

ES46A - Arquitetura de Software

Arquitetura em Camadas

DACOM – Engenharia de Computação

Diego Addan

UTFPR - 2023

Para hoje..

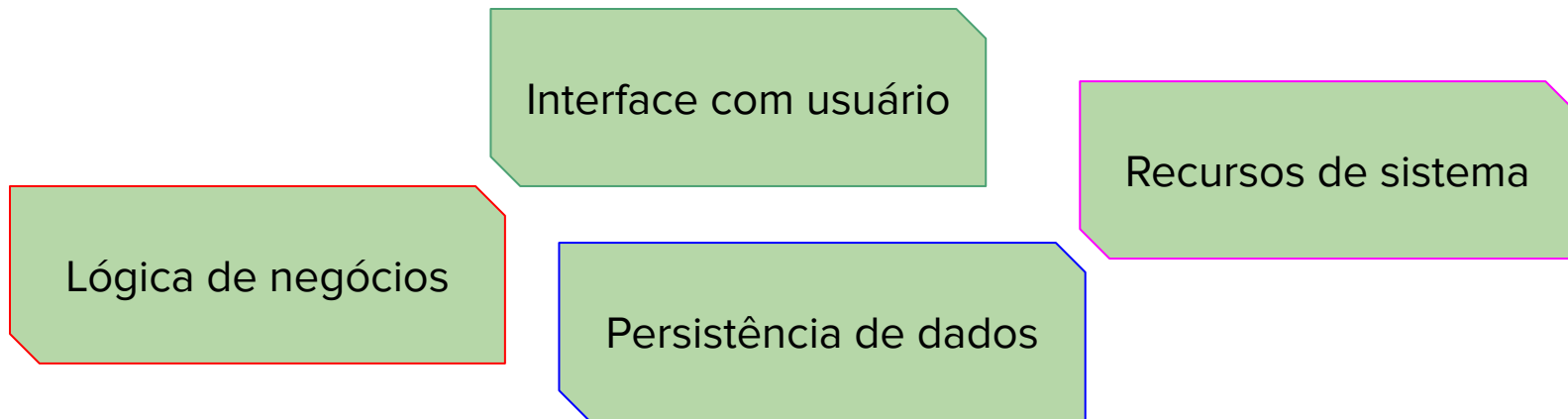
Arquitetura em camadas para aplicações Web

MVC

SOA

Na última aula

Tem como base a organização de componentes lógicos por meio de camadas/categorias de funcionalidades



Na última aula

Tipos de arquitetura em camadas: Monolítica e Microserviços

Model, Service e Controller: comumente utilizado no Backend.

Contextos ~ Aplicações

Arquitetura em camadas: MVC

MVC (Model-View-Controller) é um padrão de projeto de software

- Um padrão de projeto é uma solução para um problema dentro de um contexto
- Ao lidar com um cliente observa-se que os requisitos do software mudam constantemente. Desta maneira separar a lógica de negócios da apresentação facilita a manutenção e modificação do sistema

Arquitetura em camadas: MVC

Modelo:

- Representa a lógica do negócio
- Armazena as entidades do sistema
- Pode interagir com a base de dados

Visão:

- Responsável por exibir as informações ao usuário

Controle:

- Interliga a Visão e o Modelo, Coordenam todos os componentes

Arquitetura em camadas: MVC

MVC em ambiente web:

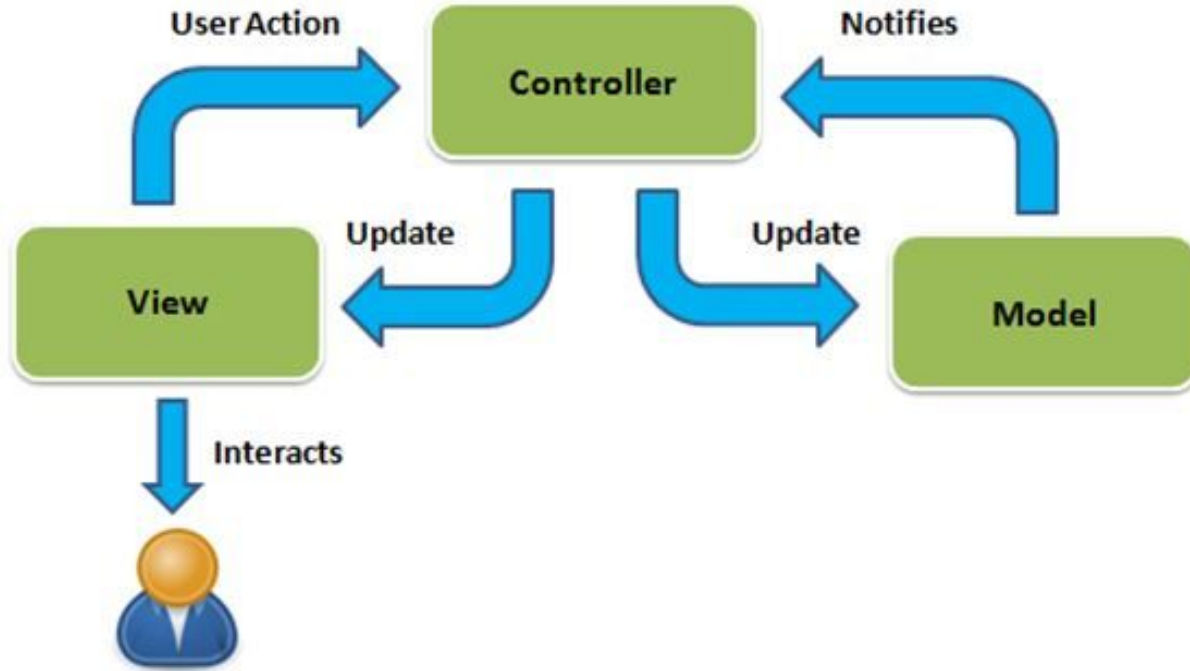
MVC retira a lógica de negócios e a interface de dentro dos scripts funcionais (Servlets)

A lógica de negócios é colocada em classes simples e se torna a camada **Modelo**

A interface é colocada dentro de templates (XML, JSP) e se torna a camada de **Visão**

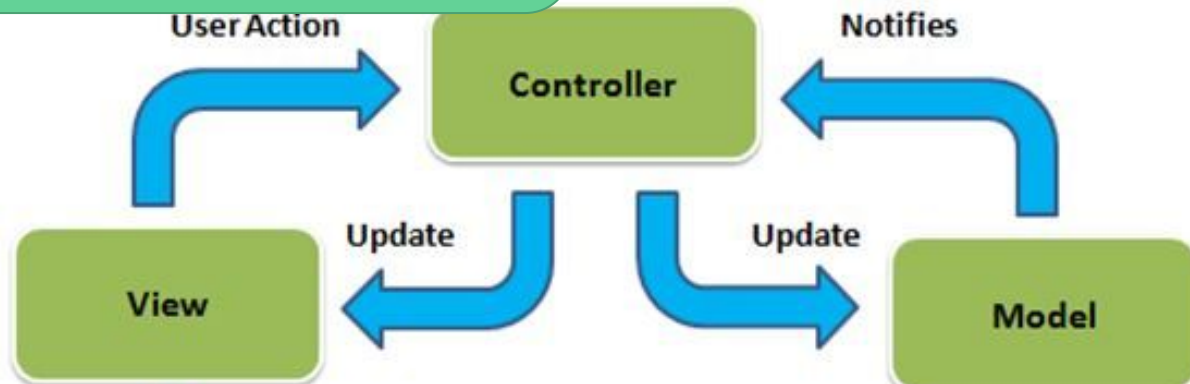
O que sobra nos scripts (Servlets) se torna a camada de **controle** do sistema

Arquitetura em camadas: MVC



MVC

1-Recupera a entrada do usuário a partir da requisição e traduz para o modelo.
2-Diz ao modelo para se atualizar e disponibiliza um novo estado para a visão



1-Responsável pela apresentação.
2-Recupera o estado e o modelo através do controller
3-Obtém entradas do usuário e repassa para o controller

1-Guarda a lógica e o estado do negócio.
2-Conhece as regras para recuperar e alterar o estado.
3-É a parte do sistema que interage com o banco de dados.

Arquitetura em camadas: MVC

Exemplo de processo simples Web de um sistema utilizando MVC que recebe o login de um usuário.

JSP, Servlet e banco de dados

Qualquer tipo de aplicação web tradicional pode ser adaptada a este tipo de arquitetura!

Arquitetura em camadas: MVC

Model: Recuperação do usuário pelo 'login' no BD

```
public Usuario getUsuario(String login) throws SQLException{
    ResultSet rs;
    String SQL = "select * from usuario where login = '" +
        login + "'";
    Usuario user = new Usuario();
    rs = stmt.executeQuery(SQL);
    while (rs.next()){
        user.setLogin(login);
        user.setId(rs.getInt("id"));
        user.setSenha(rs.getString("senha"));
        user.setNivelAcesso(rs.getInt("nivelAcesso"));
    }
    return user;
}
```

```
public class Usuario {
    private int id;
    private String login;
    private String senha;
    private int nivelAcesso;

    //GETs e SETs
}
```

Arquitetura em camadas: MVC

View: Tela de Login

```
<body>
  
  <%
    session.setAttribute("login", null);
    session.setAttribute("id", null);
  %>
  <h1>Sistema de Empréstimo de Livros</h1>
  <form action="VerificacaoUsuarios" method="processRequest">
    Login: <input type="text" name="login"><br><br>
    Senha: <input type="password" name="senha"><br><br>
    <input type="submit" value="LOGIN">
  </form>
  <br><a href="CadastrarUsuario.jsp">Não sou cadastrado ainda!</a>
</body>
```

Arquitetura em camadas: MVC

Controller: Servlet de comunicação lógica

```
@WebServlet(name = "VerificacaoUsuarios", urlPatterns = {"/VerificacaoUsuarios"})
public class VerificacaoUsuarios extends HttpServlet {
    protected void processRequest(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException, SQLException {
        response.setContentType("text/html;charset=UTF-8");

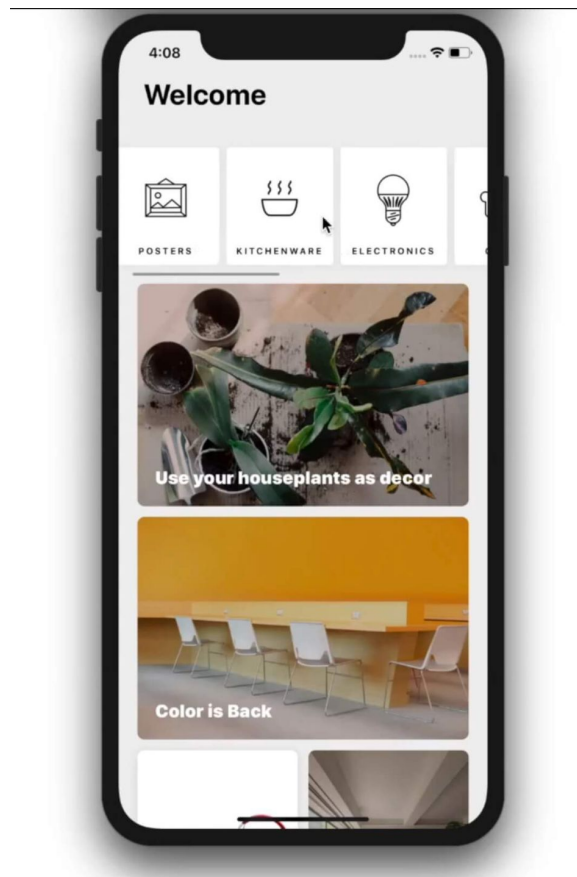
        //1-RECUPERAR DADOS DO JSP (VIEW)
        String login = request.getParameter("login");
        String senha = request.getParameter("senha");

        //2-ACESSAR O MODELO PARA VERIFICAR OS DADOS DO USUÁRIO
        Banco b = new Banco();
        Usuario user = b.getUsuario(login);
    }
}
```

Arquitetura em camadas: MVC

Aplicações em outros contextos

Pode ser que o modelo MSC seja melhor!



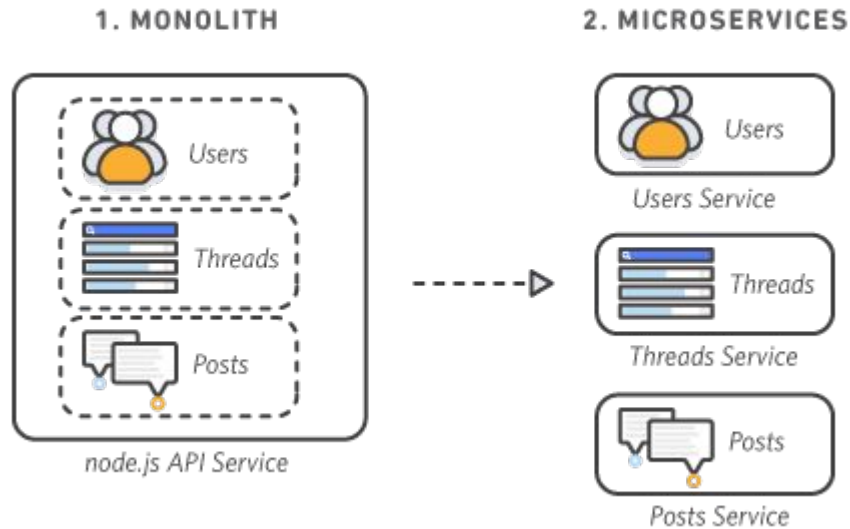
Arquitetura em camadas: MVC

Esta arquitetura facilita o desenvolvimento, manutenção e reaproveitamento de código

- Mudar a interface sem mudar o resto**
- Mudar os dados sem mudar o resto**
- Mudar o processo sem mudar o resto**

Arquitetura em camadas: SOA

ARQUITETURA ORIENTADA A SERVIÇO



Arquitetura em camadas: SOA

ARQUITETURA ORIENTADA A SERVIÇO

Um meio de desenvolver sistemas distribuídos em que os **componentes** são **serviços** autônomos.

Fracamente acoplado!

Serviços podem executar em diferentes computadores a partir de diferentes fornecedores de serviços.

Foram desenvolvidos protocolos-padrão para dar suporte à comunicação de serviços e troca de informações.

Arquitetura em camadas: SOA

Serviços podem ser prestados localmente ou terceirizados para provedores externos.

Os serviços são **independentes da linguagem**.

Investimentos em sistemas legados podem ser preservados.

A computação Inter organizacional é facilitada por meio da troca simplificada de informações. O serviço possui um contrato que descreve suas regras!

Arquitetura em camadas: SOA

ARQUITETURA ORIENTADA A SERVIÇO



Arquitetura em camadas: SOA

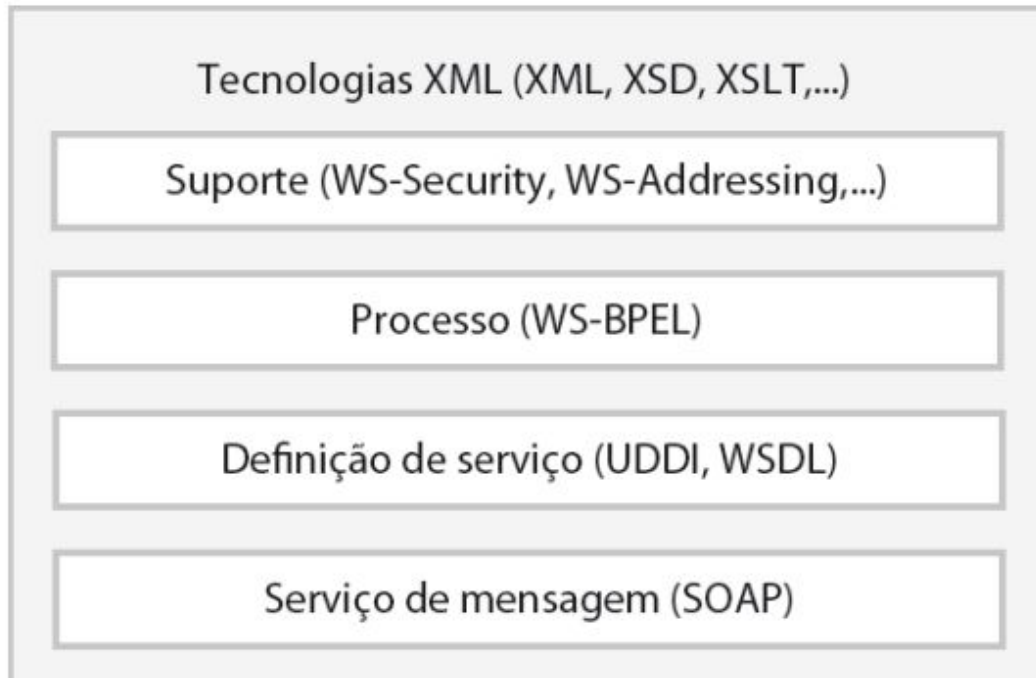
SOAP Um padrão de troca de mensagens que suporta a comunicação de serviço.

WSDL (Web Service Definition Language) Esse padrão permite que sejam definidas uma interface de serviços e suas vinculações.

WS-BPEL Um padrão para linguagens de workflow usado para definir a composição de serviços.

Arquitetura em camadas: SOA

Padrões de Web Services



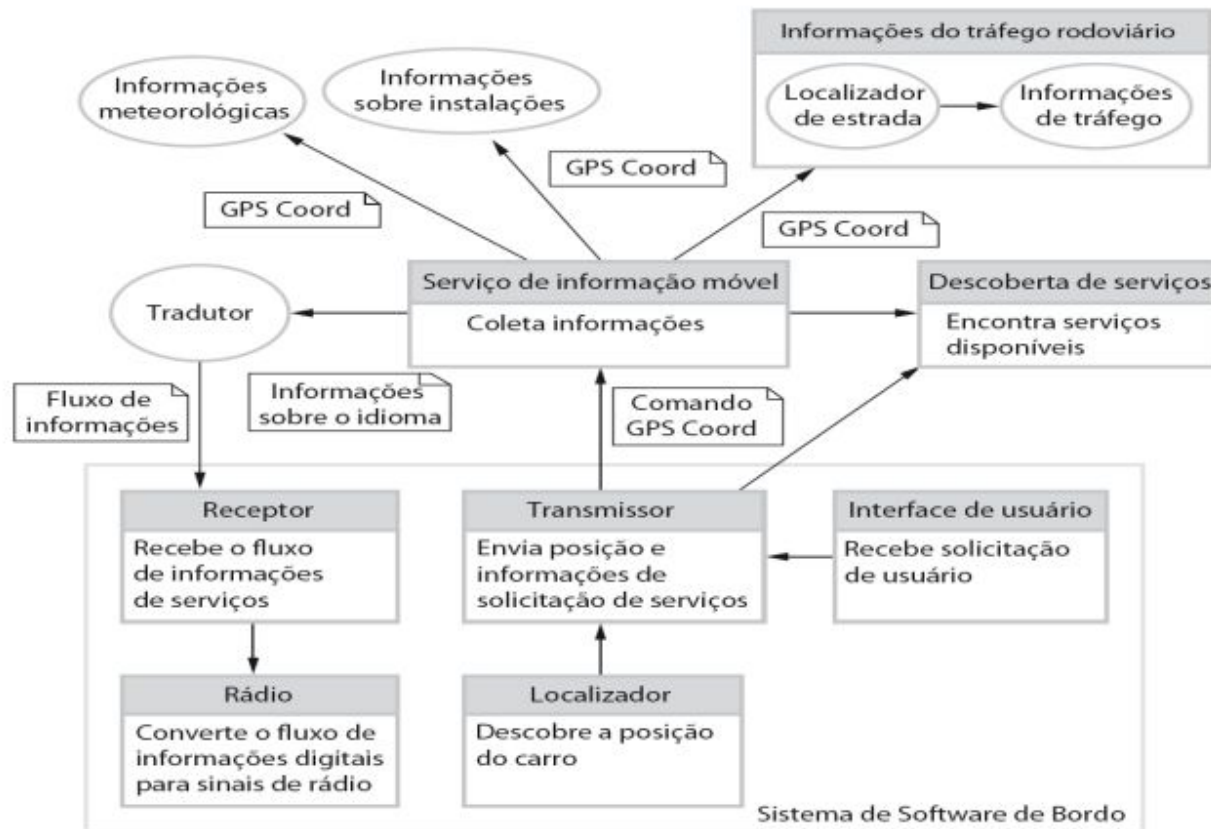
Arquitetura em camadas: SOA

Exemplo:

Um sistema de informação no interior do veículo oferece aos motoristas informações sobre o **clima, condições do tráfego rodoviário, etc.** Esse está ligado ao rádio do carro para que a informação seja entregue como um sinal em um canal de rádio específico.

O carro é equipado com receptores GPS para descobrir a sua posição e, com base nessa posição, o sistema acessa uma gama de serviços de informação. As informações podem ser entregues na linguagem especificada pelo motorista

UM SISTEMA DE INFORMAÇÕES DE BORDO BASEADO EM SERVIÇOS



Arquitetura em camadas: SOA

Vantagens de SOA neste problema

Não é necessário decidir quando o sistema é programado ou implantado, qual prestador do serviço deve ser usado ou quais serviços específicos devem ser acessados.

Como o carro se move, o software usa o serviço de descoberta de serviços para encontrar o serviço de informação mais adequado e se conecta a esse.

Por causa do uso de um serviço de tradução, ele pode mover-se através das fronteiras e, portanto, tornar as informações locais disponíveis para as pessoas que não falam o idioma local

Arquitetura em camadas: SOA

Um serviço:

encapsula funcionalidade discreta a qual pode ser distribuída e acessada por meio de programas.

Um **web service** pode ser um serviço que é acessado usando protocolos-padrão de Internet e é baseado em XML (exemplo).

SOA não descreve como projetar ou construir a implementação de serviço.

Arquitetura em camadas: SOA

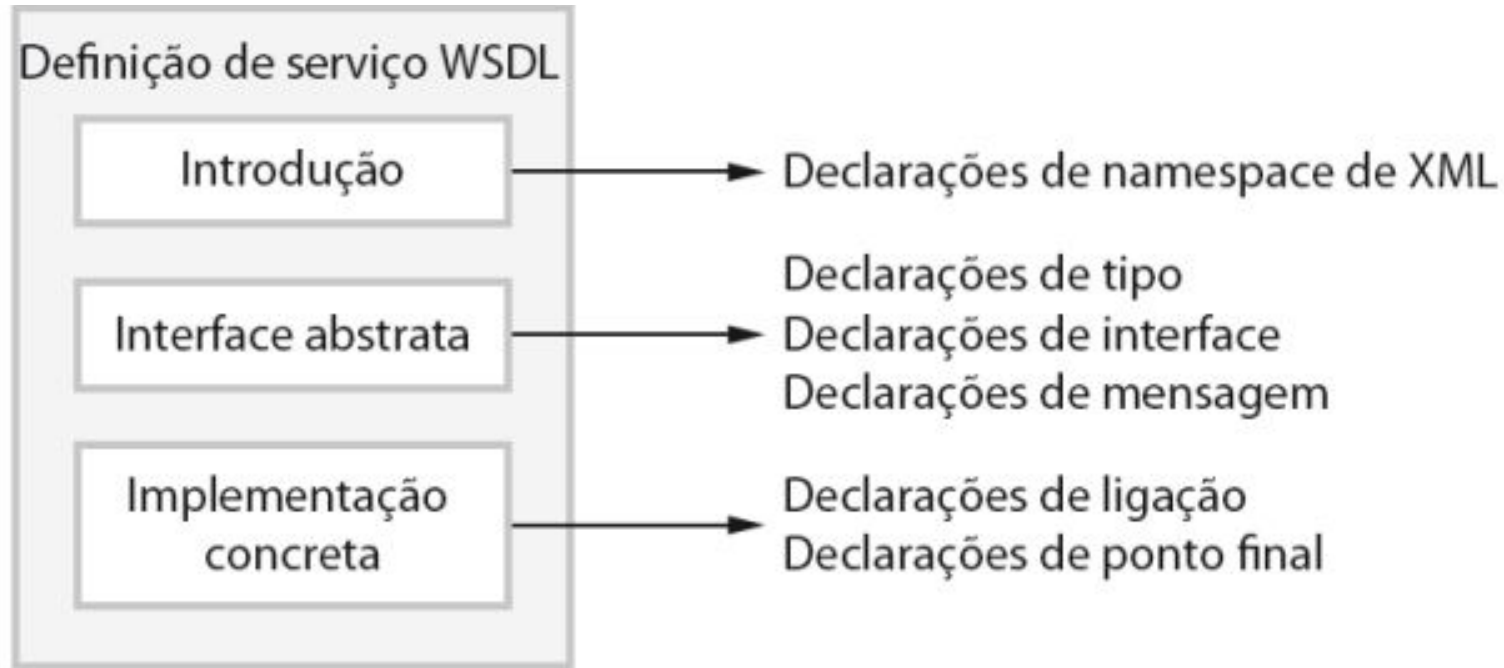
A interface de serviço é definida em uma **descrição de serviço** expressa em **WSDL** (Web Service Description Language).

A especificação WSDL define:

- As operações as quais o serviço oferece suporte e o **formato** das mensagens que são enviadas e recebidas pelo serviço.
- Como o serviço é acessado - isto é, os mapas de ligação da interface abstrata em um conjunto concreto de protocolos.
- Onde o serviço está localizado. Geralmente esse é expresso como uma **URI (Universal Resource Identifier)**.

Arquitetura em camadas: SOA

Especificação WSDL



Arquitetura em camadas: SOA

A parte 'o que' de um documento WSDL, chamada de **interface**, especifica as operações que o serviço suporta, e define o formato das mensagens que são enviadas e recebidas pelo serviço.

A parte 'como' de um documento WSDL, chamada de **ligação**, mapeia a interface abstrata para um conjunto concreto de protocolos. A ligação especifica os detalhes técnicos de como se comunicar com um web service.

A parte 'onde' de um documento WSDL descreve a **localização** de uma implementação específica de web service (endpoint).

Arquitetura em camadas: SOA

```
<wsdl:definitions targetNamespace="http://math.example.com" name="MathFunctionsDef">
  <wsdl:message name="addIntResponse">
    <wsdl:part name="addIntReturn" type="xsd:int" />
  </wsdl:message>
  <wsdl:message name="addIntRequest">
    <wsdl:part name="a" type="xsd:int" />
    <wsdl:part name="b" type="xsd:int" />
  </wsdl:message>
  <wsdl:portType name="AddFunction">
    <wsdl:operation name="addInt" parameterOrder="a b">
      <wsdl:input message="impl:addIntRequest" name="addIntRequest" />
      <wsdl:output message="impl:addIntResponse" name="addIntResponse" />
    </wsdl:operation>
  </wsdl:portType>
  <service name="MathFunctions"/>
</wsdl:definitions>
```

Arquitetura em camadas: SOA

Protocolos de comunicação

Alguns protocolos padrão para implementar a SOA incluem:

- **Protocolo Simples de Acesso a Objetos (SOAP)**
- **HTTP RESTful**
- **Apache Thrift**
- **Apache ActiveMQ**
- **Serviço de Mensagens Java (JMS)**

Arquitetura em camadas: SOA

WEB SERVICES RESTFUL

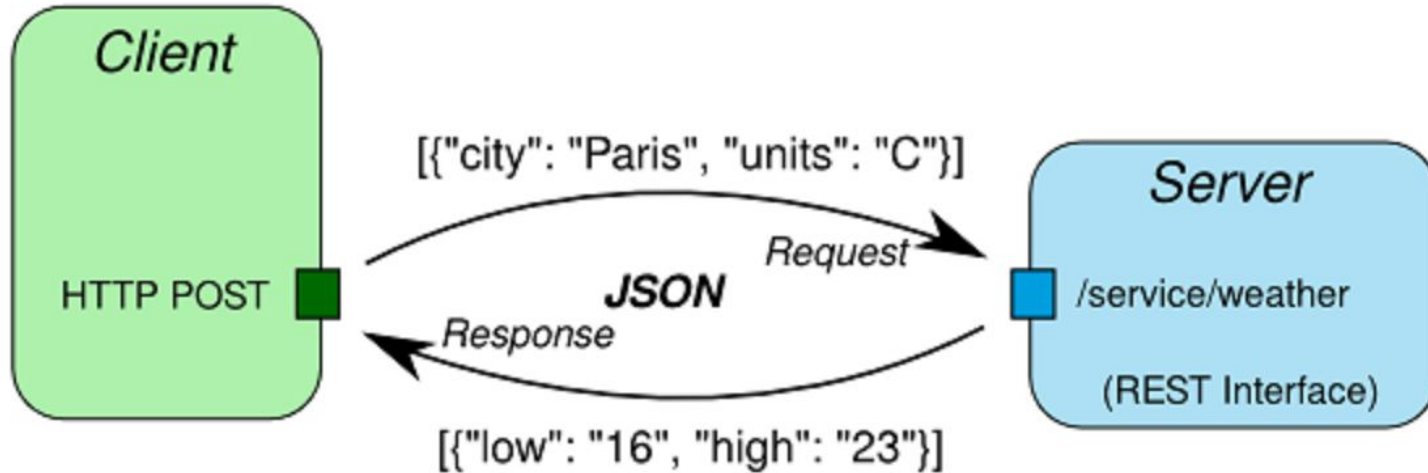
REST (Representational State Transfer) é um estilo de arquitetura baseado na transferência de representações de recursos de um servidor para um cliente.

Esse estilo reforça a web como um todo e é mais simples que o SOAP/WSDL para implementação de web services.

Serviços **RESTful** envolvem uma menor sobrecarga do que os chamados “web services grandes” e são usados por muitas organizações que executam serviços baseados em sistemas que não dependem de serviços fornecidos externamente.

Arquitetura em camadas: SOA

RESTful Web Service in Java



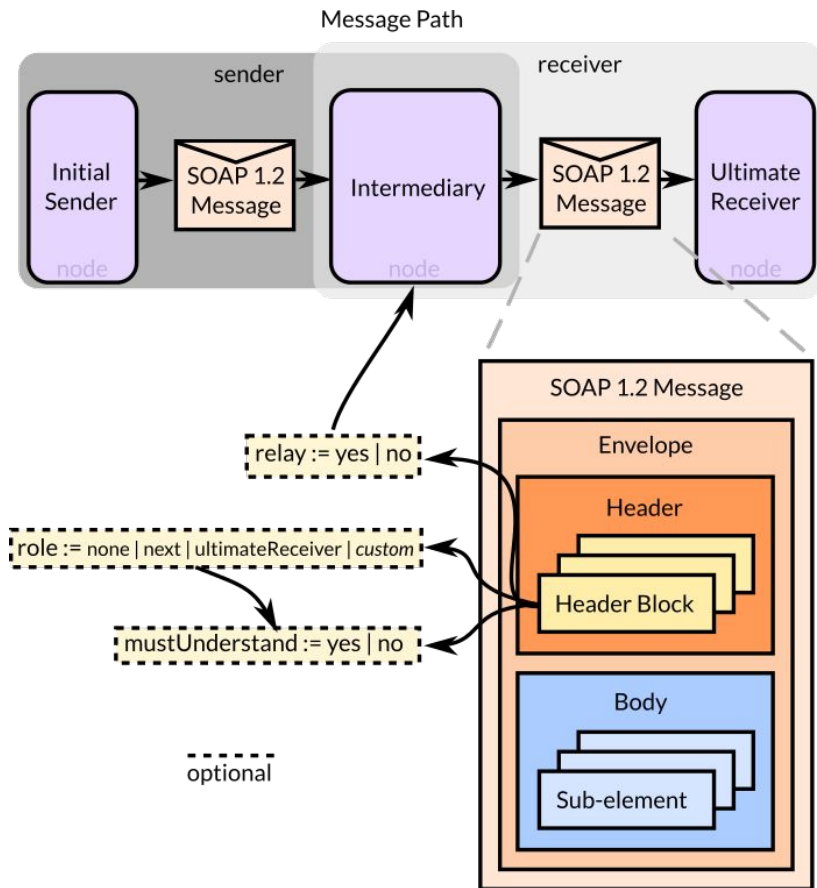
Arquitetura em camadas: SOA

SOAP é uma especificação baseada em XML para serviços da web. A especificação **SOAP 1.2** tem quatro dimensões:

- **Envelope:** divide uma mensagem em partes lógicas – cabeçalho e corpo.
- **Regras de codificação:** o mecanismo de serialização e a utilização de tipos de dados.
- **Representação RPC:** uma convenção para representar chamadas e respostas de procedimentos remotos.
- **Ligação SOAP:** a convenção para troca de envelopes entre pares usando um protocolo específico.

Arquitetura em camadas: SOA

SOAP



Arquitetura em camadas: SOA

SOAP define uma estrutura de mensagem em termos de um envelope. Um envelope divide uma mensagem em partes lógicas: um cabeçalho e um corpo

```
<env:Envelope xmlns:env="http://www.w3.org/2001/06/soap-envelope">
  <env:Header>
    <h:userDetails xmlns:h="http://www.tesira.com/userDetails">
      <h:IP>192.168.1.34</h:IP>
      <h:username>john.smith</h:username>
    </h:userDetails>
  </env:Header>
  <env:Body>
    <e:exchangeRateRequest xmlns:e="http://www.tesira.com/ws/ExchangeRateServi
    <e:quantity>5.20</e:quantity>
    <e:from>USD</e:from>
    <e:to>GBP</e:to>
    </e:exchangeRateRequest>
  </env:Body>
</env:Envelope>
```

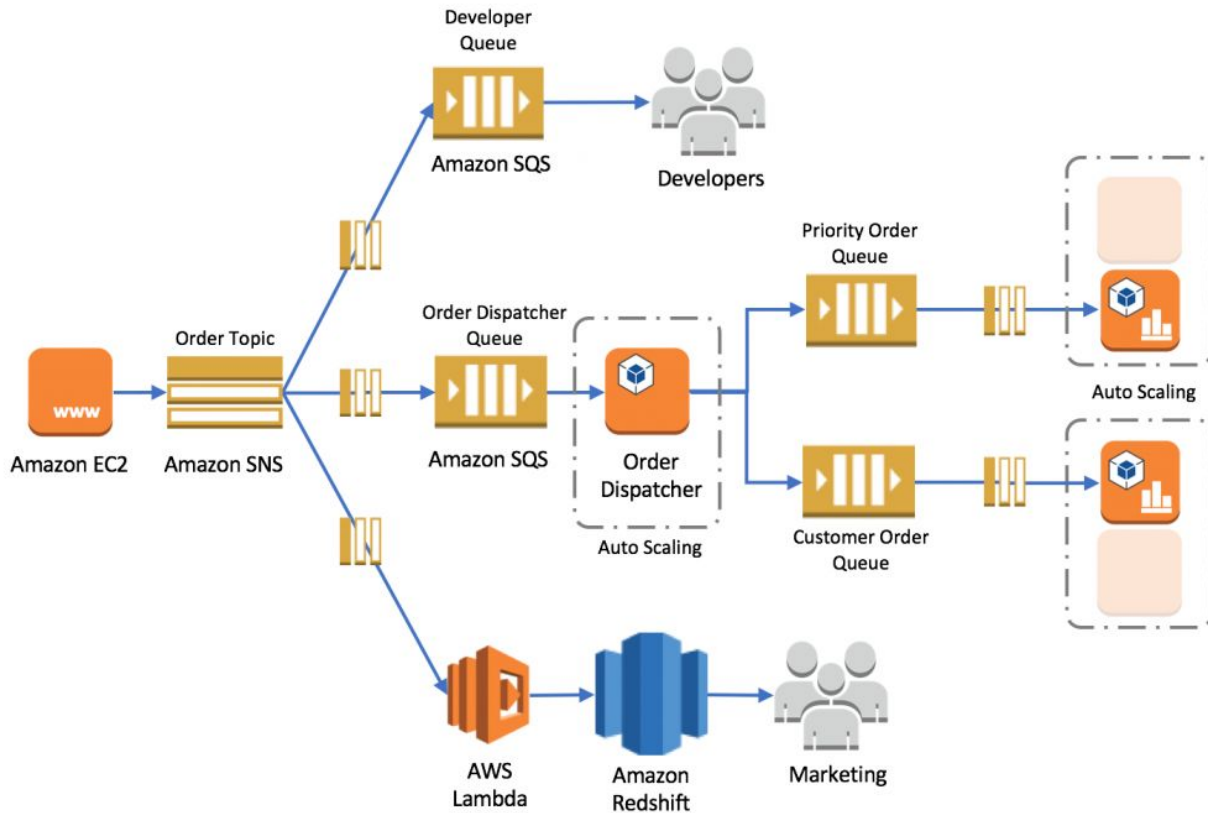
O cabeçalho declara metadados específicos do aplicativo, mas também pode conter atributos definidos pelo SOAP, como `role: mustUnderstand` e `relay`. Esses atributos determinam quem deve processar a mensagem e/ou o que deve ser feito com ela.

Arquitetura em camadas: SOA

- O conceito de serviços introduzido pela SOA agora é um aspecto central da **cloud computing** e da **virtualização**, sendo aplicado a elementos como o **middleware** e **microsserviços**.
- A arquitetura orientada a serviços usa **ESB (Enterprise Service Bus)**, enquanto microsserviços comunicam-se entre si de maneira stateless, por meio de APIs independentes de linguagem. Devido a esse aspecto das APIs nos microsserviços, as equipes de desenvolvimento também podem escolher as ferramentas com que preferem trabalhar.
- O barramento de serviços corporativos (ESB) é um padrão arquitetônico de software que oferece suporte ao intercâmbio de dados em tempo real entre aplicações diferentes.

Arquitetura em camadas: SOA

ESB



Arquitetura em camadas: SOA

Componentes da arquitetura **ESB**.

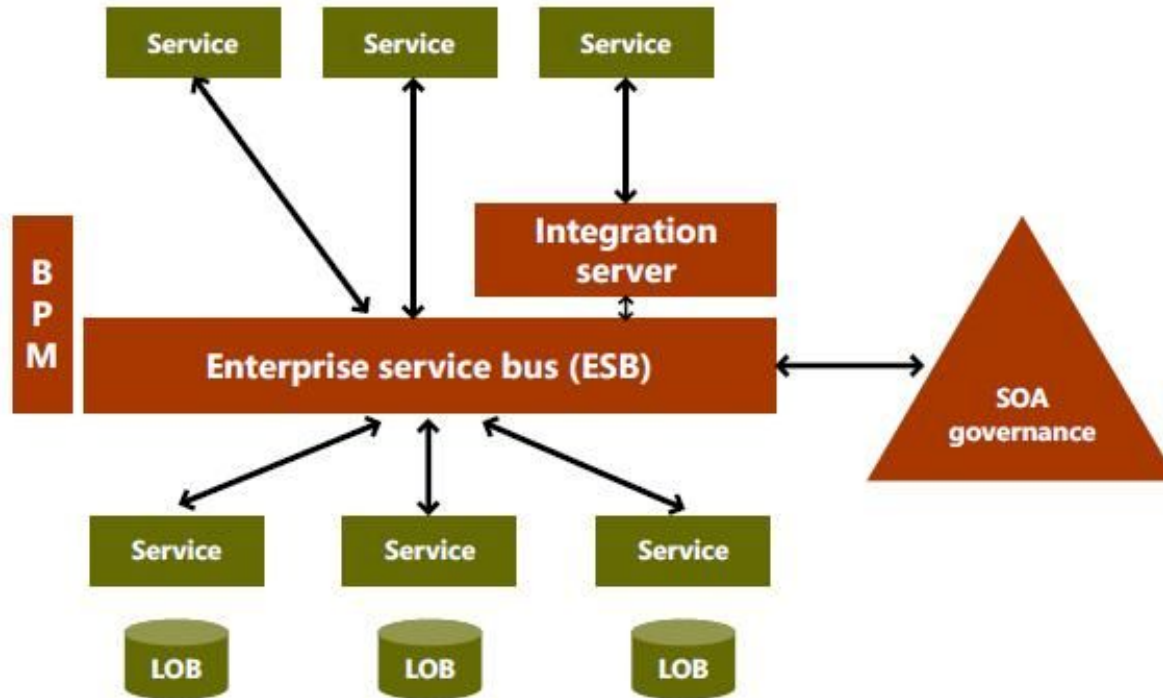
Endpoints: podem ser considerados pontos de entrada ou saída do ESB. tem um endereço ou identificador exclusivo. Você pode implementar endpoints usando várias tecnologias, como interface de serviço Web, filas de mensagens ou servidores FTP. Os endpoints também podem processar diferentes tipos de mensagens, como XML, JSON ou dados binários.

Adaptador: converte mensagens entre diferentes formatos e protocolos.

Barramento: Usa um conjunto de regras ou políticas com base em critérios descritivos, como tipo de mensagem, conteúdo ou destino, para encaminhar mensagens.

Arquitetura em camadas: SOA

SOA é diferente de **microserviços** e é diferente de **SaaS**



Arquitetura em camadas: SOA

Agilidade e produtividade do desenvolvedor: Qualquer componente pode ser modificado, testado e implementado independentemente dos demais, o que acelera os ciclos de iteração.

Escalabilidade: Qualquer componente pode ser escalonado independentemente dos outros.

Resiliência: graças ao desacoplamento, a falha de um microserviço não impacta os demais. E cada microserviço pode desempenhar conforme os seus próprios requisitos de disponibilidade sem comprometer os outros componentes ou todo o aplicativo aos maiores requisitos comuns de disponibilidade

Concluindo...

Arquitetura em camadas para aplicações Web

- Microserviços
 - SOA

MVC ~ MSC

Escalabilidade e modularização de projetos

Referências

Fowler, Martin. UML Essencial. 2o Edição. - Porto Alegre: Bookman, 2000.

Eckel, Bruce. Pensando em JAVA. Prentice-Hall, 2010

Deitel, Harvey. Java - Como programar. 6a Edição. - Porto Alegre: Bookman, 2006.