



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARAÍBA

Projeto Arquitetural

Prof. Ricardo Job

Contato: ricardo.job@ifpb.edu.br

Definição

- Consiste em se buscar uma solução para uma tecnologia específica para a satisfação dos requisitos levantados, uma busca da uma melhor configuração, onde se determinam quais processos rodarão, em quais processadores, onde serão armazenados e qual a comunicação necessária.

Definição

- O Projeto de arquitetura pode ser expresso através de diagramas de bloco apresentando uma visão geral da estrutura do sistema, podendo representar de uma forma geral o domínio da aplicação.

Definição

- Quando nos referimos a um projeto arquitetural é importante ressaltar algumas decisões, pois as mesmas são tomadas para representar, modelar e realizar profundas mudanças no modelo de domínio do software.

Definição

- Dentre as principais decisões, podemos destacar:
 - Módulos: Modularização do projeto em subsistemas.
 - Subsistema: Escolha de uma estrutura de comunicação e controle entre os subsistemas.
 - Comunicação: Definição das interfaces entre subsistemas.
 - Persistência: Escolha de uma estratégia de persistência.
 - Requisitos não-funcionais: Atendimento a requisitos especiais de desempenho e outros requisitos.
 - API de Comunicação: Exposição das interfaces para facilitar a futura integração de aplicações.

Modelar faz bem?

- Estrutura, controle e decomposição modular podem ser baseados num modelo ou estilo de arquitetura particular. Contudo, a maioria dos sistemas é heterogênea. Alguns domínios de aplicação possuem modelos específicos. Onde os modelos podem refletir em:
 - Desempenho.
 - Robustez .
 - Distributividade.
 - Manutenibilidade.

Processo de projeto de arquitetura

- Estruturação do sistema: O sistema é decomposto em vários subsistemas principais e as comunicações entre eles são identificadas.
- Modelagem de controle: Um modelo de relacionamento de controle entre as diferentes partes do sistema é estabelecido.
- Decomposição modular: Os subsistemas identificados são decompostos em módulos.



Certo...muitas
perguntas surgem.....
e como fazer o projeto?

Técnicas utilizadas

- Algumas técnicas são comumente observadas e usadas para desenvolver um software, sendo algumas fundamentais para o estudo:
 - Abstração: Característica essencial de um módulo que o diferencia de qualquer outro módulo, provendo uma delimitação conceitual clara sobre ele.
 - Encapsulamento: Agrupamento dos elementos de uma abstração incluindo elementos estruturais e comportamentais. O encapsulamento provê barreiras claras entre abstrações. Logo, os detalhes que não são necessários para o uso de uma abstração devem ser escondidos.

Técnicas utilizadas

- Separação de Interesses: Responsabilidades diferentes e não relacionadas devem ser separadas num sistema.
- Separação de Interface e Implementação: Separa "o que" deve ser feito do "como é feito", diminuindo o acoplamento e permite trocar interfaces.

Técnicas utilizadas

- Modularização do Sistema: Decompor um sistema em subsistemas. Existem vários motivos que fazem com que seja interessante modularizar um sistema em subsistemas durante o projeto arquitetural:
 - Lida com a complexidade de sistemas.
 - Mantém a coesão de cada subsistema.
 - Diminui o acoplamento geral do sistema.
 - Permite escolher a forma e como será desenvolvido cada subsistema.
 - Permite reutilizar subsistemas em várias aplicações.

Técnicas utilizadas

○ Subsistemas e Módulos

- Um subsistema é um sistema também cuja operação é independente dos serviços providos por outros subsistemas.
- Um módulo é um componente do sistema que provê serviços a outros componentes, mas que normalmente não seria considerado um sistema separado.



Então sobre os principais pontos vamos levantar alguns questionamentos.

Subsistemas

- Quais sistemas externos devem ser acessados?
- Como serão acessados?
- Há integração com o legado a ser feita? Como?
- Determinação de oportunidades para o reuso de software. Há interesse/conveniência/tempo em aproveitar tais oportunidades? Como isso pode ser feito?

Modularização e decomposição do sistema

- O sistema é centralizado ou distribuído?
- Como modularizar em subsistemas?
- Como minimizar acoplamento entre os módulos?
- Quais as camadas de cada subsistema?
- Onde colocar a lógica da aplicação?
- Qual é o nível de acoplamento, frequência de interações, volumes de dados trocados entre as camadas?
- Qual será a estrutura de comunicação e controle entre os subsistemas (Como ligar as camadas)?

Modularização e decomposição do sistema

- Quais subsistemas podem se comunicar com os outros?
- Quais são as interfaces importantes entre os subsistemas?
- Qual é o formato das mensagens?
- A programação será feita com qual paradigma?
- Que linguagens e ferramentas serão usadas?
- Que frameworks serão usados?
- Há considerações especiais de segurança que afetam o sistema?
- Como é a estratégia de tratamento de erros?

Comunicação

- O sistema será acessado usando que tipos de clientes? Browser? Uso de applet? Uso de script cliente?
- Como fazer a interface gráfica?
- Com que ferramentas?
- Com que componentes visuais? Serão desenvolvidos? Comprados?
- Javascript ou outra linguagem de script do lado cliente será usada?
- Se a interface usar um browser, como será feita a geração de HTML dinâmico? Servlets? JSP?
- Que ferramentas usar para a formatação de relatórios?

Lógica da aplicação

- Quais são os componentes principais a fazer?
- Há uso de threads?
- De que forma atender aos requisitos para uso multiusuário?
- Quais as soluções para resolver a concorrência?
- Que APIs serão usadas para acesso aos dados persistentes?
- Determinar quais são subcamadas e partições: Quais são as interfaces importantes entre os pedaços? Onde verificar os business rules? No SGBD? No middleware?
- Como implementar aspectos de segurança?
- Como implementar os requisitos de auditoria?

Persistência

- Quais são as fontes de dados? externas? internas? existentes? novas? Como acessá-las?
- Que estratégia de persistência será usada na memória (Com recursos especiais como bateria, memória flash)? Em arquivos? Usando um SGBD?
- Qual paradigma de SGBD usar? Relacional? OR? OO?
- Qual será a estratégia de interação entre a aplicação e os dados?
- De que forma atender aos requisitos para uso multiusuário?
- Para grandes escalas, como oferecer connection pooling?
- Como implementar a segurança no SGBD?

Requisitos não-funcionais

- Há necessidade de bolar formas especiais de atender aos requisitos de desempenho?
- Como prover escala?
- Quais são os módulos executáveis a produzir?
- Quais são os arquivos importantes de configuração e seu formato?
- Como será resolvida a instalação do produto?

API de comunicação

- Que interfaces devem ser expostas para facilitar a futura integração de aplicações ?
- Como será feita a exposição? Com XML? Através de uma API?

Mais algumas coisa?

- Como será o atendimento a outros requisitos (custo, mobilidade, outros)?
- Há considerações especiais de administração a levar em conta?
- Como será a troca de versões?
- Como será a distribuição do software? Via Internet/Intranet? Via mídia?
- Você já verificou se a arquitetura planejada pode atender a todos os requisitos levantados?

Referências Bibliográficas

- <http://jacques.dsc.ufcg.edu.br>
- Prof. Paulo Caetano. Notas de aula - Projeto Arquitetural
- Prof. Claudivan Cruz. Notas de aula - Projeto Arquitetural – IFPB Campus Patos.