

## ESPECIFICAÇÃO DE TRABALHO PRÁTICO

**Objetivo Geral:** este trabalho prático tem como objetivo a análise dos aspectos de projeto arquitetural, projeto detalhado, implementação e implantação centrados em arquiteturas de *software*, aplicados no contexto de um sistema moderno, com demandas por requisitos tais como: eficiência, facilidade de evolução, gerência de complexidade, escalabilidade, suporte à heterogeneidade, auto-gerenciamento e adaptação em *run-time*, *dependability* e previsibilidade temporal. O aluno deverá ser capaz de demonstrar senso crítico nas análises das decisões arquiteturais, bem como os benefícios, justificativas e consequências dessas decisões.

**Formato e Produtos a serem Gerados:** o trabalho prático será realizado em dupla (entretanto, com notas individuais) e consiste na entrega de um artigo de apresentação da análise arquitetural realizada e na apresentação oral (30 minutos) dos resultados obtidos. Os tópicos a serem apresentados no artigo e na apresentação oral são descritos a seguir, acompanhados de uma descrição do que espera-se que seja apresentado em cada tópico e sugestões para a sua correta realização. O artigo deve seguir o formato padrão para conferências definido pela Sociedade Brasileira de Computação (SBC) e disponível para *download* no seu site.

**Objetivos Específicos:** o trabalho deve conter uma apresentação geral do sistema em questão, com suas principais demandas funcionais e os requisitos não-funcionais característicos que guiaram as principais decisões arquiteturais. Em seguida o artigo apresentará modelos para as duas *views* mais características (importantes) do sistema, obtidas a partir de dois dos seguintes *viewpoints*: *viewpoint* lógico (estrutural/componente-conector - **obrigatório**), *viewpoint* de implantação (quando não-trivial), *viewpoint* de concorrência (quando significativo) e *viewpoint* comportamental. Estes dois modelos devem estar descritos na notação de modelagem (linguagem natural, gráficos informais, UML ou ADL's) mais adequada para o sistema em questão, observando sua relação custo-benefício. O próximo passo consiste na análise de um sub-sistema representativo da aplicação (aquele com as funcionalidades mais críticas de acordo com os requisitos não-funcionais) e na realização do projeto detalhado de classes. Deve ser indicado como as decisões arquiteturais utilizadas na *view* estrutural induziram e favoreceram o atendimento das propriedades funcionais e não-funcionais desejadas. O projeto finaliza com uma análise da implementação, indicando os possíveis *architecture-implementation frameworks*, soluções de *middleware* e componentes e conectores COTS (*Commercial Off-The-Shelf*) utilizados.

### Dicas Gerais:

- O artigo e os *slides* da apresentação não devem conter erros gramaticais. Utilize um corretor ortográfico e revise o material antes da apresentação. O material deve evidenciar zelo por parte do autor.
- Não dedique muito espaço no texto (e principalmente tempo na apresentação oral) para conceitos, definições e aplicações já vistos em sala de aula. Uma boa apresentação deve trazer **novos** conhecimentos à turma. Uma boa proporção é 20% de fundamentação teórica e 80% de trabalho realizado.
- Termos em inglês devem estar em itálico (todos e em todas as ocorrências).
- Informe o significado de uma sigla na primeira vez que ela aparecer no texto.
- Tenha cuidado com a qualidade das figuras utilizadas. Todas as figuras, diagramas, tabelas e códigos devem ser citados e explicados no texto. As referências também devem aparecer no texto em algum momento.
- O conteúdo do trabalho deve ser apresentado em um nível de profundidade compatível com aquele utilizado nas aulas.
- **Toda** apresentação oral deve ter uma conclusão. Não interrompa sua apresentação abruptamente, mas também não gaste cinco minutos nas conclusões.
- O tempo total disponível para a apresentação não deve ser excedido. Recomenda-se, entretanto, fazer uso de todo o tempo disponível. Pratique !
- Utilize com sabedoria os 30 minutos da sua apresentação. Não gaste muito tempo com introduções demasiadamente longas ou aspectos não relevantes ao seu trabalho. Vá direto ao assunto, o que você fez é o mais importante. Uma boa

relação é: cinco minutos para a introdução, vinte para o que foi desenvolvido e cinco para a conclusão.

**Roteiro do Artigo:** a seguir serão descritos os tópicos a serem desenvolvidos no artigo e na apresentação oral, acompanhados de uma descrição dos resultados esperados.

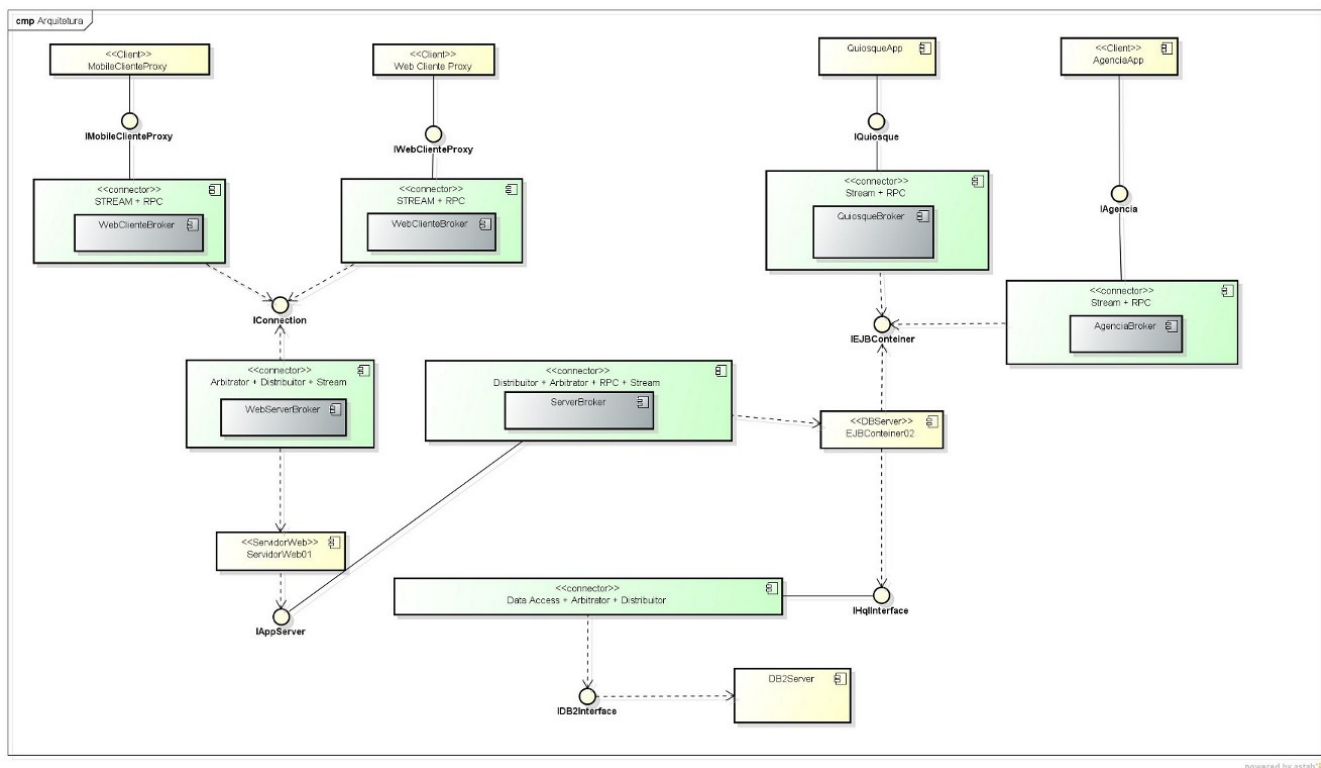
1) **Introdução:** apresenta uma visão geral do **artigo** (não do sistema): breve descrição do sistema, qual foi o trabalho realizado e as análises realizadas. O último parágrafo da introdução deve apresentar a estrutura restante do artigo.

**2) O Sistema Utilizado** (utilize um título descritivo do seu estudo de caso): esta seção apresenta os objetivos do sistema, aspectos importantes do seu ambiente de execução (possíveis integrações com outros sistemas, etc) e os requisitos funcionais e não-funcionais em questão. A seção deve deixar claro quais aspectos funcionais e não-funcionais foram corretamente atendidos pela arquitetura analisada.

3) **Projeto Arquitetural:** aqui serão apresentadas as duas *views* derivadas do projeto: estrutural (componente-conector) e uma segunda *view*, a ser escolhida pela dupla executora do trabalho. Para a *view* estrutural os seguintes aspectos devem estar **claramente** descritos: possíveis estilos ou padrões arquiteturais utilizados (isolados ou em conjunto/híbridos), principais componentes com seus respectivos serviços providos e requeridos, principais conectores utilizados (para cada um descrever os conectores básicos utilizados e os valores aplicados nas suas dimensões de variação) e a configuração final obtida. Para cada um destes aspectos é imprescindível explicar qual é o *rationale*.

## Exemplos de visões arquiteturais

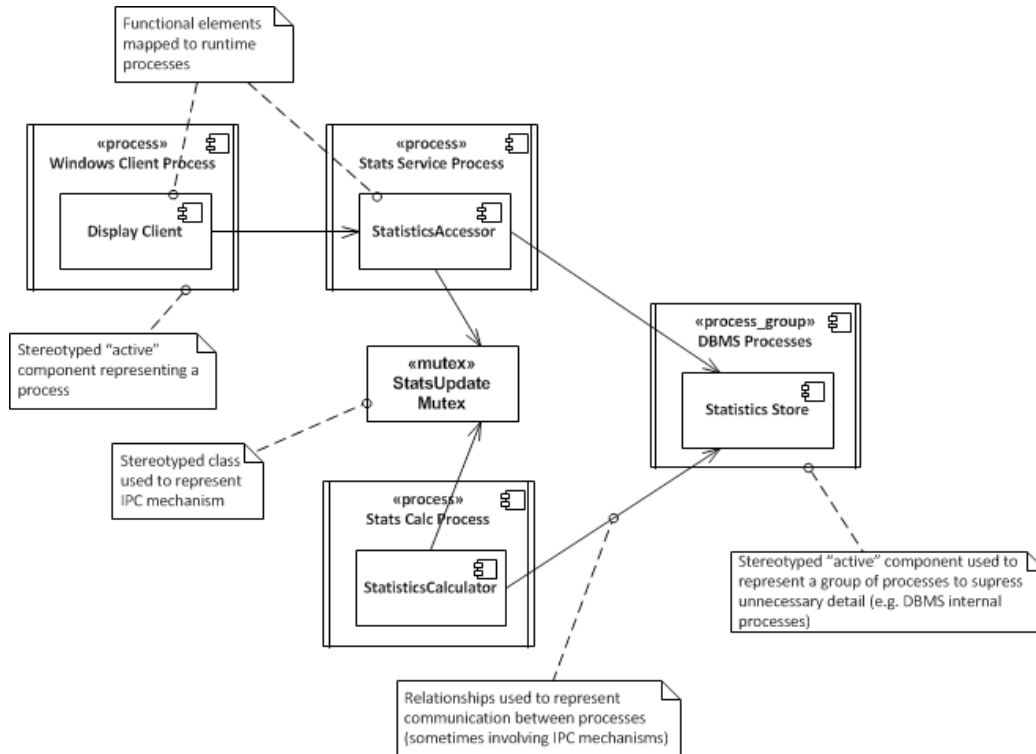
**Visão Estrutural (Componente-Conector).** Define os principais módulos (componentes) do sistema e os conectores utilizados na interação entre estes componentes. A figura a apresenta um exemplo de modelo estrutural da visão componente-conector.



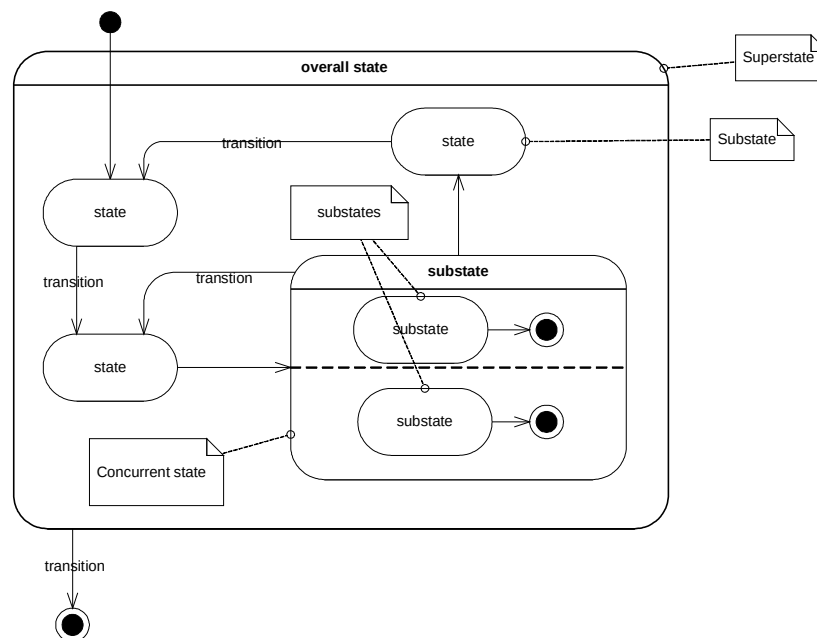
**Figura 1:** exemplo de modelo estrutural

**Visão de Concorrência.** Define o conteúdo de elementos de *runtime* do sistema (por exemplo, processos do sistema operacional) que empacotarão os elementos funcionais do sistema. Esta visão é geralmente formada por dois modelos: modelo

de concorrência e modelo de estado. As figuras 2 e 3 ilustram estes modelos.



**Figura 2:** exemplo de modelo de concorrência



**Figura 3:** exemplo de modelo de estado

**Visão de Implantação.** Define as características importantes do ambiente operacional de implantação do sistema. Esta visão inclui detalhes sobre os nós de processamento que o sistema requer para a sua instalação (por exemplo, os requisitos mínimos de memória e processamento), as dependências de *software* em cada nó (tais como bibliotecas e plataformas de *middleware* necessárias) e detalhes sobre a rede subjacente. A figura 4 apresenta um exemplo de um modelo de implantação.

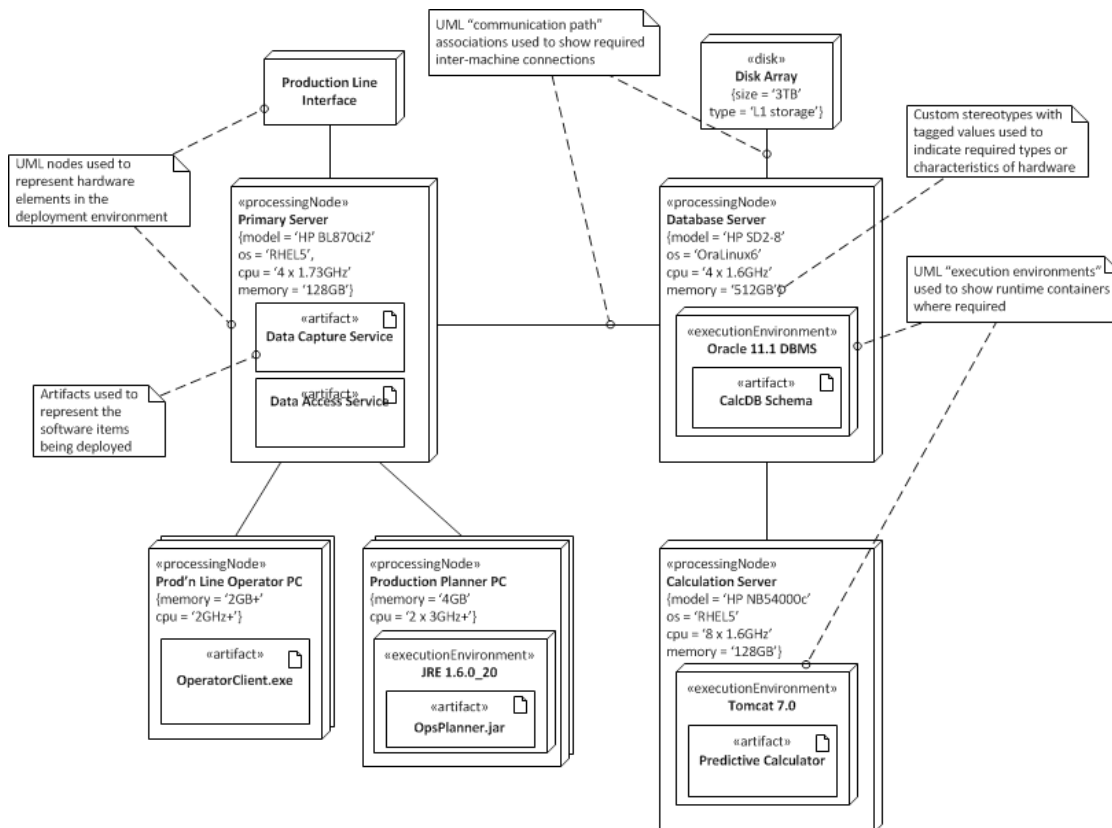


Figura 4: exemplo de modelo de implantação

**4) Projeto Detalhado:** o sub-sistema mais característico (importante) apresentado na *view* estrutural deve ser selecionado para a análise do projeto detalhado. Neste projeto, eventuais *design patterns* utilizados deverão ser explicados de forma criteriosa. A seção deve apresentar os *design patterns* utilizados, as consequências da sua utilização, o diagrama de classes final da solução e o *rationale* subjacente.

**5) Implementação e Implantação:** esta seção apresenta como foi realizado o mapeamento do modelo arquitetural nos artefatos de implementação. Todas as tecnologias facilitadores deste mapeamento, tais como *architecture-implementation frameworks* (MVC, *command-pattern*), soluções de *middleware* (EJB, CORBA, .NET, etc) e componentes e conectores COTS (*Commercial Off-The-Shelf*), devem ser apresentadas e justificadas. Os aspectos complexos do processo de implantação devem também estar presentes nesta seção, justificando as decisões de mapeamento de componentes e conectores em *hosts*.

**6) Discussão e Conclusões:** nesta seção o projeto deve ser discutido e soluções alternativas sugeridas. Os pontos fortes e fracos devem ser analisados a partir de uma visão comparativa crítica. É preciso **discutir**, ou seja, não é suficiente somente repetir a introdução com outras palavras :)

**Datas de Entrega:**

03/05/2012 - 1a entrega: visão geral do sistema + descrição dos principais módulos

14/06/2012 - 2a entrega: duas visões arquiteturais + início do detalhamento de um dos módulos

26/07/2012 - 3a entrega: detalhamento completo do módulo + apresentação

**Temas:**

- Struts 2
- EJB 3
- Netbeans
- Qt Creator
- KDevelop
- KDE Plasma

**Observações:**

- O artigo deve ser entregue impresso no momento da apresentação.
- Não haverá mudança na data da apresentação.
- Para eventuais dúvidas contacte o professor: [sandroandrade@ifba.edu.br](mailto:sandroandrade@ifba.edu.br)

Boa sorte !