



## **Exercícios – Unidade 09**

**Valor: 5 pontos**

**Entrega: 22/11/20**

**Discentes:**

BEATRIZ CUNHA RODRIGUES	201776038
IVANYLSON HONÓRIO GONÇALVES	201776002
JOÃO PEDRO SEQUETO NASCIMENTO	201776022
RAFAEL FREESZ REZENDE CORRÊA	201435033

### **Parte I – Exercícios de Fixação**

- 1 – Por que é importante incluir no projeto do sistema que está em desenvolvimento um estilo e padrão arquitetural de software? Apresente 1 (um) exemplo de estilo e padrão de arquitetura que são compatíveis para desenvolvimento de um software.
- 2 – Em que situações é indicado elaborar o Diagrama de Implantação da UML para o desenvolvimento de um sistema?
- 3 – Em que situações é indicado elaborar o Diagrama de Componentes da UML para o desenvolvimento de um sistema?

### **Parte II – Trabalho Prático**

Nesta Unidade 9, deverão ser entregues como documento de requisitos do software a ser desenvolvido os seguintes itens:

**Diagrama de Implantação** que represente os componentes de hardware do sistema;

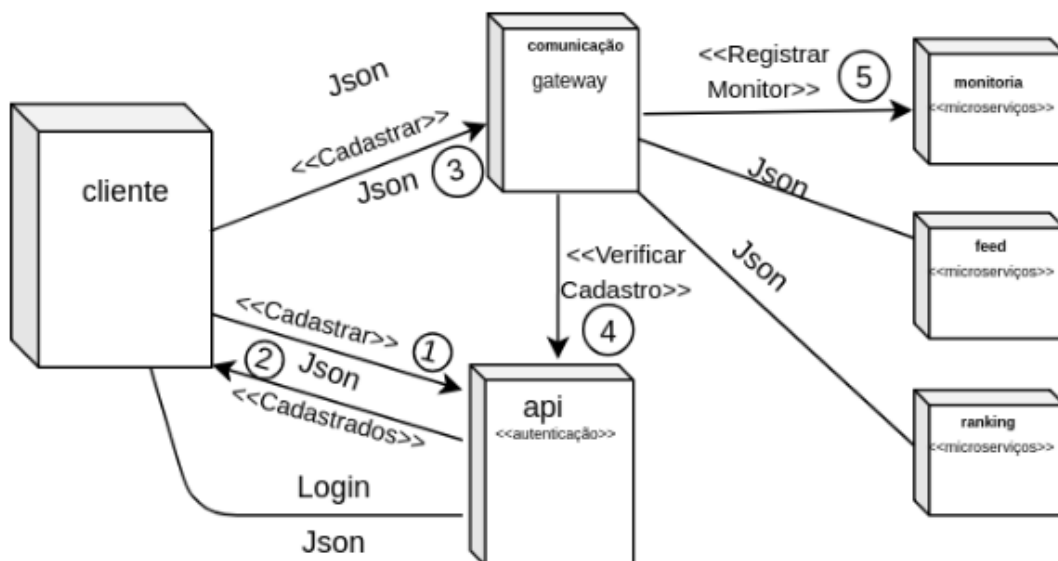
**Estilo e padrão de arquitetura de software** que serão utilizados para desenvolvimento do sistema. Justifique sua escolha.

Resposta:

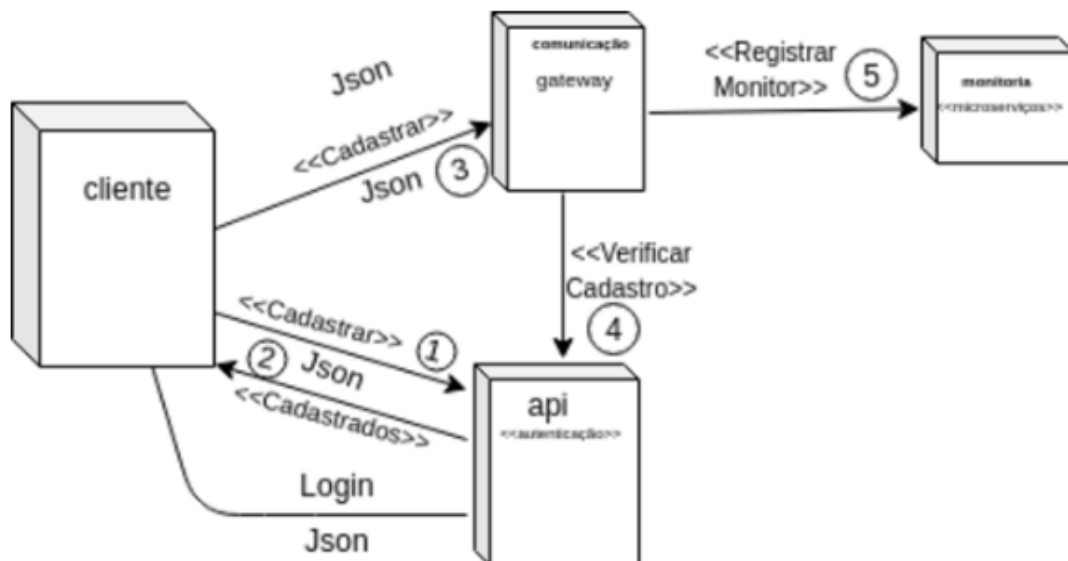
### Diagrama de implantação

Ele descreve a configuração do processamento de elementos e o mapeamento de componentes de implementação do software no decorrer do processamento dos elementos. O diagrama de implantação mostra a instância dos nós do processador, suas interconexões, instâncias dos componentes que executarão neles, e a dependência entre as instâncias.

Utilizando Json:



Versão 0.1



Versão 0.2

Passo 1: O usuário poderá cadastrar-se por meio de uma api de autenticação.

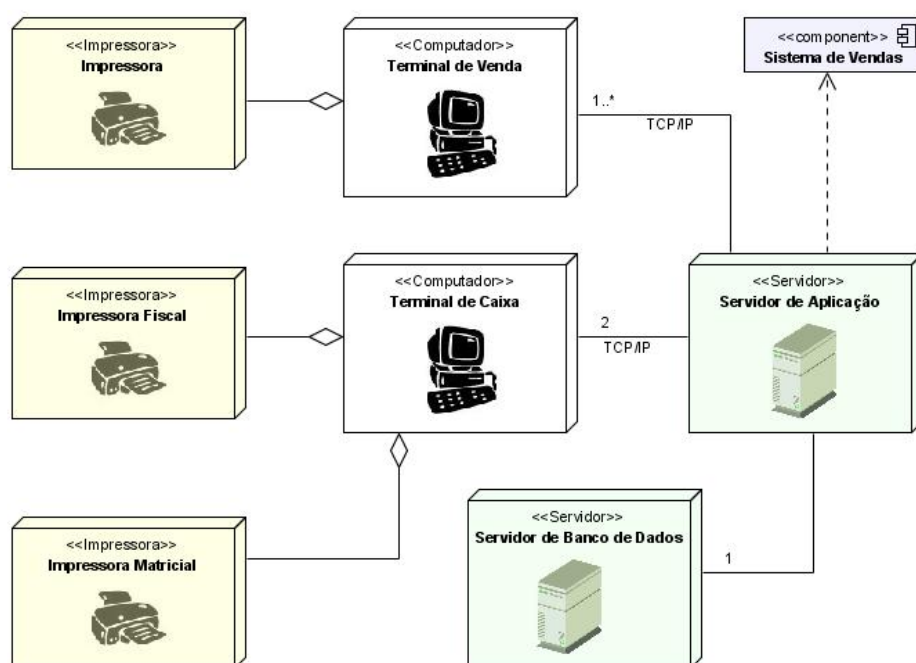
Passo 2: Retorna os usuários Cadastrados para login.

Passo 3: Envia os dados do cadastro para api de comunicação.

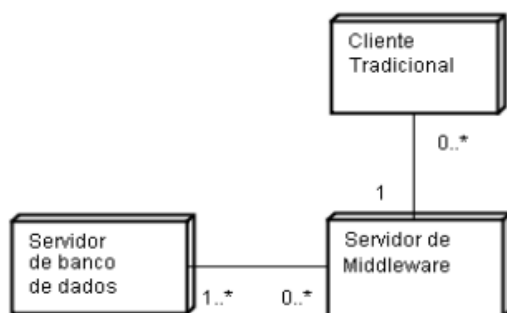
Passo 4: Verifica os dados recebidos do usuário com os dados que estão salvos na api de autenticação.

Passo 5: Registra em monitoria os dados do monitor.

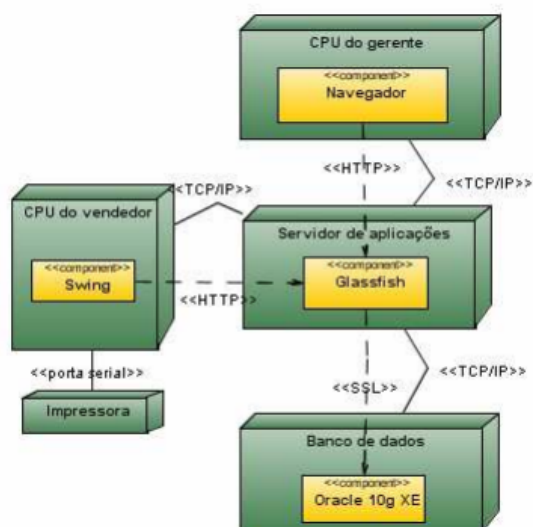
Na visão geral (Sistema venda - Loja Carioca Multimarcas):



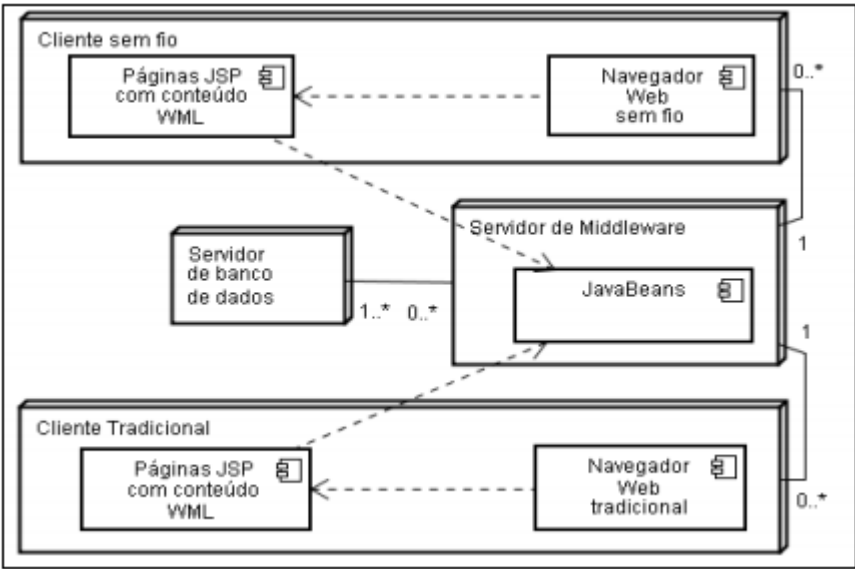
Na utilização do servidor:



E uma futura utilização web:



Planejamento para um upgrade para o sistema utilizando para web, acrescentando cliente:



**Estilo e padrão de arquitetura de software**

Estilo que vamos usar:

Abordagem	Foco	Padrões
Programação estruturada	Sistemas de pequeno porte	Estruturas de controle
Abstração e modularização	Sistemas de médio porte	Encapsulamento e ocultação de informações

Que terá a maioria destes estilos:

Estilo de Arquitetura	Descrição
Client-Server	Se grega o sistema em duas aplica,ão, onde o cliente faz uma requisição de serviço ao servidor.
Arquitetura baseada em Componentes	Decompõe o design da aplicação em componentes lógicos e funcioais que são independentes de local e expõe interfaces de comunicação bem definidas.
Arquitetura em Camadas	Separa as respnsabilidades da aplicação em grupos bem definidos (camadas).
Message-Bus	Um sistema de software que pode receber e enviar mensagens baseado em um conjunto de formatos conhecidos, de forma que sistemas possam se comunicar uns com os outros sem necessidade de conhecer o destinatário atual.
N-tier / 3-tier	Separa funcionalidade em segmentos de forma muito similar ao estilo de camadas, mas com cada segmento sendo uma camada localizado fisicamente em um computador separado.
Orientado a Objetos	Um estilo arquitetural baseado na divisão de tarefas para uma aplicação ou sistema em reutilização individual e objetos auto-suficientes, cada um contendo os dados e comportamentos relevantes ao objeto.
Apresentação Separada	Separa a lógica paragerenciar a interação do usuário da visualização da interface de usuário (UI) e dos dados com os quais o usuário trabalha.
Service-Oriented Architecture (SOA)	Refere-se a aplicações que expõe e consome funcionalidades como um serviço usando contratos e mensagens.

Cada um desses estilos são aplicados a áreas específicas de interesse:

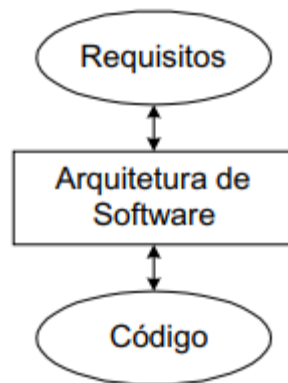
<b>Categoria</b>	<b>Estilos de Arquitetura</b>
<i>COMunicação</i>	Service-Oriented Architecture (SOA), Message Bus, Pipes e Filtros
<i>Deployment</i>	Client/server, 3-Tier, N-Tier
<i>Domain</i>	Domain Model, Gateway
<i>Interação</i>	Apresentação Separada
<i>Estrutura</i>	Baseado em Componentes, Orientado a Objetos, Arquitetura em Camadas

Para o futuro:

Componentes e conectores	Sistemas de grande porte	Estilos arquiteturais
--------------------------	--------------------------	-----------------------

Padrão de arquitetura de software MVC.

E alguns requisitos importantes sobre o sistema Carioca Multimarcas como portabilidade, escalabilidade, eficiência, corretude e interoperabilidade podem ser atingidos com a ajuda de um bom desenho arquitetural. A fase de definição da arquitetura de um software serve basicamente como uma ponte entre a fase de especificação dos requisitos do software e a fase de codificação, como ilustrado abaixo:



Definição da arquitetura que estamos utilizando

a) Tubos e Filtros

Cada componente do sistema tem um conjunto de entradas e um conjunto de saídas. Os componentes são organizados de forma que as saídas de um componente são conectados às entradas de outros.

b) Invocação implícita baseada em evento

A ocorrência de um determinado evento é anunciada pelo sistema a todos os componentes que se registraram para este evento. Desta forma, os componentes que recebem a notificação executam então as tarefas determinadas para a ocorrência deste evento.

c) Sistemas em camadas

Os componentes são organizados em forma de camadas. Cada camada fornece serviços à camada superior e solicitar serviços à camada inferior.

#### d) Repositórios

Esta arquitetura é composta basicamente de duas entidades: uma estrutura central que representa o estado do sistema e um conjunto de componentes independentes que manipulam esta estrutura. A arquitetura blackboard é um exemplo desta arquitetura, onde o quadro negro representa a estrutura de dados central.

#### e) Cliente/Servidor (Futuro web, mas terá que ter um servidor local)

Apresenta os componentes classificados em dois grupos: servidores e clientes. Os componentes servidores oferecem serviços que são requisitados pelos componentes clientes. Esta é a arquitetura mais utilizada por aplicações Web.

#### \*\*\*Camada de apresentação:

Esta camada é responsável pelo gerenciamento das solicitações/respostas dos clientes feitas através da rede e respectivas interfaces gráficas. A camada de apresentação é composta por um conjunto de Servlets e páginas JSP. As páginas Web que fazem a interface entre cliente e sistema são todas implantadas através de páginas JSP. Cada solicitação é redirecionada pelo servidor Tomcat a uma entidade receptora, que pode ser uma página JSP ou um servlet. Caso a solicitação envolva passagem de dados, esta entidade é responsável pela consistência dos dados recebidos. Neste caso, se os dados estiverem com problemas, beans com mensagens de erro são anexados ao objeto implícito request e a solicitação é imediatamente redirecionada para uma página JSP que é encarregada de administrar uma resposta ao cliente. Caso contrário, a entidade receptora “visita” os ambientes definidos na camada de lógica da aplicação e utiliza os recursos neles disponíveis para criar os beans que serão utilizados para responder à solicitação do cliente. Em seguida, anexa os beans aos objetos implícitos correspondentes e redireciona a solicitação para uma página JSP. Esta, por sua vez, faz a inclusão dos dados contidos nos beans anexados à uma estrutura HTML e envia a resposta ao usuário.

#### \*\*\*Camada de lógica da aplicação:

Esta camada contém a estrutura lógica da aplicação e é refinada em duas subcamadas. São elas:

##### a) Sub-camada de ambientes

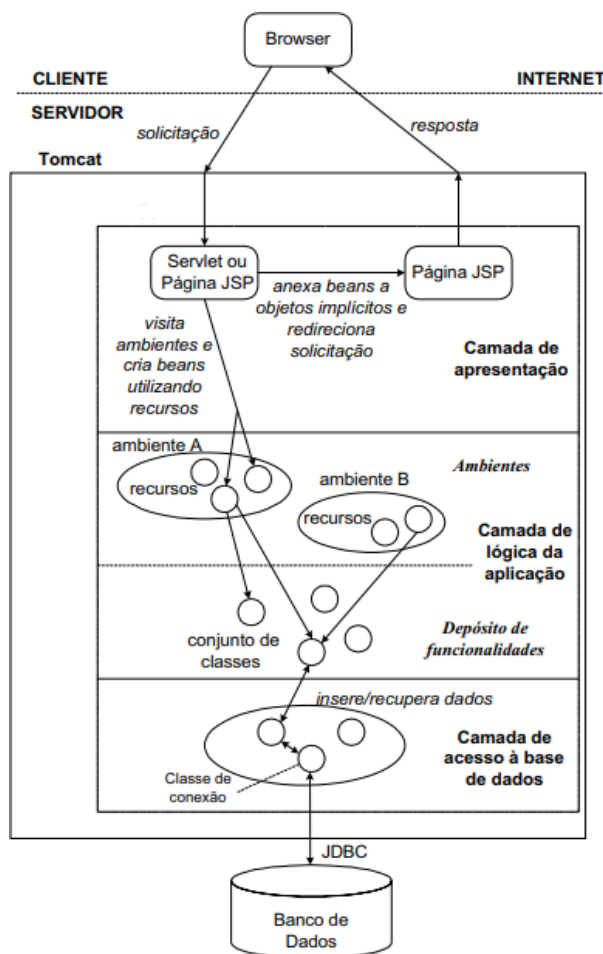
Esta sub-camada é composta por classes que representam ambientes definidos o mais próximo possível dos ambientes reais do estabelecimento. Cada ambiente oferece recursos específicos. Estes recursos são disponibilizados na forma de atributos fixos que são utilizados na apresentação da visão ao usuário, por exemplo, o nome do ambiente em que se encontra irá compor o título da

página JSP, ou na forma de funcionalidades oferecidas dentro de cada ambiente, por exemplo, a lista dos pedidos feitos pelo usuário.

b) Sub-camada de depósito de funcionalidades Composta por classes que realizam as funcionalidades exigidas pelos recursos disponíveis nos ambientes. Caso seja necessário fazer algum acesso ao banco de dados, é solicitado o serviço da camada de acesso à base de dados para recuperar ou inserir informações persistentes.

\*\*\*Camada de acesso à base de dados

Esta camada é responsável pelo gerenciamento de acessos à base de dados. Estes acessos são feitos utilizando uma interface padrão de comunicação com um banco de dados relacional chamada JDBC (Java Database Connectivity). Para cada tabela da base de dados existe uma classe nesta camada responsável por construir os comandos SQL para deleção, inserção e modificações de informações. Estas classes utilizam uma conexão comum à todos os acessos administrada por um único objeto que obedece o padrão singleton



Visão geral da arquitetura do sistema Carioca Multimarcas Web, mas terá local também.