FUNDAMENTOS DE ARQUITETURA DE SISTEMAS

Serviços Web

Soluções para apps se comunicarem independente de linguagem, softwares, hardwares utilizados.

Nem toda API é um serviço Web, mas todo serviço Web é uma API.

A linguagem de marcação é quase universal, permitindo comunicação via protocolo HTTP entre aplicações diferentes.

Vantagens:

-Linguagem comum

-Integração

-Reutilização de implementação

-Segurança

-Custos

Principais Tecnologias:

-SOAP

-REST

-XML

-JSON

SOAP – Simple Object Acess Protocol:

É um protocolo baseado em XML para acessar serviços web principalmente por HTTP.

Vantagens:

-É independente de plataforma e software

-Meio de transporte genérico, ou seja, pode ser usado por outros protocolos além do HTTP.

ESTRUTURA DO SOAP:

-SOAP Envolope – encapsular toda mensagem

-SOAP header – possui informações de atributos e metadados da requisição

-SOAP body – Contém os detalhes da mensagem

WSDL – Web Services Description Language

Usado para descrever Web Services, funciona como um contrato do serviço.

A descrição é feita em um documento XML

XSD – XML Schema Definition

Usado para definir a estrutura de dados que será validada no XML

Documentação de como deve ser montado o SOAP Message.

REST – Representational State Transfer

É um estilo de arquitetura de software que define a implementação de um serviço web

Vantagens:

* Permite integrações entre aplicações e entre cliente e servidor em páginas web aplicações.
* Arquitetura de fácil compreensão.

**API – Application Programming Interface.**

São conjuntos de rotinas documentados e disponibilizados por aplicação para que outras aplicações possam consumir suas funcionalidades.

Código de Estado: Usado pelo servidor para avisar o cliente sobre o estado da operação solicitada

* 1xx – Informativo
* 2xx – Sucesso
* 3xx – Redirecionamento
* 4xx – Erro do Cliente
* 5xx – Erro do Servidor

Conceito de Arquitetura em aplicações para internet

Comparando Monolito e Micro serviços

-Prós e contras Monolito

Pros:

* Baixa complexidade
* Monitoramento simplificado

Contras:

* Stack única
* Compartilhamento de recursos
* Acoplamento
* Mais complexo a escalabilidade

Micro serviços

-Prós:

* Stack dinâmica
* Simples escalabilidade

-Contras:

* Acoplamento
* Monitoramento mais complexo
* Provisionamento mais complexo

**Gerenciamento de erros**

Onde é mais complexo:

* Processos assíncronos (Micro serviços)
* Pipeline

Solução

* Dead letter queue
* Filas de tentativas