

### Ponto 1 - Questões 1-5.

A mudança de paradigma no uso dos dados criados em sistemas de software trazem à tona discussões que podem impactar nossa relação com esses sistemas. Destaco algumas mudanças que nos trouxeram ao momento atual:

- a) Capacidade de processamento avançadas. Tecnologias como BIG DATA e Inteligência Artificial fomentam grandes avanços em processamento.
- b) Capacidade de produção dos dados acelerada. O uso extensivo de celulares, dispositivos inteligentes e agora com Internet das Coisas, a produção de dados é exponencial.
- c) Capacidade de armazenamento. Soluções em CLOUD, EDGE e Fog viabilizam a infraestrutura para tudo isso.

Antes disso não era possível, mas com mudanças como essas, os dados passam a ser um recurso produtivo. Considerando os recursos tradicionais os dados tem algumas diferenças como a limitação física de produção. O minério, por exemplo, é finito. Já os dados são produzidos diariamente em larga escala. Outra diferença chave é a questão da propriedade. O carro, por exemplo, o direito da propriedade pertence ao dono da terra onde está o carro. O dono pode explorar, vender e lucrar com isso. Mas e os dados?

A discussão de propriedade e privacidade levou a criação de uma nova tecnologia que se propõe justamente a responder essa pergunta, chamada de BLOCKCHAIN (cadeia de blocos). Relacionada principalmente a "carteira digital" ou "contabilidade distribuída" a blockchain é uma estrutura de dados para armazenamento de transações de modo descentralizado. Alguns pontos são importantes de considerar nessa tecnologia:

- É descentralizada. Em um modelo centralizado, os dados, transações e todo controle pertencem a uma autoridade central. Esse é o modelo usado pelo Stack do Google por exemplo, que processa e lucra com os dados de seus usuários. No modelo descentralizado, que é o da blockchain, não há essa entidade central. A autoridade é compartilhada por todos os usuários que fazem parte da rede.

- Segurança. Cada usuário tem sua identidade anonimizada, dessa forma a privacidade é garantida. Cada usuário também deve ter as permissões para interagir na blockchain, e isso é feito através de uma chave privada única e criptografada. Assim as transações são realizadas anonimamente mas só por participantes autorizados. Os usuários passam a serem nós que compõem a blockchain.

- Contrato Inteligente: todas as regras de segurança e permissões da blockchain tem como as transações autorizadas são restringidos por contratos que ~~ger~~ guiam o uso da blockchain. Nenhuma transação não cadastrada é permitida, o que também contribui para a segurança.

- Ponto de falha: relacionado ao modelo descentralizado a blockchain possibilita que toda a cadeia seja pública e acessível, com todo o histórico de transações para todos os nós da rede. Isso supera problemas de disponibilidade, pois enquanto um nó estiver de pé a blockchain segue em funcionamento (mesmo que outros falhem) e de transparência pois todos os nós na rede tem acesso as mesmas informações o tempo todo.

A tecnologia de blockchain tem se popularizado como uma alternativa ao controle monetário de bancos e também com aplicações descentralizadas (DApps). As aplicações descentralizadas permitem a utilização dos mais diversos tipos de serviço com o benefício de ter a blockchain como camada de comunicação. Para o desenvolvimento desse tipo de aplicação devemos considerar:

- Desenvolvimento em código aberto: para que o controle e a transparência sejam mantidos pela comunidade interessada no DApp.
- Consenso: devem ser estabelecidos mecanismos e algoritmos para controle de alterações e mudanças que só são feitas após consenso entre os usuários e que estejam aderentes com o contrato inteligente da blockchain utilizada.
- Incentivo: realizar incentivos para as ações dentro do próprio DApp. Em geral os incentivos são em formato de tokens que permite m realizar transações no DAPP. "Proof of Work" é um dos mecanismos que podem ser utilizados em que o serviço do usuário em melhor no DAPP ou mineração na blockchain são reconhecidos. "Proof of Stake" pode ser um mecanismo aplicado na questão do consenso, em que os usuários podem fazer valer seus votos conforme a força da rede.

Atualmente temos três tipos de DApps. Tipo 1 - são os Dapps que possuem sua própria blockchain. Um exemplo é o Ethereum, um aplicativo de pagamentos que roda na blockchain do mesmo nome. Tipo 2 - são Dapps que rodam em blockchains criadas pelo tipo 1. Um exemplo é o Golem, um aplicativo para alugar recursos computacionais, com pagamento feito em Tokens Golem. Por ultimo temos o tipo 3: DApps que dependem de DApps do tipo 2. Um exemplo é o DeML, uma aplicação de "machine learning" que usa tokens Golem na ~~seu~~ <sup>seu</sup> infraestrutura.

A tecnologia blockchain e as aplicações descentralizadas são avanços promissores para uma nova era de sistemas de software.

## Ponto 6 - Questão 6.5

A Engenharia de Software apoia o planejamento, a implementação, manutenção e evolução, através de processos e práticas, para que o produto resultante seja um software de qualidade.

A questão da qualidade de software, entretanto, é um desafio. Diferentes autores abordam essa questão por diferentes objetivos, mas a qualidade permanece um conceito complexo e contextual, pois depende de quem vê.

Qualidade de software tem se tornado cada vez mais relevante, pelo uso intensivo de software nos mais diversos domínios. Sistemas de Telemedicina, automação industrial e mais recentemente ambientes inteligentes - todo esse potencial e uso exponencial do software trazem preocupações quanto sua qualidade.

Para entender melhor a qualidade podemos considerar ~~os~~ modelos que procuram contextualizar o conceito em diferentes perspectivas, como por exemplo:

- Visão transcendental: onde a qualidade do software pode ser percebida mas não definida ou gerenciada.
- Visão do usuário: onde a qualidade é considerada por quem usa o software, principalmente na aderência aos requisitos, se o software realmente faz o que se propõe. Por exemplo, usabilidade.
- Visão de produto: relacionada a questões estruturais do software e percebida por pessoas envolvidas no processo de construção do produto, como os engenheiros de software. Por exemplo, manutenibilidade.
- Visão de Valor: qualidade associada ao custo do produto ou o quanto estão dispostos a pagar pelo software. Por exemplo, custo do projeto.

Essas visões podem ser adaptadas para três pontos da qualidade do software: Qualidade do Produto de Software, Qualidade do Processo de Software e Qualidade ~~no~~ no Contexto do Software.

A qualidade do produto de software considera a adequação aos requisitos e restrições de comportamento. Frequentemente discutidos como requisitos funcionais e não-funcionais, essas definições e atributos guiam

O que o produto deve (ou não) fazer). A qualidade do produto tem portanto o aspecto funcional (como a implementação de uma regra de negócio), ~~os~~ atributos externos que são percebidos pelos usuários (como uma autenticação de dois fatores que é percebida como segurança), ou atributos internos que são percebidos pela equipe de desenvolvimento (como por exemplo reusabilidade). O modelo QPS (Qualidade de produto de software) pode ser usado para analisar essa perspectiva.

A qualidade do processo de software procura observar a aderência com boas práticas de desenvolvimento, definição do ciclo de vida e as atividades em relação a processos centrais, de apoio ou gestão envolvidos na construção do produto. Existe uma relação entre a qualidade do produto e a qualidade do processo que faz o produto, portanto é essencial que a qualidade do processo seja observada. Modelos como o MPSBr e o CMMI podem auxiliar na gestão e maturidade dos processos de software.

A qualidade no contexto considera que o produto de software é construído em uma organização, que reforça esse processo. Nesse contexto, a perspectiva do negócio deve ser considerada. Em contraponto com a qualidade técnica ou em uso, a alta direção está preocupada com prazos, riscos, custos e retorno de investimento. Assim, esse aspecto de qualidade não deve ser desconsiderado.

~~Porém~~ A qualidade de software não é algo que vem naturalmente, precisa ser buscada e construída. Para isso precisamos colocar em práticas mecanismos e estratégias para gestão da qualidade, destacando três direções:

- 1) Aplicação dos métodos da Engenharia de Software: a definição de processos para desenvolver, operar e evoluir o ~~produto~~ produto de software é essencial. Os métodos podem ser apoiados por tecnologias e ferramentas que ajudam no controle das atividades e compartilhamento de conhecimento ao longo de todo o ciclo de vida.
- 2) Gestão do projeto: a cultura da organização e a equipe de desenvolvimento, além de outros fatores, tem impacto direto no produto e sua qualidade. Uma gestão de projetos eficaz, com cronograma adequado, plano de comunicação e gestão de riscos pode contribuir para a qualidade final.
- 3) Garantia da Qualidade: A garantia da qualidade é em si um processo, por isso é ideal ter um plano, atividades e responsabilidades bem definidas. É importante considerar práticas para: EVITAR erros, faltas, falhas, defeitos ou qualquer anomalia de software. O uso de padrões e boas práticas de código podem ser uma opção. REMOVER defeitos: técnicas de verificação, validação e testes são essenciais para ter como remover defeitos desde a concepção até realização do software. TOLERAR defeitos: nenhum software está isento de problemas por isso mecanismos de redundância e uma arquitetura de alta disponibilidade podem ajudar na tolerância e manter o sistema em funcionamento.

Temos tido grandes avanços tecnológicos na área de Engenharia de Software, mas ainda temos desafios a serem vencidos na construção de sistemas de software de qualidade. A Gestão da Qualidade é um avanço nessa direção.

### Ponto 8 - Questão 8.5

O paradigma de Internet das Coisas tem atraído grande interesse de várias comunidades, tanto no aspecto de pesquisa para o avanço da tecnologia quanto pelo potencial e abrangência das soluções propostas.

Internet das Coisas (do inglês - INTERNET OF THINGS - IOT) propõe a criação de novos sistemas de software a partir de dispositivos com comportamentos de identificação, sensoriamento e processamento que, a partir de uma comunicação, podem interagir para alcançar os mais diversos objetivos.

As preocupações para o desenvolvimento de software para IOT ~~podem~~ podem ser tratadas pela Engenharia de Software, que enxerga o software como um produto de engenharia e propõe processos, princípios e práticas para planejar, desenvolver, operar e evoluir esse produto. Nesse paradigma o desafio se mantém, desenvolver produtos de software de qualidade dentro das particularidades de IOT.

Essas particularidades se dão pois as soluções IOT são multifacetadas, com atuação de diferentes habilidades em seus elementos (ou BUILDING BLOCKS). Os elementos variam por domínio e necessidade da solução, mas em geral uma solução IOT considera:

- **Ambiente:** as condições ambientais e o contexto de uso do IOT pode afetar seu funcionamento. Por exemplo, o monitoramento de um laboratório emite um alarme quando a temperatura está elevada.
- **Dispositivos:** são os sensores, atuadores e qualquer "coisa" com capacidades em software para compor a solução. No exemplo, temos um sensor de temperatura.
- **Comportamento:** são as definições para atingir o objetivo da solução IOT. No caso, o comportamento é de monitoramento. Mas poderia ser de atuação se ao tocar o alarme, o ar condicionado ligasse para ajustar a temperatura.
- **Inteligência:** ligado ao nível de autonomia que a solução tem, em relação a atuação humana. Também referido como SMARTNESS.

- **Conectividade**: todos os elementos da solução devem ser capazes de e **Interatividade** interagir de alguma forma. Isso se relaciona a interação entre as coisa e os diferentes atores do sistema.
- **Dados**: todos os dados devem ser processados de modo a apoiar na tomada de decisão e alcançar os objetivos da solução.

Todos esses elementos devem ser definidos, organizados e gerenciados em uma solução IoT. Um dos passos para essa materialização é organizar a arquitetura da solução. A definição de uma arquitetura permite uma ~~visualização~~ visualização dos elementos bem como relacionamentos e interações. Isso facilita o compartilhamento do conhecimento e a comunicação entre os colaboradores. As decisões arquiteturais são muito importantes pois podem impactar a qualidade do produto, como desempenho por exemplo.

Para soluções IoT uma das propostas é tratar a arquitetura em três camadas, que podem ser discutidas quanto preocupações, tecnologias e qualidade.

- ① **Camada de Sensoriamento (ou de Dispositivo)**: essa camada é responsável por organizar os dispositivos e tratar a conectividade inicial (interna). As principais preocupações para considerar são a capacidade da arquitetura de prover um identificador único para cada dispositivo e conseguir comunicar (considerando entrada e saída) para a próxima camada. Protocolos como o MQTT podem contribuir para essa comunicação, que deve atentar para uma gestão de recursos eficiente, pois os dispositivos podem ter diferentes capacidades (energia, processamento...).
- ② **Camada de agregação (ou processamento ou análise)**: organizar os dados dos dispositivos e provê-los como informação é feito nessa camada. As preocupações são com agregar dados de diferentes dispositivos (e em grande quantidade), bem como processar e armazenar tudo isso. Tecnologias de middleware podem ajudar na agregação, análise pode ser feita com Hadoop ou soluções Elastic, e o armazenamento em nuvem são opções para essa camada. Deve ter atenção com a acurácia, pois os resultados das análises, tendências e insights dependem da qualidade dos dados capturados.
- ③ **Camada de Aplicação (ou serviço)**: As preocupações dessa camada estão relacionadas ao provisionamento e serviços, pois é responsável pela forma que os dados serão utilizados. Tecnologias como dashboards permitem a interação com atores humanos e serviços disponibilizados por APIs podem ser usados por outros sistemas. Nessa camada precisamos atentar para a escalabilidade da rede e ter recursos para garantir a disponibilidade dos serviços.

A segurança é um ponto essencial, que deve ser considerado em todas as camadas com mecanismos adequados.

Considerando os elementos, camadas da arquitetura, tecnologias e qualidade temos muitas oportunidades de pesquisa e desenvolvimento para contribuir com o estabelecimento de soluções de Software baseadas em Internet das Coisas.