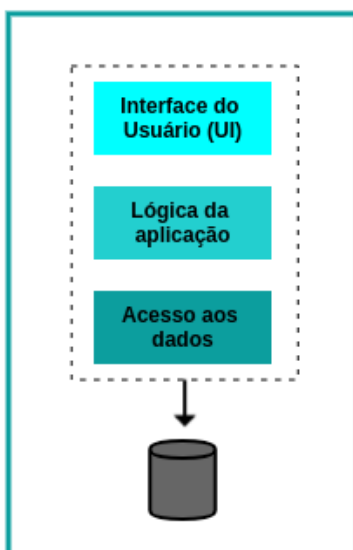


Nome: Felipe Braz Marques  
Matrícula: 22.1.4030

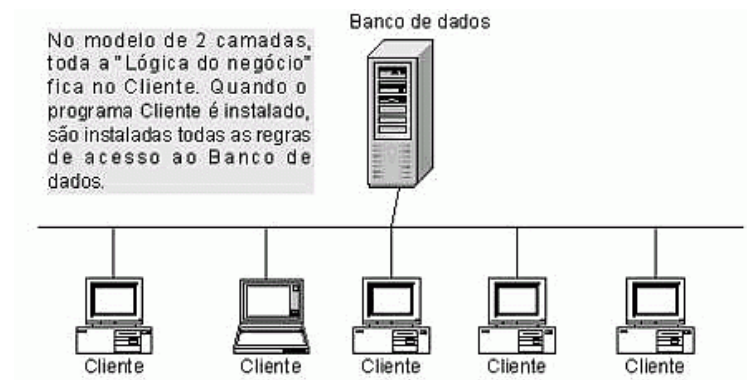
1. Estudo da melhor forma de se montar e executar um projeto, desde sua base, testes, APIs, banco de dados, e suas utilizações, divisões de equipes, e formas de se fazer num geral.
2. “É um esforço temporário empreendido para criar um produto, serviço ou resultado exclusivo”, é tudo aquilo que precisamos para gerar algo novo.
3. “A arquitetura de software de um sistema consiste na definição dos componentes de software, suas propriedades externas, e seus relacionamentos com outros softwares. O termo também se refere à documentação da arquitetura de software do sistema. A documentação da arquitetura do software facilita a comunicação entre os stakeholders, registra as decisões iniciais acerca do projeto de alto-nível, e permite o reuso do projeto dos componentes e padrões entre projetos. ”. Ou seja, é a forma como os componentes se comunicam, e estão distribuídos na base, como se interligam e seus protocolos de uso, entrada e saída.
4. Componente é um pedaço de software independente, mas que pode ser usado por outros softwares, é como se fosse uma espécie de biblioteca, algo que mesmo estando atrelado a um software específico, funciona separado, e pode ser anexado e usado em outros pontos. E o desenvolvimento baseado em componentes é uma técnica de programação que se baseia em tratar o sistema todo como vários mini sistemas, cada parte é gerida individualmente e todas juntas formam o sistema.
5. MONOLÍTICA: é um sistema único em todos os pontos, não dividido, que roda em um único processo, onde diferentes componentes estão ligados a um único programa dentro de uma única plataforma. Mais simples, de desenvolver, testar, alterar e adicionar, porém, é mais difícil de dar manutenção, é mais delicado, pois uma linha quebrada quebra tudo, e não há flexibilidade na linguagem utilizada.

**Arquitetura Monolítica**



CLIENTE-SERVIDOR: é uma arquitetura onde o processamento da informação é dividido em módulos ou processos distintos, um processo é responsável por manusear (servidor) e

outro por obter os dados (cliente). Os recursos são centralizados e têm maior facilidade de manutenção, porém é fácil de se ter uma sobrecarga por conta do único caminho.



**CAMADAS:** é um processo de decomposição de sistemas complexos em camadas para facilitar a compreensão do mesmo, como também, facilitar a manutenção deste sistema, ainda afirma que, esta técnica foi emprestada da arquitetura de computadores, que utilizam camadas de chamada ao sistema operacional, drivers e afins. Fácil manutenção, pois, isola as funções do SO, cria uma hierarquia de níveis protegendo as camadas mais internas, mas em cada camada tem um novo modo de acesso.



6. Alta coesão (um único foco), autônomos (independência de outros serviços que interajam com ele), resiliência (que saibam reagir a falhas inesperadas), observável (saber em tempo real o que está acontecendo), automatização (integração e entrega contínua) e centrado no domínio do negócio (conseguir dominar todos os pequenos conjuntos, conseguindo dominar o todo).

7. As bibliotecas são uma coleção de funções e códigos para uma linguagem de programação, já os frameworks são além de um conjunto de bibliotecas, elas possuem toda uma estrutura e um guia para a construção de algo, que herdará as características desta estrutura. A linguagem C contém diversas bibliotecas para diversas utilidades, já os frameworks um muito utilizado é o jQuery para manipulação de arquivos HTMLs. As bibliotecas são melhores quando queremos criar a base e o fluxo, já o framework é melhor quando queremos apenas “montar” as peças e usar.

8. “APIs são um conjunto de padrões que fazem parte de uma interface e que permitem a criação de plataformas de maneira mais simples e prática para desenvolvedores. A partir de APIs é possível criar softwares, aplicativos, programas e plataformas diversas”.

9. Acoplamento Fraco é aquele em que cada um dos seus componentes tem ou faz uso de pouco ou nenhum conhecimento das definições de outros componentes separados.

Alta Coesão é aquele componente que tem apenas uma única responsabilidade, que possui em suas funções apenas aquilo que ele deve fazer e mais nada.

10. É sempre bom separar pois durante a manutenção, que vai acontecer, um não interfere no outro, dois grupos podem trabalhar independentes do outro para uma maior eficiência e menor interferência.

11. Reuso de código é uma qualidade onde um código pode ser utilizado em diversos serviços, é como uma função de soma, que faz a soma de todo tipo de valor, int, double, float, etc. São vantajosos no ponto de não precisar ficar reinventando a roda, porém são mais complexos e abstratos que o comum, são importantes para ajudar na manutenção e clareza do código.

12. Teste, criam-se os testes, suas entradas e saída. Bibliotecas, criam-se os cabeçalhos, o que vai entrar e sair de cada função. Implementação, aí sim começa-se a implementar, pensar em qual estrutura usar e etc.

13. “A verificação tem como propósito averiguar se o software está de acordo com as especificações preestabelecidas, e a validação é o processo de confirmação de que o sistema está apropriado e consistente com os requisitos”.

14. Unitário: oferece valores válidos e inválidos, e verifica se os retornos são os esperados.

Funcional: verifica se o código funciona realmente, sem se preocupar com o método.

Integração: é testar o conjunto formado por várias funções, como testar uma chamada ao banco de dados.

Sistêmico: testa o sistema por completo, se interagem corretamente, são compatíveis, etc.

Aceitação: testa do ponto de vista do usuário, se é exatamente aquilo que se esperava e dar o feedback.

15. Caixa-Branca: é um teste que vem para validar a lógica do produto, o teste tem acesso aos códigos e as formas de implementação, visando avaliar a lógica utilizada, para garantir a qualidade da implementação.

Caixa-Preta: é um teste para avaliar o funcionamento do código, sem levar em conta a implementação e métodos, apenas de entra e sai o que se espera,

Caixa-Cinza: é apenas a junção dos dois métodos, testar a entrada e a saída, levando também em conta os métodos e implementações utilizadas.