

**A organização lógica de um sistema distribuído em componentes de software é referenciada como sua "arquitetura de software". Dois conceitos centrais aparecem aqui: o de "componente" e o de "conector". Descreva esses dois conceitos.**

O componente é uma unidade substituível em seu sistema, que requer e provém interfaces bem definidas. O conector é um mecanismo mediador da comunicação, coordenação e cooperação entre os componentes. Ele permite que os componentes funcionem em determinada ordem e que dados entre eles sejam trocados.

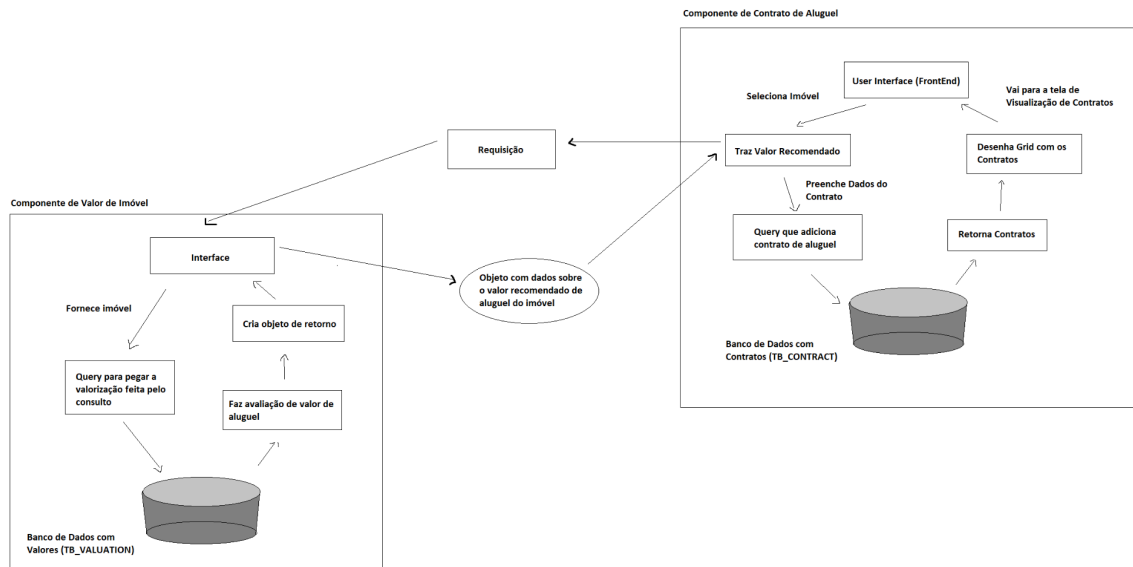
**Escolha uma aplicação distribuída que você conhece ou usa e esboce sua possível "arquitetura de software" em termos de componentes de software e conectores (use um ou mais estilos arquiteturais apresentados no texto como referência básica). Inclua comentários justificando suas escolhas.**

A aplicação distribuída se baseia no controle de Imóveis, sendo portanto uma arquitetura de software em camadas. Temos um componente responsável pelo contrato de aluguéis, então toda a parte de criação e edição de contratos de aluguéis faz parte da função dele.

Por mais que os contratos de aluguéis possam ter o valor preenchido na mão, convém ter um componente para recomendar um bom valor de aluguel. Para isso, temos outro componente, que possui dados sobre os valores do imóveis e quais seriam os valores recomendados de aluguel de acordo com o tamanho, arquitetura e localização do imóvel.

Pensando na arquitetura, temos a arquitetura em camadas, visto que a requisição parte de uma interface com o usuário e desce para o nível de processamento e para o nível de banco de dados, onde os contratos e os valores dos imóveis estão alugados. Ao mesmo tempo, a requisição feita na hora de criar um contrato para saber qual seria o valor recomendado retorna um objeto. Sempre que o componente de valores de imóveis for requisitado, ele retornará um objeto contendo os dados solicitados. Representando os dados contidos no bd de forma crua, ou processados na camada de processamento.

Essa troca de dados é feita pelo conector entre os componentes, por meio de procedure calls referente ao componente de valores de imóveis.



A instaciação de uma arquitetura de software em máquinas reais define a "arquitetura de sistema" da aplicação. Essa instanciação define como os componentes de software serão organizados e de que forma deverão interagir. Descreva as características centrais das três formas de organização: centralizada, descentralizada e híbrida.

**Centralizada:** Normalmente com servidor e cliente. O servidor é responsável por boa parte dos componentes enquanto que os clientes acessam remotamente suas funcionalidades

**Descentralizada:** Os processos da arquitetura são iguais, sendo clientes e servidores ao mesmo tempo. Logo, as funcionalidades são representadas por todos os processos existentes no sistema.

**Híbrida:** São uma combinação das arquiteturas centralizada e descentralizada. Normalmente a entrada/requisição é feita em um dos servidores, e então temos uma colaboração descentralizada entre os processos para atuar na requisição.

**Voltando à aplicação escolhida na questão 2, esboce sua possível arquitetura de sistema. Inclua comentários justificando suas escolhas.**

Na arquitetura escolhida, já há indícios de qual seria a melhor arquitetura de sistema. Naturalmente, pela arquitetura de software ser fortemente baseada em camadas, teremos uma arquitetura de sistema centralizada. Então, teremos o servidor com os componentes, e o cliente fará a requisição de consultar ou criar contratos de locação.

Usando o esboço do livro, vemos que o cliente faz a requisição da operação, o servidor faz a chamada dos dados de acordo com a operação requirida. Então, o

servidor de dados retorna os dados solicitados. A partir deles, a aplicação pode processá-los a fim de retorná-los ao cliente, que está esperando pela resposta.

Caso precisássemos de mais poder de processamento, poderíamos pensar em uma arquitetura híbrida, dividindo a requisição em processos a serem executados por máquinas distintas.

No entanto, no caso aqui considerado mais simples levamos em consideração apenas a arquitetura centralizada. Nela, podemos ter o cliente, a aplicação e a base de dados numa mesma máquina, assim como em máquinas diferentes. Os processos serão passados de uma máquina para a outra, no processo de camadas sempre esperando a resposta do nível inferior.

