúde, onde sensores corporais coletam dados de pacientes em tempo real. Esses dados podem ser transmitidos através de um Middleware para uma plataforma de análise de saúde, permitindo a médicos monitorar e responder a emergências de saúde de forma mais rápida e informada.

Há, também, as aplicações de mensageria. Essas plataformas, como o RabbitMQ e o Kafka, exemplificam perfeitamente o conceito de Middleware, situando-se no cerne da comunicação entre sistemas. Elas gerenciam o fluxo de mensagens em arquiteturas complexas, como os microsserviços, assegurando que os dados sejam processados de maneira eficiente e ordenada, evitando sobrecarga e garantindo a integridade das informações.

Podemos citar, também, o Business Intelligence (BI), um processo essencial para a análise e gestão de dados. Ferramentas como o Power BI permitem a coleta, organização e visualização de dados de maneira estruturada, facilitando a tomada de decisões com base em informações consolidadas. O BI atua como um Middleware ao intermediar o acesso e a interpretação dos dados armazenados, oferecendo insights valiosos através de dashboards interativos e análises profundas.

A autenticação é outro exemplo crucial de Middleware, essencial para a segurança dos sistemas. Essa camada de Middleware gerencia o acesso através da verificação de credenciais, como usuário, senha e tokens, garantindo que apenas entidades autorizadas possam acessar recursos específicos. A autenticação pode ser

implementada via API Gateways, reforçando a segurança e o controle de acesso em arquiteturas de software.

Por último, o balanceamento de carga, vital em arquiteturas de microsserviços e sistemas de alta disponibilidade. Este tipo de Middleware distribui dinamicamente as solicitações de usuários ou serviços entre vários servidores, ou instâncias, otimizando o uso de recursos e garantindo uma experiência de usuário estável e confiável.

Esses exemplos visam solidificar a compreensão de como o Middleware opera efetivamente como um intermediário, facilitando, protegendo e otimizando a comunicação e o processamento de dados entre diferentes componentes de um sistema.

Ao concluir esta parte, espera-se que os alunos tenham uma visão clara e aplicável de Middleware, reconhecendo sua importância e versatilidade em diversos contextos tecnológicos. Nas próximas sessões, continuaremos a explorar outras variações e capacidades do Middleware, ampliando ainda mais nosso entendimento sobre este conceito essencial no desenvolvimento de software.

Exemplos Básicos de Middleware sem API

Avançando, nos debruçaremos sobre exemplos básicos de Middleware que operam sem a necessidade de APIs, uma dimensão até então pouco explorada, mas essencial para compreender a amplitude e versatilidade do Middleware no contexto tecnológico. Adentraremos no universo do Business Process Management (BPM) e do Business Activity Monitoring (BAM), explorando como essas ferramentas desempenham o papel de Middleware sem uma interface de programação de aplicações (API) direta.

BPM e BAM são conceitos centrais no gerenciamento e monitoramento de processos de negócios, respectivamente. Embora se assemelhem ao Business Intelligence (BI) em sua finalidade de analisar e otimizar operações, seu foco está mais alinhado ao gerenciamento e eficiência dos processos empresariais. Essas

ferramentas, embora não sejam Middleware no sentido tradicional de facilitar a comunicação direta através de APIs, desempenham um papel crucial na integração e no melhoramento da cooperação entre sistemas distintos e departamentos em uma organização.

A prática proposta envolve a exploração de plataformas como o Miro, [Monday.com](https://monday.com), Kissflow, Pipefy, e SmartSheet, exemplificando a integração de processos de negócios sem a intervenção direta de APIs. Estas ferramentas permitem a criação de fluxos de trabalho automatizados, gerenciamento de tarefas e a análise de performance do negócio, atuando como intermediários que facilitam a gestão de projetos e equipes. Apesar de sua natureza não-API, elas se conectam e se integram a outros sistemas através de APIs disponibilizadas por essas plataformas, permitindo uma comunicação eficiente entre diferentes ferramentas e sistemas usados em uma empresa.

A discussão se aprofunda ao analisar como o Middleware, sem API, pode se integrar a uma arquitetura composta por soluções de BI, uma vez que, ao contrário do que ocorre com o BI, o Middleware, com foco em BPM e BAM, proporciona uma perspectiva mais ampla e direcionada ao gerenciamento de processos. Esse tipo de Middleware analisa o fluxo de trabalho, sugere melhorias, e automatiza tarefas baseadas no comportamento e desempenho dos processos na organização.

Ao final desta parte da aula, espera-se que os alunos tenham uma compreensão clara de como o Middleware sem API, particularmente através do BPM e BAM, desempenha um papel essencial na orquestração de processos empresariais. Essas ferramentas oferecem uma maneira eficaz de integrar, monitorar e melhorar continuamente as operações de negócios, garantindo uma gestão eficiente e a otimização dos processos internos. A seguir, abordaremos como aplicar esses conceitos de Middleware em projetos práticos na disciplina, proporcionando uma experiência hands-on e relevante para os futuros desafios profissionais dos alunos.

Saiba mais sobre Middleware

Por fim, chegamos ao momento de consolidar nosso aprendizado e ampliar nosso conhecimento, explorando a diversidade de contextos e aplicações dos Middlewares com exemplos concretos de sua importância prática no mercado de trabalho. Imagine, por exemplo, um Middleware de autenticação utilizado por bancos para garantir a segurança das transações online dos clientes. Esse Middleware verifica a identidade do usuário e autoriza as transações, atuando como uma camada de segurança crítica entre o usuário e os sistemas bancários.

Além disso, considere o uso de Middleware em sistemas de logística para otimizar a entrega de pacotes. Neste caso, o Middleware pode integrar sistemas de rastreamento de veículos, gerenciamento de inventário e plataformas de atendimento ao cliente, proporcionando uma visão unificada e eficiente do processo de entrega.

Middlewares são elementos cruciais na arquitetura de software, atuando como intermediários para facilitar a comunicação e integração entre diferentes sistemas e aplicações. A aplicabilidade dos Middlewares é vasta, incluindo desde servidores web e aplicações em nuvem, até sensores e periféricos, como sensores de temperatura ou de fumaça. Esses exemplos reforçam a ideia de que Middlewares podem assumir diversas formas e funções, dependendo do contexto e das necessidades específicas de cada projeto.

Além de revisar conceitos já explorados, como automações e mensageria, esta aula também enfatiza a importância de considerar as soluções de Middleware já disponíveis antes de se aventurar na criação de novas implementações. A profusão de ferramentas e soluções no mercado pode, muitas vezes, oferecer respostas eficientes e de bom custo-benefício para os desafios encontrados no desenvolvimento de projetos tecnológicos.

Incentivo a manter uma mentalidade aberta e criativa na busca por soluções de Middleware, lembrando-nos que, em muitos casos, uma simples automação, um script, ou uma API específica podem resolver complexidades de forma surpreendentemente simples e eficaz. Essa perspectiva não apenas otimiza o

desenvolvimento de soluções tecnológicas, mas também enriquece nossa capacidade analítica e nosso repertório de ferramentas ao enfrentarmos desafios no campo da tecnologia da informação.

Encerrando esta aula sobre Middleware, estamos agora mais preparados para aplicar estes conceitos em contextos práticos, seja no mercado de trabalho, em entrevistas de emprego, ou na elaboração e explicação de projetos tecnológicos. Com a compreensão de que qualquer elemento que atue como intermediário em um processo pode ser considerado um Middleware, abrimos caminho para uma vasta gama de aplicações inovadoras e eficientes.

Nas próximas aulas, daremos um passo adiante, trazendo um Middleware de autenticação para ser integrado a uma aplicação em Node.js, oferecendo uma oportunidade valiosa para visualizar e praticar a implementação desses conceitos em um ambiente de desenvolvimento real. Até lá, convido a todos a refletir sobre as possibilidades que o entendimento aprofundado de Middleware nos oferece, tanto para a solução de problemas quanto para a otimização de nossos projetos. Nos vemos na próxima aula!

GitHub da Disciplina

Você pode acessar o repositório da disciplina no GitHub a partir do seguinte link: <https://github.com/FaculdadeDescomplica/ProgramacaoII>. Esse repositório tem como principal objetivo guardar os códigos das aulas práticas da disciplina para aprimorar suas habilidades em vários tópicos, incluindo a criação e consumo de APIs com controle de autenticação utilizando Node.js e utilizando boas práticas de programação e mercado.

Conteúdo Bônus

Para aprofundar seus conhecimentos sobre Middleware e entender ainda melhor seu papel entre diferentes softwares, sugiro o vídeo “Middleware (A ‘Cola’ dos

Softwares) // Dicionário do Programador” do canal Código Fonte TV no YouTube. Este vídeo é uma excelente fonte de informação, apresentando de forma didática e acessível os conceitos fundamentais do Middleware, exemplos práticos de sua aplicação, e como ele facilita a comunicação e integração entre diferentes sistemas e aplicações.

Referências Bibliográficas

Bibliografia Básica:

ELMASRI, R.; NAVATHE, S. B. **Sistemas de banco de dados**. 7.ed. Pearson: 2018.

MEDEIROS, L. F. de. **Banco de dados: princípios e prática**. Intersaberes: 2013.

VICCI, C. (Org.). **Banco de dados**. Pearson: 2014.

Bibliografia Complementar:

CARDOSO, L. da C. **Design de aplicativos**. Intersaberes: 2022.

JOÃO, B. do N. **Usabilidade e interface homem-máquina**. Pearson: 2017.

LEAL, G. C. L. **Linguagem, programação e banco de dados: guia prático de aprendizagem**. Intersaberes: 2015.

PUGA, S.; FRANÇA, E.; GOYA, M. **Banco de dados: implementação em SQL, PL/SQL e Oracle 11g**. Pearson: 2013.

SETZER, V. W.; SILVA, F. S. C. **Bancos de dados**. Blucher: 2005.

Ir para exercício