1 Introdução

Com o rápido avanço tecnológico principalmente no âmbito de desenvolvimento de sistemas, é facilmente perceptível, o quanto as empresas estão atrás de soluções e estratégias para seus produtos, visando aumentar diversos aspectos positivos dos mesmos, como a escalabilidade, segurança, qualidade, resiliência e agilidade de desenvolvimento. Nesse contexto, a arquitetura REST (Representational State Transfer) surge como uma das principais abordagens para a criação de APIs, facilitando a comunicação entre sistemas de forma padronizada e eficiente. Dessa forma lidando positivamente com todos os aspectos mencionados anteriormente.

O padrão arquitetural REST foi adotado por muitas empresas, como uma ótima forma de abordar e tratar a comunicação entre os sistemas. Qual a definição de REST, de acordo com Lima(2020) “O princípio fundamental é usar o protocolo HTTP para comunicação de dados. A arquitetura REST é simples e fornece acesso aos recursos para que o cliente REST acesse e renderize os recursos no lado do cliente. No estilo REST, URI ou IDs globais ajudam a identificar cada recurso. Esta arquitetura usa várias representações de recursos para representar seu tipo, como XML, JSON, Texto, Imagens e assim por diante.”

2 Fundamentação teórica

O padrão arquitetural REST é amplamente utilizado e tem como objetivo proporcionar uma solução para tornar sistemas escaláveis e flexíveis, aproveitando seus conceitos de comunicação entre softwares de maneira extensiva.

Stateless (Sem estado): É outro dos princípios restritivos imposto pelo estilo REST e diz respeito à interação entre cliente e servidor. A comunicação deve ser feita sem o armazenamento de qualquer tipo de estado no servidor, ou seja, cada requisição do cliente para o servidor deve conter todas as informações necessárias para que ela seja entendida. Portanto, estados de sessão, quando necessários, devem ser totalmente mantidos no cliente. De Oliveira Junior, J. M., do Nascimento Correa, L. P., & Fonseca, L. C. C. Integração do Moodle com Sistema de Gestão Acadêmica, Página 4.

Cacheable: Cache é o princípio que estabelece uma forma de diminuir o impacto da desvantagem trazida pela redução de desempenho. O mesmo exige que os dados de uma resposta, vindos de uma requisição ao servidor, sejam marcados como cacheable ou noncacheable (passíveis ou não de utilização da cache). Se uma resposta é marcada como cacheable, então ela será utilizada como resposta para as futuras requisições equivalentes. De Oliveira Junior, J. M., do Nascimento Correa, L. P., & Fonseca, L. C. C. Integração do Moodle com Sistema de Gestão Acadêmica, Página 4.

2.1 Princípios e Padrões:

O ideal é que, ao desenvolver aplicações, o usuário aproveite a arquitetura da Web em seu favor. Entre as principais ações a serem consideradas nesse processo, podemos destacar:

* Atribuir identificadores únicos para cada elemento;
* Conectar e tornar os elementos interativos;
* Utilizar métodos que sigam padrões definidos;
* Representar os recursos de formas diversificadas;
* Priorizar uma comunicação sem manutenção de estado.

Apesar de muitas aplicações web não seguirem rigorosamente as convenções de métodos e respostas, é essencial, sempre que viável, programar conforme essas práticas recomendadas.

2.2 Comparação com outras arquiteturas:

Diferenças entre SOAP e REST

O REST, por ser mais intuitivo e amplamente acessível, é mais fácil de compreender. Contudo, há uma falta de padronização clara, sendo amplamente reconhecido como um estilo arquitetural.

Por outro lado, o SOAP já está consolidado no mercado, com protocolos bem definidos e um conjunto rígido de normas. Diferente do REST, que opera sobre HTTP/HTTPS, o SOAP permite que as requisições sejam transmitidas por qualquer canal de transporte disponível, como SMTP ou JMS (Java Messaging Service).

* O SOAP, baseado em XML, funciona através de um envelope que:
* Define o conteúdo da mensagem e orienta seu processamento;
* Estabelece um conjunto de regras de codificação para os tipos de dados;
* Organiza o layout para as chamadas e respostas de procedimentos.

Esse envelope pode ser enviado, por exemplo, via HTTP/HTTPS. Em seguida, uma RPC (Remote Procedure Call) é executada, retornando com as informações no formato XML.

Embora essa abordagem possa ser vista como mais detalhada e com processamento um pouco mais lento, ambas as tecnologias são viáveis para desenvolvedores web. Atendendo a diversas demandas de programação, SOAP e REST podem até ser utilizados em conjunto.

2.3 Aplicações práticas:

Uma aplicação prática de uma arquitetura REST pode ser encontrada em diversos cenários onde sistemas precisam se comunicar de forma eficiente, escalável e independente de plataforma. Exemplo:

* APIs de redes sociais: Aplicações como o Twitter, Facebook ou Instagram utilizam REST para permitir que desenvolvedores integrem funcionalidades dessas plataformas em seus próprios sistemas. Por exemplo, através de uma API REST, um site pode exibir tweets ou postagens diretamente, sem precisar que os usuários saiam do site.

3 Conclusão:

A arquitetura REST transformou significativamente a forma como interagimos com serviços na web, oferecendo uma solução simples, escalável e eficiente para a comunicação entre sistemas. Sua aderência a padrões abertos e seu foco na simplicidade tornam o REST ideal para a criação de APIs robustas e flexíveis.

Entretanto, apesar de suas vantagens, é fundamental que os desenvolvedores considerem as limitações e adequem suas implementações às necessidades específicas do projeto, buscando sempre seguir as boas práticas de design e uso de recursos. No final, o sucesso na adoção da arquitetura REST depende de um equilíbrio cuidadoso entre flexibilidade e padronização, garantindo que as soluções ofereçam o máximo de eficiência e acessibilidade para os usuários e outros sistemas.

Com sua ampla aceitação e suporte em diversas plataformas, REST continua sendo uma das arquiteturas mais relevantes para a web moderna, e seu uso adequado pode ser determinante para o sucesso de projetos de software no longo prazo.

4 Referências bibliográficas

LIMA, Guilherme. REST: Conceito e fundamentos. 2020. Disponível em: [https://www.alura.com.br/artigos/rest-conceito-e-fundamentos](https://www.alura.com.br/artigos/rest-conceito-e-fundamentos#:~:text=REST(Representational%20State%20Transfer%2C%20que,existentes%20da%20Web%20sem%20a). Acesso em: 12 outubro. 2024.

Equipe Totvs: Disponível em: <https://www.totvs.com/blog/developers/rest/.> Acesso em: 12 outubro. 2024.

Oliveira Junior, J. M., do Nascimento Correa, L. P., & Fonseca, L. C. C. Integração do Moodle com Sistema de Gestão Acadêmica. <https://www.tise.cl/Volumen14/TISE2018/272.pdf>