

Web Services

Ementa

- ▶ O que é um WebService
- ▶ O que é uma API
- ▶ WS x API
- ▶ SOAP x REST
- ▶ Criando um WebService SOAP
- ▶ Criando um WebService REST
- ▶ Criando um cliente para consumir os WebServices

WebService

WebService (O que é)

- ▶ É um conjunto de métodos que são invocados por outros programas utilizando tecnologias Web.
- ▶ Um Webservice (WS) é utilizado para transferir dados através de protocolos utilizadas nessas plataformas.
- ▶ Não confundir WS com uma página Web, por exemplo, um WS transmite apenas informações.
- ▶ Um WS é muito utilizado em integrações entre sistemas.

WebService (Como Funciona)

- ▶ Em todo WS existe um protocolo de comunicação para que sistemas externos possam se “conversar”.
- ▶ Os dois mais utilizados são: **SOAP** e o **REST**

WebService (Benefícios)

- ▶ **Integração de informações entre Sistemas:** É possível realizar a troca de informações entre dois sistemas. Permitem ligar qualquer tipo de sistema, independente de plataforma (Windows, Linux e etc...) e independente de linguagem (Java, Perl, Python, Delphi e etc...).
- ▶ **Reutilização de Código:** Um WS pode ser utilizado por várias plataformas com diferentes objetivos de negócio. O código é feito uma única vez e utilizado várias vezes e em lugares diferentes.
- ▶ **Maior Segurança:** Um WS evita que se comuniquem diretamente com a base de dados.

API

API (O que é)

- ▶ É um acrônimo para *Interface de Programação de Aplicativos* que permite a comunicação entre dois softwares.
- ▶ Fornecem serviços de softwares que se comunicam com serviços de outros softwares sem precisar saber como ambos são implementados.

API x WS (Qual é a Diferença?)

- ▶ A principal diferença da API para o WS está no tipo de protocolo de comunicação. Enquanto o WS é usado para REST, SOAP e XML, a API é utilizada para qualquer padrão de comunicação.

API x WS (Qual é a Diferença?)

WebService

- ▶ É uma aplicação
- ▶ Todo Webservice é uma API
- ▶ Protocolo de comunicação robusto e complexo, menos seguro
- ▶ Métodos customizados
- ▶ Não armazena os dados em Cache
- ▶ Linguagem complexa e difícil de integrar
- ▶ Necessário ter uma rede para o seu funcionamento

API

- ▶ É uma interface
- ▶ Todas as API não são Webservice
- ▶ Arquitetura de Software com mais recursos, abordagem simplificada, com diretrizes mais simples e mais segura
- ▶ Métodos padronizados
- ▶ Armazena dados em Cache
- ▶ Não necessita de um rede para o seu funcionamento (Serverless)

SOAP

SOAP (O que é)

- ▶ SOAP (Simple Object Access Protocol - Procolo de acesso simples a objetos).
- ▶ Utiliza como mensagem documentos do tipo XML para a troca de informações e, geralmente, serve-se do protocolo HTTP/HTTPS para transportar tais dados.
- ▶ Juntamente com o protocolo SOAP temos o documento WSDL (Web Service Definition Language), que descreve a localização do WS, as operações disponíveis e informações para realizar a comunicação.

SOAP (Como Testar?)

- ▶ Software mais popular para testar WS SOAP's:
- ▶ <https://www.soapui.org/downloads/soapui/>



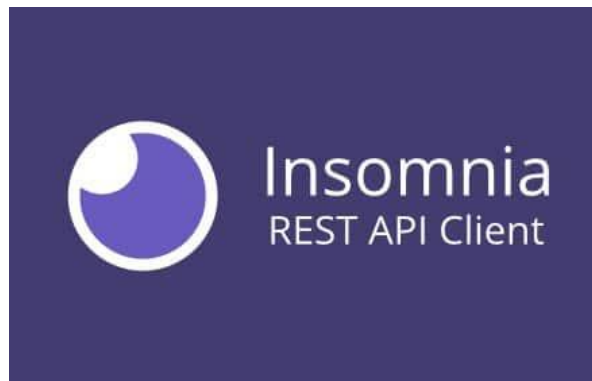
REST

REST (O que é)

- ▶ REST (Representational State Transfer - Transferência de Estado Representacional)
- ▶ É um protocolo de comunicação mais recente que veio para simplificar a comunicação com os WS's.
- ▶ Também se baseia no protocolo HTTP/HTTPS e permite utilizar vários formatos de documentos na troca mensagem, como: JSON, XML, RSS e outros.
- ▶ É um dos protocolos mais rápidos e com maior capacidade de transmissão de dados.

REST (Como Testar?)

- ▶ Software mais popular para testar WS REST:
- ▶ <https://www.postman.com/downloads/>



Exercícios

01 - Instalando os Clientes para SOAP e REST

02 - Testando um SOAP com o SoapUI

03 - Testando um REST com o Insomnia

Verbos HTTP

Verbos HTTP

- ▶ Sua API deverá prover uma URL base e os clientes irão operar de acordo com os verbos HTTP. (www.dominio.com/rest/notas/)
- ▶ **Verbo GET:** Sem passagem de ID, retorna todos os dados daquele serviço/recurso. Com a passagem de ID, retorna um registro específico.
- ▶ **Verbo POST:** Cria um novo dado daquele serviço/recurso. Passando o ID como parâmetro.
- ▶ **Verbo DELETE:** Remove o dado daquele serviço/recurso. Passando o ID como parâmetro.
- ▶ **Verbo PUT:** Atualiza o dado daquele serviço/recurso. Passa o ID como parâmetro e passa os dados no body da requisição (normalmente no formato JSON).
- ▶ **Verbo PATCH:** Atualiza o dado daquele serviço/recurso. Passa o ID como parâmetro e passa somente os dados atualizados no body da requisição (normalmente no formato JSON).

Status Code HTTP

Status Code HTTP

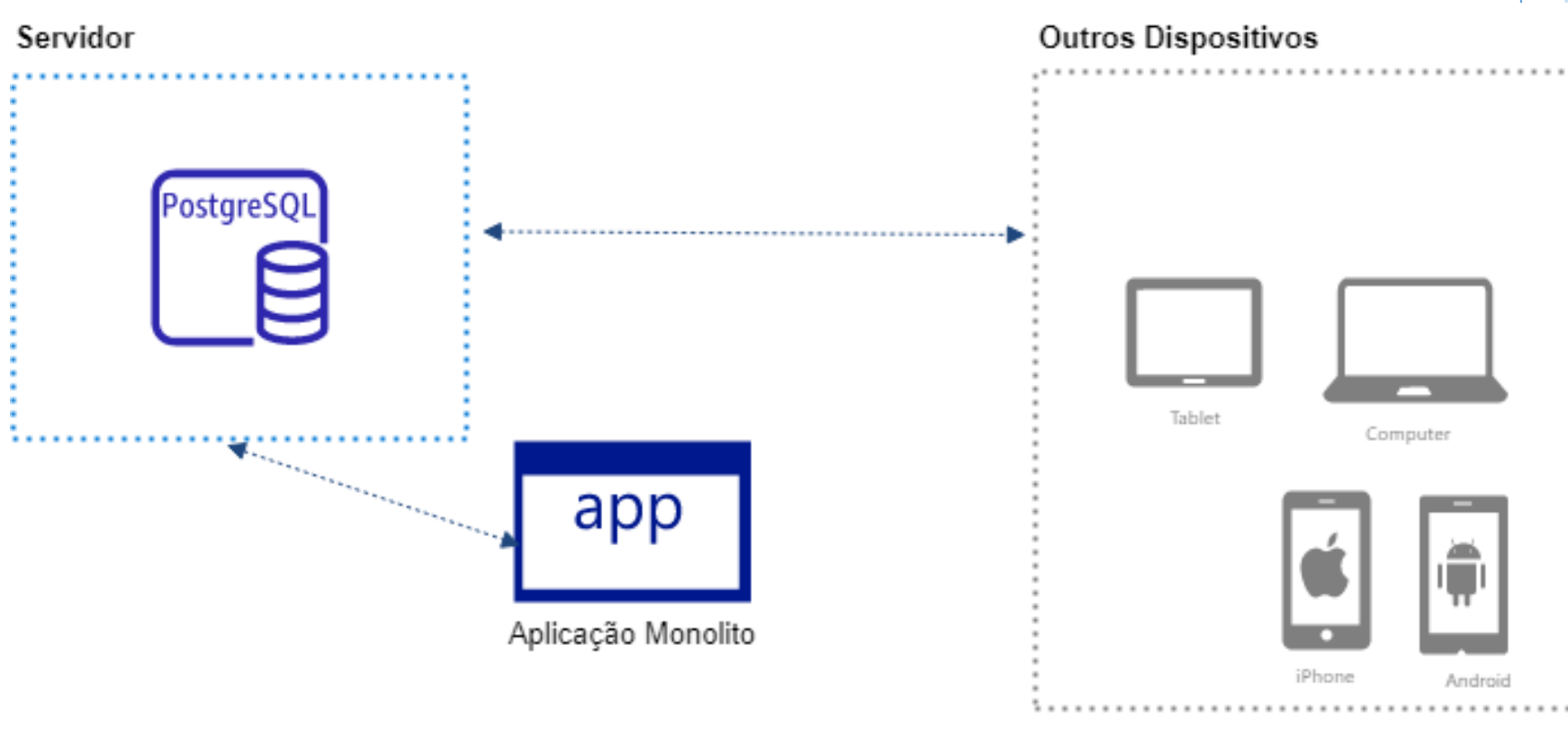
Os mais comuns que vamos ver:

- ▶ 200, 400, 404 e 500.

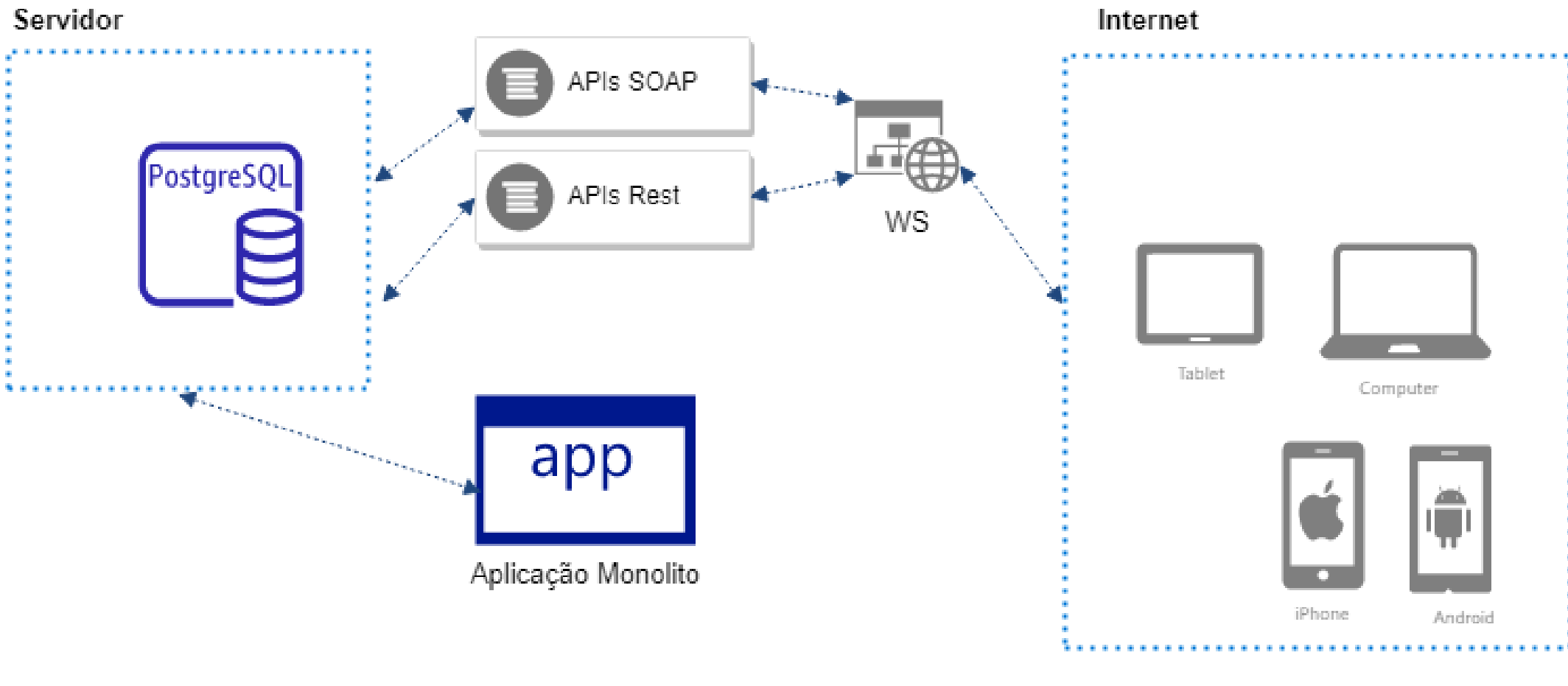
1XX Informational		4XX Client Error Continued	
100	Continue	409	Conflict
101	Switching Protocols	410	Gone
102	Processing	411	Length Required
2XX Success		412	Precondition Failed
200	OK	413	Payload Too Large
201	Created	414	Request-URI Too Long
202	Accepted	415	Unsupported Media Type
203	Non-authoritative Information	416	Requested Range Not Satisfiable
204	No Content	417	Expectation Failed
205	Reset Content	418	I'm a teapot
206	Partial Content	421	Misdirected Request
207	Multi-Status	422	Unprocessable Entity
208	Already Reported	423	Locked
226	IM Used	424	Failed Dependency
3XX Redirectional		426	Upgrade Required
300	Multiple Choices	428	Precondition Required
301	Moved Permanently	429	Too Many Requests
302	Found	431	Request Header Fields Too Large
303	See Other	444	Connection Closed Without Response
304	Not Modified	451	Unavailable For Legal Reasons
305	Use Proxy	499	Client Closed Request
307	Temporary Redirect	5XX Server Error	
308	Permanent Redirect	500	Internal Server Error
4XX Client Error		501	Not Implemented
400	Bad Request	502	Bad Gateway
401	Unauthorized	503	Service Unavailable
402	Payment Required	504	Gateway Timeout
403	Forbidden	505	HTTP Version Not Supported
404	Not Found	506	Variant Also Negotiates
405	Method Not Allowed	507	Insufficient Storage
406	Not Acceptable	508	Loop Detected
407	Proxy Authentication Required	510	Not Extended
408	Request Timeout	511	Network Authentication Required
		599	Network Connect Timeout Error

Arquiteturas

Arquitetura Client/Server - Sem WS



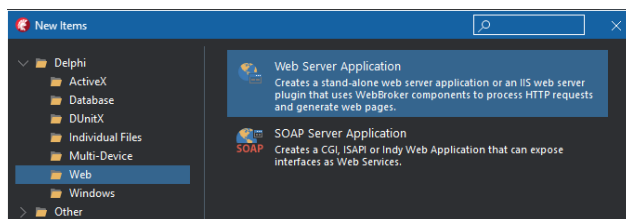
Arquitetura Cloud - Com WS



Delphi e WebServices

Delphi e WebServices

- ▶ Alguns Frameworks para criação de WebServices no Delphi:



<- SOAP - Nativo



RAD Server



<- REST - Nativos



Horse



<- REST - Open Source

Criando um WS em SOAP

Exercícios

04 - Criando o primeiro WS SOAP

05 - WS de uma Calculadora

Criando um WS em REST

Exercícios

06 - Criando o primeiro WS REST

07 - WS de uma Calculadora

Criando um Client

Para consumir WS em SOAP

Exercícios

08 - Consumir o Hello World

09 - Consumir a Calculadora

Criando um Client

Para consumir WS em REST

Exercícios

10 - Consumir o Hello World

11 - Consumir a Calculadora