DESENVOLVIMENTO WEB

Arquiteturas e Metodologias para Desenvolvimento Web

ROTEIRO

- Introdução
- Estratégias de geração de conteúdo
  - Linguagens de programação e de marcação
- Aplicações Web
- Arquiteturas Web
  - Monolíticas x Micro-Serviços

INTRODUÇÃO

- No passado, o universo Web era baseado especialmente em páginas estáticas (sites simples)
- Atualmente o universo Web é muito mais rico e inclui inúmeras aplicações e a integração com diversas APIs
- Seus aplicativos do celular são escritos em linguagem Web, pois códigos dos sistemas operacionais são muito específicos
    - Por isso que não é possível criar um app que roda tanto em Android e em iOS.

INTRODUÇÃO

- Por essa razão há frameworks de desenvolvimento e metodologias baseadas em linguagens da Web como JavaScript por exemplo
- Isso permite uma grande versatilidade de criar aplicações e gerar produtos digitais
  - Pode ser uma página textual ou complexas ferramentas de vendas na Internet

INTRODUÇÃO

- Há uma variedade enorme em relação ao escopo de projetos da Web pois incluem simples sites até aplicações multifuncionais
- Isso indica que as equipes de desenvolvimento também sejam multifuncionais. Por exemplo, no desenvolvimento de uma aplicação complexa podemos ter diversas áreas:
  - Segurança em Rede
  - Gestão dos bancos de dados
  - Desenvolvedor front-end e back-end
  - Engenharia Web

ESTRATÉGIAS - GERAR CONTEÚDO

- Uma aplicação Web típica tem em sua estrutura códigos que serão gerados a partir do processamento de dados em servidores de aplicações (Web)
- Há códigos relacionados a linguagem de marcação (HTML) e também relacionados a programação como PHP, JSP, Ruby, etc
  - É por isso que temos a figura do desenvolvedor front-end e do back-end

ESTRATÉGIAS - GERAR CONTEÚDO

- Linguagens como PHP por exemplo são utilizadas para criar e estruturar as regras de negócio da aplicação e para manipular os dados armazenados em SGBDs
- Por outro lado a linguagem de marcação HMTL é usada para interagir com o usuário por meio de criação de interfaces intuitivas e acessíveis

ESTRATÉGIAS - GERAR CONTEÚDO

- Embora atualmente haja uma divisão do que é o front-end e o back-end não é tão simples conciliar os dois mundos pois:
  - Há ferramentas distintas (IDE x Editores)
  - Habilidades diferentes dos profissionais (Fron-end x Back-end)
  - Uma IDE pode ser boa para escrever código JAVA, PHP mas não tão adequada para escrever código HTML e CSS
  - Um editor HTML e CSS pode ser ruim para editar e construir códigos em linguagens que executam no back-end

ESTRATÉGIAS - GERAR CONTEÚDO

- Boas estratégias de geração de conteúdo envolvem:
  - Organizar a forma de trabalho com as tecnologias e com os profissionais envolvidos. Isso é fundamental pra sistemas grandes e complexos e com equipes multidisciplinares
  - Criar o hábito de documentos o seu código
  - Aprenderr com arquitetos e desenvolvedores experientes

APLICAÇÕES WEB

- A variedade de aplicações Web é enorme, tanto em domínio quanto em tipo
  - Redes Sociais
  - Blogs Pessoais
  - Comércio Eletrônico
  - Internet-Banking
  - Plataforma de Ensino Online

APLICAÇÕES WEB

- Essas aplicações Web também apresentam grande variedade de requisitos não funcionais, como por exemplo:
  - Segurança
  - Desempenho
  - Escalabilidade
  - Disponibilidade
- Deste modo, ao pensarmos em uma aplicação Web atual precisamos considerar não somente a interface com o usuário, mas se esta aplicação atende a alguns desses requisitos não funcionais, os quais dependem por exemplo em que local a aplicação está hospedada, etc.

ARQUITETURAS WEB

- Arquitetar uma aplicação de software é fundamental para separar as responsabilidades entre os componentes
  - Ajuda a entender os requisitos não-funcionais esperados pela aplicação
  - Embora a complexidade aumente, pois quanto mais componentes, mais pontos para tratar

ARQUITETURAS WEB

- No curto, médio e longo prazo a arquitetura escolhida pode impactar o funcionamento da aplicação
  - No curto prazo em geral focamos apenas no desenvolvimento
  - No médio prazo o foco é na produção da aplicação
  - No longo prazo a preocupação recai sobre a manutenção da aplicação

ARQUITETURAS WEB

- As arquiteturas separam as responsabilidades em camadas em que cada uma delas pode estar em servidores diferentes
- Exemplos
  - Arquitetura Cliente-Servidor
    - Em geral tem relação com o processamento e armazenamento
  - Arquitetura em 2 camadas
    - Separa o processamento/armazenamento e a apresentação dos resultados
  - Arquitetura em 3 camadas
    - Separa aplicação da apresentação e do processamento/armazenamento

ARQUITETURAS WEB

- Podemos também organizar as arquiteturas em monolíticas e de micro-serviços
  - Monolítica
    - É simples e foi muito utilizada no passado
      - Suas características são:
        - Suporta diversos tipos de clientes (mobile e desktop)
        - Pode integrar com outras aplicações com REST ou SOAP
        - Trata requisições HTTP, executa regras de negócios, acessa banco de dados, etc.

ARQUITETURAS WEB

- Continua...
  - Monolítica
    - Exemplo: Um sistema para reservas em um hotel
      - Recebe reservas dos clientes (app cliente)
      - Verifica disponibilidade de quartos (back-end)
      - Valida pagamento (back-end)
      - Faz reserva (back-end)
      - Avisa o hotel (back-end)
    - *Veja o número de camadas da aplicação!!!*

ARQUITETURAS WEB

- Continua...
  - Monolítica
    - Vantagens
      - Desenvolvimento inicial rápido
      - Mais simples de aprender
      - Usa uma infraestrutura simples
    - Desvantagens
      - Componentes com acomplamento forte
      - Mudanças em um componente, precisa atualizar a aplicação inteira
      - Tempo de inicialização grande e muito uso de CPU e memória

ARQUITETURAS WEB

- Micro-serviços
  - Arquitetura mais atual e muito popular entre arquitetos e desenvolvedores Web
  - Oferece muito das funcionalidades da arquitetura monolítica
  - Os componentes têm baixo acoplamento e exigem monitoramento
  - São mais leves e com funcionalidades específicas
  - Cada componente pode ser desenvolvido com uma stack de tecnologia própria
  - Podem ser desenvolvidos, testados, implantados e escalados de forma independente

ARQUITETURAS WEB

- Continua…
  - Micro-serviços
    - Exemplo: Um sistema de vendas online em que usuários podem navegar entre categorias, escolher e adicionar produtos ao carrinho, fazer e acompanhar pedidos
      - O sistema precisa fazer gestão de estoque, de usuários, métodos de pagamento diferentes, gestão de compras e de entregas
      - É preciso que a interface com o usuário tenha um sistema de autenticação, e seja simples e fácil de usar

ARQUITETURAS WEB

- Continua...
  - Micro-serviços
    - Vantagens
      - Baixo acomplamento entre os componentes, mais fácil de testar e inicializa mais rápido
      - A falha em um componente isolado não para toda a aplicação
      - Os times de desenvolvimento são independentes (bom para estratégias de desenvolvimento Ágeis)
    - Desvantagens
      - No geral, o desenvolvimento fica mais complexo
      - Os testes e a integração são custosos e complexos (têm muitas partes separadas)
      - Precisa de uma infraestrutura mais complexa (dockers)

Metodologias de Desenvolvimento

- Há diversos métodos e abordagens que podem ser utilizados para desenvolver um software e elas se encaixam atualmente em duas categorias:
  - Metodologias Ágeis
    - Scrum
    - Kanban
    - Lean
    - Smart
  - Cascata

Metodologias de Desenvolvimento

- Metodologia Ágil
  - Foca no projeto e visa fazer melhorias contínuas na aplicação, com base em feedbacks de usuários, clientes e do time de desenvolvimento. Não tem estruturas rígidas e atua com processos de desenvolvimento curtos. Problemas podem ser resolvidos em estágio inicial, o que pode melhorar a qualidade

  - Scrum
    - Tem como objetivo agregar mais produtividade aos processos
    - Recursos: listas sobre trabalhos pendentes, reuniões diárias, sessões de planejamento para tratar os problemas. Equipes podem identificar e corrigir os problemas rapidamente

Metodologias de Desenvolvimento

- Continua…
  - Kanban
    - Interessante para equipes que recebem muitos pedidos. Atualizações solicitadas são liberadas quando ficam prontas. Não tem período fixo para entrega de tarefas. É flexível e muito boa para desenvolver sistemas que têm muitos requisitos de mudanças
  - Lean
    - Estratégia que se concentra na redução das perdas durante o processo de desenvolvimento. O valor a ser gerado para os clientes é o ponto fundamental de todas as atividades. O projeto é analisado de forma minuciosa desde o início. É bom para projetos com orçamento limitado e com tempo pequeno para ser desenvolvido.

Metodologias de Desenvolvimento

- Continua…
  - Smart
    - É uma ferramenta que tem o objetivo de definir metas construídas de forma a se considerar 5 atributos: S (Específico), M (Mensurável), A (Atingível), R (Relevante) e T (Temporal)...

- Cascata
  - É o oposto da metodologia Ágil, sendo portanto mais travada, pois adota controle e processos com mais rigor. Só inicia uma etapa quando a anterior for concluída

REFERÊNCIAS

1. https://mundodevops.com/blog/desenvolvimento-web/
2. https://www.monitoratec.com.br/blog/metodologias-de-desenvolvimento-de-software/#
3. https://www.atlassian.com/br/agile/scrum
4. https://www.totvs.com/blog/negocios/metodologia-agil/
5. https://rockcontent.com/br/blog/kanban/
6. https://www.ludospro.com.br/blog/metodologia-lean
7. Ajax, Rich Internet Applications e Desenvolvimento Web para Programadores
8. Desenvolvimento de Software II: Introdução ao Desenvolvimento Web com HTML, CSS, JavaScript e PHP
9. https://rockcontent.com/br/blog/metodologias-ageis/

DESENVOLVIMENTO WEB

Arquiteturas e Metodologias para Desenvolvimento Web