Nome: André Sacilotto Santos

Data: 04/09/2022

1) O que está incluído no projeto arquitetural de um software?

Arquitetura é a organização fundamental de um sistema de software observado entre seus componentes. Estes componentes herdam esta organização através dos relacionamentos entre si e seu ambiente, a partir dos princípios que guiam seu design e evolução. Assim, o projeto arquitetural de software consiste em avaliar os requisitos que guiam estes fundamentos e organizar o projeto de forma a contentá-los. Dessa forma, é necessário ponderar entre as abstrações possíveis levando em consideração as consequências no funcionamento da implementação no contexto aplicado, fazendo-se utilizar das tecnologias que fornecem o melhor arcabouço para a estratégia escolhida. Em outras palavras, o projeto arquitetural de software envolve compreender os desafios propostos e adaptar a arquitetura da implementação, balanceando os componentes críticos entre possibilidade de concretização, necessidades principais e ciclo de vida saudável do produto.

2) Cite vantagens e desvantagens de uma arquitetura centralizada em comparação com uma arquitetura distribuída quando o foco é segurança.

A arquitetura centralizada fornece maior facilidade de projetar e construir código dado o simples acesso à informação, que fica inteiramente contida numa única estrutura. Porém, quando a segurança é quebrada toda a informação fica vulnerável. De forma contrária, se a informação estiver numa arquitetura distribuída, apenas uma parte da informação poderia ser acessada num ataque, mas em contrapartida também tornaria o acesso mais difícil em situações comuns no fluxo de dados.

3) Por que é importante minimizar a complexidade de um sistema de software?

É importante minimizar a complexidade de um sistema de software para mantermos o software fácil de ser refatorado, diminuindo a possibilidade de geração de erros ao adicionarmos novas funcionalidades. Dessa forma a manutebilidade do produto se torna mais simples de ser feita e por consequência, menos custosa.

4) Você está desenvolvendo um produto para uma empresa de finanças. Considere os desafios que afetam as escolhas arquiteturais e escolha dois que considera mais importantes. Explique sua decisão.

Um sistema de finanças é um alvo constante de ataques externos, devendo possuir grande resiliência e segurança dos dados. O sistema deve estar também sempre disponível, pois apenas um momento fora do ar pode trazer prejuízos enormes para a empresa. Estes são os requisitos mais importantes a serem considerados, embora a maioria das transações necessitem também de uma performance razoável, além de estarem disponíveis. Aleḿ disso a segurança dos dados não deve gerar uma experiência de usuário truncada ou com muitos passos de verificação, o que pode retirar o interesse do cliente e manter seus dados seguros ao mesmo tempo.

5) Explique brevemente por que estruturar uma arquitetura de software em camadas funcionais ajuda a minimizar a complexidade geral do sistema.

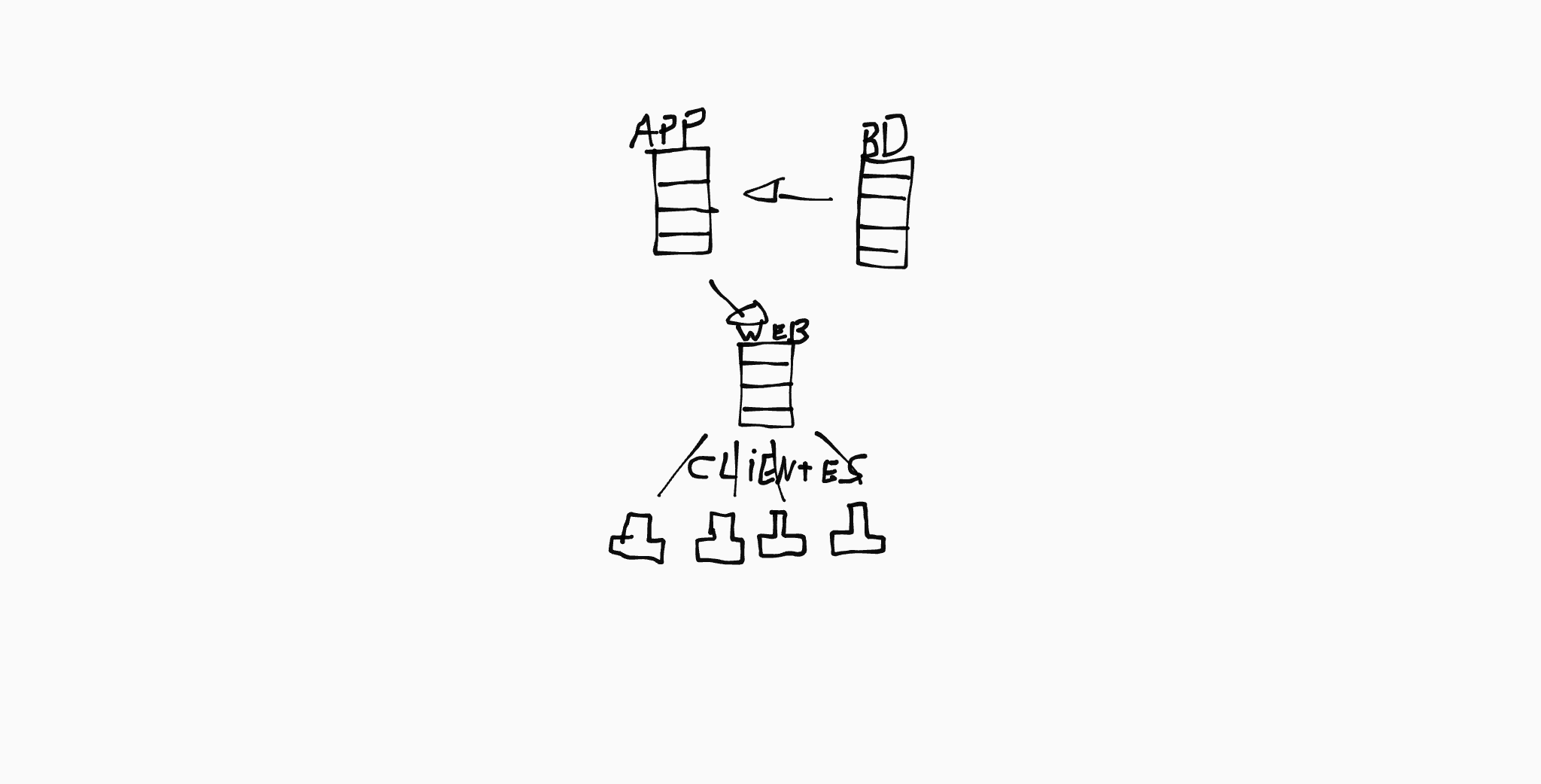
Projetos com arquiteturas orientadas a camadas ajudam a minimizar a complexidade geral do sistema por separar os componentes de forma independente, evitando sobreposição de funcionalidades. Dessa forma a arquitetura fica mais limpa e menos repetitiva, gerando clareza em cada componente e entre suas interações.

6) Imagine que seu gerente lhe perguntou se vale a pena começar a usar descrições mais formais da arquitetura baseadas em UML. Escreva um pequeno texto aconselhando-o.

O uso da documentação baseada em UML facilita a comunicação das dependências e funcionalidades previstas nos requisitos durante a implementação. Além disso, seu uso facilita a visualização do contexto geral do projeto e a relação dos componentes, facilitando a sua compreensão e até mesmo a otimização de seus relacionamentos durante a fase de prototipação. Portanto o uso de UML é altamente recomendado e pode produzir software de maior qualidade com menos devaneios referentes ao entendimento das especificações durante a implementação.

7) Desenhe um diagrama para mostrar como uma arquitetura genérica para um sistema WEB pode ser implementada usando uma arquitetura cliente-servidor com várias camadas físicas.

A apresentação é enviada para o servidor web a partir do servidor de aplicação junto dos dados necessários do servidor do banco de dados, que repassa a apresentação a todos os clientes que acessam o endereço através de seus respectivos navegadores. Dessa forma uma arquitetura cliente-servidor foi implementada com 4 camadas, embora fisicamente estas camadas se refiram à suas funcionalidades, podendo estar fisicamente acopladas em uma mesma máquina.



8) Em qual tipo de situação você colocaria a maior parte do processamento no cliente ao invés do servidor em uma arquitetura cliente-servidor?

Em situações onde não há a necessidade da consulta em nenhuma base de dados externa, pode ser útil manter a maioria do processamento no cliente. Por exemplo, no caso onde o usuário está interagindo com uma interface gráfica que responde apenas a suas entradas, necessitando de poucas requisições externas para mostrar os respectivos resultados.

9) Pesquise as principais diferenças entre bancos de dados relacionais e NoSQL. Sugira 3 tipos de produtos de software que podem se beneficiar do uso de bancos de dados NoSQL. Explique por que bancos NoSQL seriam adequados em cada caso

Os bancos de dados relacionais armazenam dados de acordo com esquemas específicos. Por outro lado, os sistemas NoSQL permitem que os dados sejam armazenados usando qualquer estrutura necessária, mas fornece uma maneira de atualizar esses dados ao alterar essa estrutura. Os bancos de dados relacionais são baseados em tabelas, enquanto os não relacionais podem se basear em qualquer tipo de documento, grafo, chave-valor e matrizes multidimensionais. Uma variedade de aplicativos modernos se beneficiam do uso de bancos NoSQL, tais como para dispositivos mǘeis, web e jogos. Os Bancos NoSQL se destacam nestes casos por sua flexibilidade, sendo ideais para dados semi estruturados e não estruturados, por sua escalabilidade, pois geralmente são projetados para serem escalados horizontalmente usando clusters distribuídos, por sua alta performance, por serem otimizados para modelos de dados específicos e padrões de acesso que permitem maior performance do que quando se tenta realizar uma funcionalidade semelhante com bancos de dados relacionais e por serem altamente funcionais, por serem criados especificamente para cada um de seus respectivos modelos de dados.