

**UNIVERSIDADE ANHANGUERA – UNIDERP**

POLO DE (Cotia-SP)

**Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas- TADS**

(MARCIANO JOSE DA COSTA – RA 24866488)

PROJETO INTEGRADO I

Disciplinas Norteadoras:

**Engenharia de Software**

**Redes e Sistemas Distribuídos**

**Programação Web**

**Desenvolvimento Mobile**

Tutor (a) EAD: CRISTIANO GIROLDO

COTIA / SP 2021

SUMÁRIO

1 – INTRODUÇÃO……………………………………....…...…………………...pg. 3

2 - TAREFA 1 – **O SCRUM** .........................…................................………....pg. 4

3 - TAREFA 2 - **O que é um Sistema Distribuido?** ..……............................pg. 9

4 - TAREFA 3 – **O que é Middleware?** ………………....………….................pg. 12

5 – TAREFA 4 - **O que é Media Query?**........................................................pg. 17

6 – TAREFA 5 - **Plataformas e Linguagens para Desenv. Android**...........pg. 18

7 – CONCLUSÃO....................................................................................…....pg. 21

8 – REFERÊNCIAS BIBLIOGRÁFICAS..................................................…....pg. 22

INTRODUÇÃO

Este trabalho de Projeto Integrado atinge o objetivo de resumir alguns assuntos estudados durante o 3.º semestre do Curso de Desenvolvimento e Análise de Sistemas da Universidade Anhanguera.

Nossa contextualização irá abordar o tema Fake News tendo em mente o quanto uma notícia falsa pode prejudicar a nossa sociedade.

*“No portal da Fiocruz, no dia 15 de abril de 2020, foi publicada dados de pesquisa sobre 'fake news' relacionadas à Covid-19. Os dados revelam que 10,5% das notícias falsas foram publicadas no Instagram, 15,8% no Facebook e 73,7% circuladas pelo WhatsApp. Os resultados também apontam que 26,6% das fake news publicadas no Facebook atribuem a Fiocruz como orientadora no que diz respeito à proteção contra o novo coronavírus. O estudo ainda revela que 71,4% das mensagens falsas circuladas pelo WhatsApp citam a Fundação como fonte de textos sobre a Covid-19 e com medidas de proteção e combate à doença. A Organização Mundial da Saúde (OMS) e o Fundo das Nações Unidas para a Infância (Unicef), juntas, somam 2% das instituições citadas como fonte de informações sobre cuidados e medidas contra o novo coronavírus em mensagens de WhatsApp.”*

Os assuntos abordados serão: A metodologia ágil de desenvolvimento SCRUM, onde será explicado as principais características dessa metodologia desde sua história até as especificidades de sua utilização, demonstrando como essa ferramenta é fundamental para acelerar e organizar os processos de desenvolvimento.

Em seguida abordaremos o tema Sistemas Distribuídos onde será relacionado as principais vantagens em se implementar um sistema distribuído ao invés de um sistema centralizado. Iremos fornecer uma definição sobre o que é “Middleware” e apresentar exemplos das plataformas mais utilizadas.

Logo após, já pensando sobre os frameworks de desenvolvimento que buscam a responsividade das aplicações iremos abordar o importante recurso chamado de “Media Query”.

Por fim, iremos sugerir uma plataforma de desenvolvimento multiplataforma que acelera o desenvolvimento para aplicações Android e iOS.

TAREFA 1:

**ATIVIDADES:**

Sabendo que um trabalho que realiza a identificação de Fake News deve ser rápido e dinâmico, uma empresa que trabalha com isso, deve ter o mesmo comportamento em seus processos. Por este motivo, indique e explique uma metodologia ágil que poderia ajudar a dinamizar os processos de uma instituição. Não esqueça de apresentar as principais características desta metodologia, a equipe de trabalho e os métodos que essas metodologias adotam.

**SCRUM**

O Scrum é um framework de gerenciamento de projetos, da organização ao desenvolvimento ágil de produtos complexos e adaptativos com o mais alto valor possível, através de várias técnicas, utilizado desde o início de 1990 e que atualmente é utilizado em mais de 60% dos projetos ágeis em todo o mundo.

Este framework não é um processo linear, em vez disso, é um conjunto de conceitos e técnicas dentro do qual você pode empregar vários outros processos ou técnicas.

Analisando a eficácia relativa das práticas de gerenciamento e desenvolvimento, de modo que você possa melhorá-las.

Este também não é um processo prescribente, ou seja, não descreve o que fazer em cada situação, pois em trabalhos complexos é impossível predizer tudo o que irá ocorrer.

É um conjunto de valores, princípios e práticas que fornecem a base para que a sua organização adicione suas práticas particulares de engenharia e gestão e que sejam relevantes para a realidade da sua empresa.

O resultado será uma versão de Scrum exclusiva.

Scrum possui seu foco no gerenciamento e projeto da organização onde é difícil planejar à frente.

Mecanismos do Controle de Processo Empírico, onde ciclos de feedback constituem o núcleo da técnica de gerenciamento que são usadas em oposição ao tradicional gerenciamento de comando e controle.

É uma forma de planear e gerenciar projetos trazendo a autoridade da tomada de decisão a níveis de propriedade de operação e certeza.

Apesar de a palavra não ser um acrônimo, algumas empresas que implementam o processo a soletram com letras maiúsculas como SCRUM.

Isto pode ser devido aos primeiros artigos de Ken Schwaber, que capitalizava SCRUM no título.

Apesar de ser criado para gerenciamento de projetos de software, também pode ser utilizado em equipes de manutenção de software ou como uma abordagem geral de gerenciamento de projetos/programas.

Atualmente, as técnicas de DevOps são utilizados por programadores no desenvolvimento de software em conjunto com técnicas de gestão e desenvolvimento ágil de software, como o Scrum.

Inicialmente, o Scrum foi concebido como um estilo de gerenciamento de produtos em empresas de fabricação de automóveis e produtos de consumo, por Takeuchi e Nonaka no artigo "The New Product Development Game" (Harvard Business Review, janeiro-fevereiro 1986).

Eles notaram que projetos usando equipes pequenas e multidisciplinares (cross-functional) produziram os melhores resultados, e associaram estas equipes altamente eficazes à formação Scrum do Rugby (utilizada para reinício do jogo em certos casos). Jeff Sutherland, John Scumniotales e Jeff McKenna conceberam, documentaram e implementaram o Scrum, conforme descrito abaixo, na empresa Easel Corporation em 1993, incorporando os estilos de gerenciamento observados por Takeuchi e Nonaka.

Ken Schwaber e Jeff Sutherland fizeram a primeira co-apresentação do Scrum na conferência OOPSLA de 1995.

Esta apresentação essencialmente documentou o aprendizado que Ken e Jeff tiveram ao longo dos anos anteriores na aplicação do Scrum e ajudaram a implantá-lo no desenvolvimento de softwares em todo o mundo.

Scrum junta conceitos de Lean, desenvolvimento iterativo e do estudo de Hirotaka Takeuchi e Ikujiro Nonaka.

A função primária do Scrum é ser utilizado para o gerenciamento de projetos de desenvolvimento de software.

Ele tem sido usado com sucesso para isso, assim como Extreme Programming e outras metodologias de desenvolvimento.

Porém, teoricamente pode ser aplicado em qualquer contexto no qual um grupo de pessoas necessitem trabalhar juntas para atingir um objetivo comum, como iniciar uma escola pequena, projetos de pesquisa científica, ou até mesmo o planejamento de um casamento.

Mesmo que idealizado para ser utilizado em gestão de projetos de desenvolvimento de software ele também pode ser usado para a gerência de equipes de manutenção, ou como uma abordagem para gestão de programas: Scrum de Scrums.

No SCRUM existem papéis bem definidos e diversas etapas que devem ser cumpridas em prazos estipulados visando entregar o produto de forma rápida e que ao mesmo tempo em que atenda as expectativas do cliente.

O proprietário do produto ou Product Owner representa os stakeholders e o negócio, a equipe ou Team é formada por cerca de 7 pessoas, as equipes com poucos membros e multidisciplinares são uma das principais características do SCRUM, com o tempo foi observado em diversos projetos que equipes pequenas conseguem muitas vezes serem mais produtivas dependendo da forma como trabalham.

Para gerenciar o projeto surge a figura do ScrumMaster que atua como um gerente de projeto, coordenando a equipe para que as metas sejam alcançadas. Agora vejamos as etapas do processo:

**Product Backlog**

É basicamente uma lista com requisitos e suas prioridades definidas pelo Product Owner.

No product backlog são definidas as funcionalidades a serem entregues ao cliente, este documento pode ser alterado a qualquer momento, está é outra característica importante do SCRUM, neste processo a equipe tem que ter ciência da natureza mutável do software, ou seja, as necessidades dos clientes podem e muito provavelmente vão mudar a qualquer momento e um processo de desenvolvimento ágil tem de ser flexível neste ponto. Quando se tem o Product Backlog pronto pode-se partir para a próxima etapa o Sprint Backlog.

**Sprint Backlog**

Consiste em uma lista de itens selecionados do Product Backog que serão realizados no próximo Sprint, esta lista é definida durante uma reunião chamada Sprint Planning Meeting na qual a equipe decide o que vai ser realizado em determinado Sprint, aliás as reuniões são um dos pontos centrais do SCRUM.

**Sprint**

A julgar pela duração de um sprint (entre 7e 30 dias) o nome sprint(corrida) é bem apropriado, de grosso modo um sprint é o desenvolvimento de um incremento de um software a ser entregue em determinado prazo, mas esse processo envolve algumas nuances a mais que veremos a seguir.

**Daily Scrum**

É uma reunião diária realizada a cada dia de um sprint, está reunião tem um conjunto de regras bem definido:

• A reunião começa precisamente no horário marcado.

• Todos são bem-vindos, mas apenas "poucos" podem falar.

• O encontro tem duração determinada (Time-Box) e dura 15 minutos.

• A reunião deve acontecer no mesmo local e mesma hora todos os dias

• Durante a reunião, cada membro da equipe responde a três perguntas:

◦ O que você tem feito desde ontem?

◦ O que você está planejando fazer hoje?

◦ Você tem algum problema impedindo você de realizar seu objetivo?

Uma das vantagens das reuniões curtas é que evitam o cansaço causado por reuniões delongadas e que muitas vezes acabam sendo pouco produtivas, também é importante destacar que desta forma também se garante que todos os membros da equipe estarão a par de cada progresso feito durante o projeto.

**Reunião de Revisão da Sprint (Sprint Review)**

Esta reunião tem dois objetivos principais: rever o trabalho concluído e o não concluído e apresentar uma demo ao cliente. Dessa forma a cada sprint o cliente poderá ver uma parte do produto funcionando e participar do processo fazendo sugestões de melhorias.

Retrospectiva da Sprint (Sprint Retrospective)

Está reunião tem o objetivo de rever os erros e acertos no sprint realizado.

Está é uma das partes mais importantes do processo, já que nela é possível aprender com os erros e tentar aprimorar o produto, com isso é possível obter o que se procura não só quando se está desenvolvendo software, mas em qualquer outro produto: a melhoria contínua.

Conclusão

Resumindo o SCRUM é um processo que se bem utilizado pode agregar muito valor ao produto que vai ser entregue.

No entanto é preciso enfatizar que nenhum dos métodos ágeis existentes faz milagre, você até pode conseguir entregar mais rápido, mas para conseguir usar todo o potencial dos métodos ágeis será preciso ter uma boa equipe que se comprometa em colaborar para melhoria constante do processo, é como dizem:

só a prática leva a perfeição.

TAREFA 2:

**ATIVIDADE:**

No âmbito de uma futura expansão e melhoria de todo o sistema computacional, para a detecção de fake news, explique a respeito das principais vantagens ao se implementar um sistema distribuído ao invés de um sistema centralizado.

Os sistemas distribuídos estão em todo o lugar, ou melhor, acessíveis a partir de qualquer lugar.

Segundo Tanenbaum, um sistema distribuído é um conjunto de computadores independentes entre si (e até diferentes), ligados através de uma rede de dados, que se apresentam aos utilizadores como um sistema único e coerente.

No mundo da tecnologia atual, são vários os exemplos de sistemas distribuídos. Alguns que se destacam:

• Sistemas de pesquisas (motores de busca)

• Sistemas financeiros

• Jogos Online

• Redes Sociais e plataformas idênticas

• etc.

Todos os sistemas apresentados anteriormente têm máquinas dispersas por todo o mundo, com acessos concorrentes e muitos desafios nos processos de comunicação (ex. hora do sistema – devido à distribuição geográfica, possíveis falhas de componentes, etc.).

A comunicação é simplesmente feita com base em mensagens.

Mas há outros desafios na implementação de sistemas distribuídos que se destacam como é o caso da:

**• Heterogeneidade**

◦ Sistemas Operativos diferentes

◦ Tipos de Redes de Comunicação variadas

◦ Hardware diferente

**• Sistemas Abertos**

◦ Para que os sistemas “falem” entre si é importante que estes sejam abertos, pois só assim será também mais fácil a extensão/reimplementação.

**• Segurança**

◦ Um fator fundamental para a transmissão de informação entre sistemas. A encriptação pode ser usada para a comunicação de mensagens na rede.

**•   Escalabilidade**

◦ Um sistema distribuído deve ser escalável para que possam ser adicionados novos sistemas de uma forma simples, com custos reduzidos e sem impacto no serviço.

**• Tratamento de Falhas**

◦ Qualquer processo, computador ou rede pode falhar de forma independente: A acontecer, tal não deve afetar o serviço.

**• Concorrência**

◦ A presença de múltiplos utilizadores num sistema distribuído pode significar o acesso/modificação, simultâneo de um recurso.

**• Qualidade de Serviço (QoS)**

◦ O QoS está relacionado com a fiabilidade do serviço.

**As maiores vantagens são:**

• **Escalabilidade**: Você pode “facilmente” crescer a capacidade de atendimento adicionando novos nós de execução, e isso faz atender demandas grandes no prazo estabelecido.

Ao contrário do que as pessoas acham, não fica mais rápido.

Você ter mais caminhões transportando uma carga não entrega a carga mais rapidamente, apenas entrega mais carga no mesmo prazo.

• **Resiliência**: Se um nó tiver problemas os outros continuam operando e o sistema como um todo não para.

Todas as outras vantagens são derivadas destas ou sem muita importância.

Principais desvantagens:

• **Complexidade:** Este é um dos problemas mais difíceis de resolver em toda computação, coordenar tudo é complicado, custa caro em todos os sentidos e a chance de algo dar errado no processo é grande.

Aqui incluo problemas de segurança que aparecem, ainda que não são insolúveis, apenas são complexos.

• **Desperdício**: É comum usar muito mais recursos para operar nesta modalidade já que há um custo de coordenação.

**Também há desvantagens derivadas destas ou eventuais.**

**Depende:**

• **Custo**: Há casos que o custo pode ser menor usando um ou outro. O que muita gente não percebe é que o que parece barato pode se tornar caro.

Há situações muito específicas que não fazer distribuído fica inviável, e essa é a boa notícia.

O problema vem quando parece que dá para fazer sem distribuição, e aí o hardware custará muito caro, ou mesmo o software tem que ser feito em termos muito detalhados com muita otimização e isto pode ser absurdamente caro.

O contrário parece ser mais frequente.

As pessoas optam pelo distribuído por conta das vantagens, e nem se importam com o custo, ele custará mais caro, mas a pessoa acha que precisa de qualquer forma.

Acontece que raríssimos problemas realmente precisam da distribuição, pelo menos a distribuição real, aquela que precisa de coordenação e não apenas parcelamento simples de carga de trabalhos absolutamente independentes (isto não é distribuição).

TAREFA 3

**ATIVIDADE:**

Defina Middleware e explique qual o seu papel em um Sistema Distribuído e apresente pelo menos 2 exemplos de plataformas de middlewares mais utilizadas, explicando suas principais características.

**Middleware** é um software que fornece serviços e recursos comuns a aplicações. Gerenciamento de dados, serviços de aplicações, sistema de mensageria, autenticação e gerenciamento de APIs são recursos comumente operados por um software de middleware.

Com o middleware, os desenvolvedores podem criar aplicações com mais facilidade e eficiência, pois esse tipo de software tem o papel de conectar aplicações, dados e usuários.

Para as organizações com ambientes multicloud e em containers, o uso de middleware torna o desenvolvimento e a execução de aplicações em escala algo econômico.

Em termos gerais, o middleware engloba de tudo, desde servidores web a sistemas de autenticação e ferramentas de mensageria.

Para o desenvolvimento de novas aplicações:

O middleware viabiliza o uso de ambientes de execução modernos e populares em uma ampla gama de casos de uso.

Desenvolvedores e arquitetos podem trabalhar com agilidade em diferentes plataformas, seguindo conjuntos de ambientes de execução, frameworks e linguagens de programação fundamentais. Além disso, o middleware oferece funcionalidades comumente utilizadas, como servidores web, servidor de SSO, sistema de mensageria e armazenamento em cache em memória.

Para otimização das aplicações existentes, com o middleware, os desenvolvedores podem transformar aplicações monolíticas legadas em aplicações nativas em nuvens com mais facilidade, conferindo mais desempenho e portabilidade a ferramentas valiosas.

Integração abrangente:

As ferramentas de integração de middleware conectam sistemas internos e externos críticos.

Os recursos de integração, como de transformação, conectividade e mensageria corporativa, combinados à autenticação por SSO tornam mais fácil para os desenvolvedores estender os recursos do sistema a aplicações diferentes.

Muitas interfaces de programação de aplicações (APIs)

Muitos serviços de middleware podem ser acessados por meio de APIs.

**QUAIS SÃO OS PRINCIPAIS BENEFÍCIOS DO MIDDLEWARE?**

É muito comum em grandes empresas a utilização de aplicativos de diferentes arquiteturas, seja pela aquisição de novas tecnologias ou pela fusão com outra empresa a qual utiliza outras aplicações, o que torna necessária uma solução para trabalhar com múltiplas arquiteturas.

O Middleware foi desenvolvido para suportar as arquiteturas por meio de métodos orientados a processos, permitindo que dados possam ser movidos de uma aplicação para outra.

**ACESSO A TECNOLOGIAS DISTINTAS**

Por maior que seja a infraestrutura de TI em uma empresa, será possível prover acesso a tecnologias distintas e localizadas em diferentes ambientes físicos (inclusive tecnologias em nuvem e móveis) graças à integração que é feita pelo Middleware.

**PROCESSAMENTO DE INFORMAÇÕES**

As informações capturadas pelos leitores, cuja camada antecede ao Middleware, podem retornar valores nulos e irrelevantes, gerando um enorme desperdício de processamento devido ao volume.

Com o Middleware intermediando essas informações de fontes distintas, elas passam a ser filtradas e refinadas, reduzindo consideravelmente o volume de dados processados e apresentando apenas as informações de alto-nível.

Todos os benefícios mencionados colaboram para a otimização do gerenciamento dos processos de negócio (Business Process Management — BPM).

Uma ferramenta de middleware popular open source para transmissão de dados em tempo real é o Apache Kafka.

O **Apache Kafka** foi desenvolvido para processar fluxos de dados provenientes de diversas fontes, bem como para entregá-los a vários clientes. Em resumo, essa plataforma movimenta volumes imensos de dados não apenas do ponto A ao ponto B, mas também de A à Z e para qualquer outro local que você precisar simultaneamente.

O Apache Kafka é uma alternativa aos sistemas de mensageria corporativos tradicionais. Inicialmente, ele era um sistema interno desenvolvido pela (LinkedIn) para processar 1,4 trilhão de mensagens por dia. Mas agora é uma solução de transmissão de dados open source aplicável as variadas necessidades corporativas.

Outro importante é o **Servidor de Aplicação Java EE**

Ele consiste em uma plataforma de programação para servidores amplamente utilizados e, como o próprio nome já diz, na linguagem Java.

Sua principal utilização é para o desenvolvimento de software em ambientes corporativos, com aplicações como web e rede, além de outros recursos seguros, escaláveis e multicamadas que facilitam muito o nosso dia a dia.

O Java EE é formado por uma série de contêineres e especificações com diferentes funcionalidades.

Segue abaixo os principais:

**• Java Database Connectivity (JDBC)**: Usado para acessar bancos de dados.

**• Enterprise Java Beans (EJBs):** Usados para desenvolver componentes de software. Graças a esses recursos, o programador consegue focar sua atenção nas necessidades de cada cliente, porque detalhes como escalabilidade, segurança, disponibilidade e infraestrutura ficam sob responsabilidade do servidor de aplicações.

**• Servelets:** Esse recurso é usado para desenvolver aplicações web com conteúdo dinâmico. Ele é composto por uma API que tem como objetivo fornecer ao programador os recursos do servidor web de uma forma mais simples e fácil de operar.

• **Java Transaction API (JTA):** Trata-se de uma API que busca padronizar o tratamento de transações em uma aplicação Java.

**• Java Server Pages (JSP):** Pode ser conceituado como uma especialização do servlet que possibilita ao programador desenvolver com mais facilidade conteúdos dinâmicos.

• **Java Persistence API (JPA):** Trata-se de uma API que tem como objetivo padronizar o acesso a banco de dados por meio de um mapeamento Objeto Relacional dos Enterprise Java Beans.

**• Java Connector Architecture (JCA):** Consiste em uma API que uniformiza a ligação a aplicações legadas.

• **Java Server Faces (JSF):** Trata-se de uma especificação Java que ajuda o programador na construção de interfaces de usuário a partir de componentes para aplicação em ambiente web.

**• Java Message Service (JMS):** Por meio desta API para middleware orientada a mensagens, torna-se possível a comunicação entre aplicações de forma assíncrona.

**Quais as principais vantagens?**

**1- Atualização do sistema:** Graças à utilização dos componentes corporativos, o programador consegue realizar alterações sem afetar os demais componentes da plataforma.

**2- Independência:** Esses componentes possibilitam a independência entre as linguagens utilizadas.

**3- Extensibilidade:** Para trazer mais funcionalidades e recursos, é possível remover e adicionar componentes a um aplicativo.

**4- Eficiência:** Alguns componentes agregam mais agilidade às telas de apresentação do sistema, tornando o desenvolvimento mais assertivo e ajudando na identificação da lógica e no acesso aos dados.

TAREFA 4

**ATIVIDADE:**

Dentro do processo de estilização de documentos HTML, utilizando naturalmente CSS, além da riqueza visual, também é esperado que eles respondam adequadamente aos mais variados tipos de dispositivos, como SmartTV, Smartphone, Tablet, computadores pessoais e outros. Para obtermos êxito neste requisito para a página WEB do sistema de detecção de fake news, podemos recorrer à um importante recurso chamado Media Query.

Explique de forma sucinta o funcionamento básico deste recurso, e como ele pode ajudar o sistema a se tornar responsivo nos diferentes tipos de dispositivos que for visualizado.

**O que é Media Query?**

É um recurso do CSS3 que permite a renderização de conteúdo para se adaptar a diferentes condições, como resolução da tela (por exemplo, tamanho da tela do celular e do desktop).

Ele se tornou um padrão recomendado pelo W3C em junho de 2012, e é uma tecnologia fundamental para o design responsivo da Web.

"Design Responsivo" é uma técnica que faz com que sua página "responda" para o navegador e dispositivo no qual o conteúdo é exibido... para parecer incrível não importe como.

Media Queries são as ferramentas mais poderosas para fazer isso.

Uma media query consiste de um media type e pelo menos uma expressão que limita o escopo das folhas de estilo usando media features, tal como largura, altura e cor.

Media queries, adicionadas no CSS3, deixam a apresentação do conteúdo adaptado a uma gama especifica de dispositivos não precisando mudar o conteúdo em si.

Uma media query consiste em uma consulta (query) que testa por medias e features, ou seja, é uma consulta que testa o meio e as características desse meio para, condicionalmente, aplicar determinados estilos.

Em outras palavras, uma media query é um teste que envolve media types e media features.

TAREFA 5

**ATIVIDADE:**

Ao desenvolver um aplicativo para Android e iOS, os desenvolvedores precisam trabalhar com plataformas e linguagens completamente diferentes, e isto podo comprometer todo o processo de desenvolvimento e atualização de um sistema mobile. Neste sentido, as empresas começaram a adotar soluções de plataforma cruzada sobre as soluções nativas para criar aplicativos para iOS e Android mais rapidamente usando um único idioma. Indique e explique uma plataforma de desenvolvimento multiplataforma que poderia ser utilizado para desenvolver o aplicativo mobile de fake news de forma mais ágil para Android e iOS.

**React Native**

O React Native é um framework baseado no já aclamado React, desenvolvido pela equipe do Facebook, que possibilita o desenvolvimento de aplicações mobile, tanto para Android, como para iOS, utilizando apenas Javascript.

O que tornar o React Native diferente, é que todo o código desenvolvido é convertido para a linguagem nativa do sistema operacional.

O React Native possui diversas características marcantes e fundamentais para sua ampla adoção, como podemos ver abaixo:

• Acessar a interface e os recursos nativos do Android e iOS utilizando JavaScript;

• O código produzido é semelhante ao React para Web;

• Possui a base de conhecimento compartilhada entre o desenvolvimento mobile e front-end;

• Todo código desenvolvido é convertido para a linguagem nativa do sistema operacional;

• Com o React Native conseguimos desenvolver aplicações para Android e iOS utilizando um código único;

• Por ser multiplataforma, podemos desenvolver aplicações com React Native utilizando qualquer sistema operacional (Windows, macOS ou Linux).

As principais vantagens de uma aplicação nativa sobre uma aplicação web mobile são:

• Experiência do usuário fluída;

• Carregamentos em geral mais rápidos;

• Melhor integração entre funções do celular como câmera, giroscópio etc.;

• Segurança superior;

• Melhor performance em geral.

O React Native possui diversas funcionalidades muito interessantes que aumentam a praticidade e a produtividade do desenvolvimento.

Uma delas é o Hot Reloading, que faz com que o programa fique rodando em desenvolvimento, e a cada atualização no código uma versão nova do arquivo modificado é injetado na aplicação, levando menos de 1 segundo para atualizar (mesmo com aplicações grandes e complexas).

Para o desenvolvimento mobile isso é um grande passo, pois em outros frameworks nativos, a cada mudança no código, a aplicação precisava ser recompilada por completo, levando muito mais tempo.

Existe também a possibilidade de depurar a aplicação via Google Chrome, como se fosse uma aplicação web padrão, o que ajuda muito os desenvolvedores web que já estão acostumados a utilizar o Dev Tools.

Outra feature (recurso) interessante do React Native é a possibilidade de combinar seu código Javascript com algum código nativo em Objective-C, Java ou Swift, caso você queira utilizar componentes já prontos, ou otimizar alguns aspectos da sua aplicação.

Após o anúncio do React Native, feito na ReactConf 2015 (maior conferência de React do mundo, que ocorre no escritório do Facebook, na Califórnia), o código do framework foi colocado no Github, com isso milhares de desenvolvedores se engajaram para contribuir com a ferramenta, que já ultrapassa 10.000 commits.

Diversos aplicativos de empresas gigantes já foram desenvolvidos com React Native como o Uber, AirBnB, Instagram, Baidu Mobile, Vogue App, Facebook Ads Manager e diversos outros, sendo o Facebook Groups o primeiro app feito 100% com o framework.

Mesmo ainda não sendo uma plataforma totalmente estável e ‘madura’, o React Native é sem dúvidas uma das maiores promessas para o desenvolvimento mobile em breve, pois o número de desenvolvedores web é crescente e é muito mais prático para esses desenvolvedores utilizar uma linguagem que já estão familiarizadas do que aprender linguagens e paradigmas de programação novos para criar um aplicativo mobile.

**7 - CONCLUSÃO**

Concluímos que o SCRUM ou as outras metodologias ágeis disponíveis são fundamentais para organizar e acelerar o desenvolvimento, diminuindo o tempo de entrega dos projetos, preparando o terreno para as constantes mudanças no projeto e diminuindo o retrabalho. Contando com a participação e “feedbacks” constante dos “stakeholders” e com o engajamento da equipe de desenvolvimento.

Concluímos também, que middlewars e os sistemas distribuídos é uma forma de ser ágil também nos aspectos relacionado a infraestrutura, pois reutilizando sistemas já difundidos ou contratando apenas os módulos necessários para seu projeto diminui o custo e tempo de desenvolvimento dos sistemas.

“Não é preciso inventar a roda ela já exite e é segura”.

Sobre o Media Query, continuamos na mesma linha de raciocínio que o desenvolvimento tem que ser ágil, e utilizando os diversos frameworks existentes nos preocupamos apenas em desenvolver o que o cliente precisa sem precisar inventar a roda.

Sobre React Native conseguimos visualizar que tem tudo para se tornar a principal plataforma para desenvolver android e mais uma vez nos deparamos com o aspecto Ágil onde escrevemos o código somente uma vez e ele já será convertido para as demais linguagens automaticamente.

O mundo de desenvolvimento é ágil!

**8 - REFERÊNCIAS BIBLIOGRÁFICAS**

PRESSMAN, R.; MAXIM, B. Engenharia de software. 8. ed. Porto Alegre: AMGH, 2016

https://www.organicadigital.com/blog/o-que-e-react-native/

https://www.treinaweb.com.br/blog/o-que-e-o-react-native/

<https://www.youtube.com/watch?v=mqltv3kFdgE>

https://www.youtube.com/watch?v=ItHnMnrIVh0

https://www.zup.com.br/

LINGUAGEM KOTLIN - <https://www.zup.com.br/blog/java-vs-kotlin-vantagens-desvantagens>

https://4linux.com.br/o-que-e-middleware/

**https://www.redhat.com/pt-br/topics/middleware/what-is-middleware**

https://pplware.sapo.pt/informacao/afinal-o-que-e-um-sistema-distribuido

https://www.youtube.com/watch?v=BSufWy0ZIeo

WAZLAVICK, Engenharia de software conceitos e praticas. e. ed. Elsevier, 2013

https://www.youtube.com/watch?v=dW\_lzLn96z0

TANENBAUM; STEEN. Sistemas Distribuídos: Princípios e Paradigmas ed. Pearson, 2007

https://www.youtube.com/watch?v=Oz15GwRK3\_I

GEORGE COULOURIS. Sistemas Distribuídos. ed. Bookman, 2013

O que é middleware - <https://4linux.com.br/o-que-e-middleware/>

Middleware - <https://pt.wikipedia.org/wiki/Middleware>