

**UNIVERSIDADE ANHANGUERA – UNIDERP**

POLO DE (Cotia-SP)

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

(MARCIANO JOSE DA COSTA – RA 24866488)

**PTI - PRODUÇÃO TEXTUAL INTERDISCIPLINAR**

Disciplinas Norteadoras:

**Engenharia de Software**

**Redes e Sistemas Distribuídos**

**Programação Web**

**Optativa I**

**Ed – Design Thinking**

Tutor (a) EAD: CRISTIANO GIROLDO

COTIA / SP 2021

SUMÁRIO

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

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

3 - TAREFA 2 - **O que é um Sistema Distribuido?** ..…….........................pg. 15

4 - TAREFA 3 – **O que é Web?**……………………………..............................pg. 22

5 – TAREFA 4 - **Plataformas e Linguagens para Desenv. Android**.…...pg. 27

6 – CONCLUSÃO..................................................................................…...pg. 32

7 – REFERÊNCIAS BIBLIOGRÁFICAS................................................…...pg. 33

-

INTRODUÇÃO

Tarefa 1:

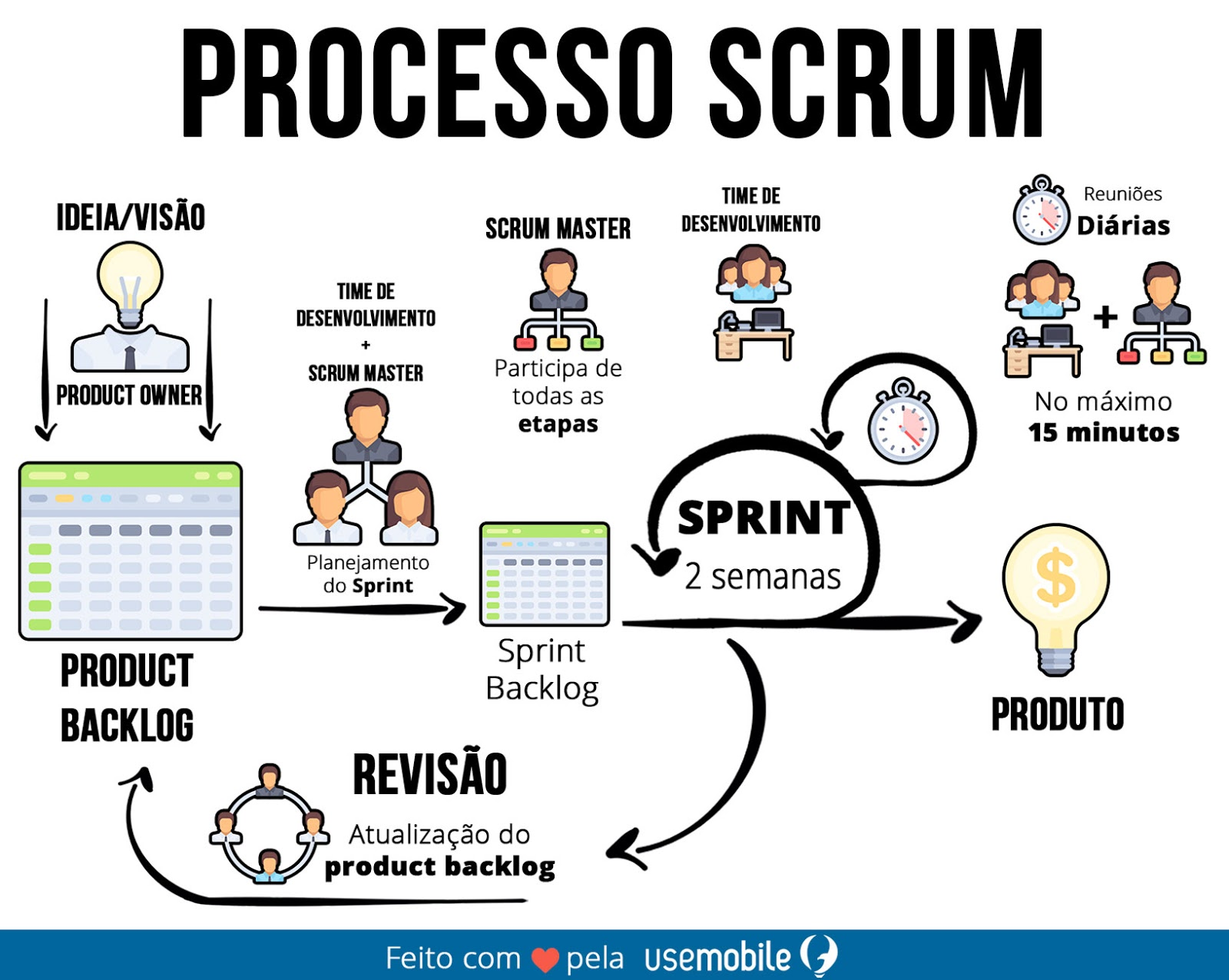
1. O primeiro ponto a ser abordado no relatório, será a escolha de uma metodologia ágil.Por isto, sua equipe deve discorrer sobre as características, valores, práticas, organização do time e entre outras características existentes para a metodologia ágil escolhida. Lembre-se de ressaltar o motivo da escolha realizada.

A T-Shirt deve utilizar as duas metodologias ageis em conjunto.

**O SCRUM e o XP(Extreme Programming).**

Ressalto que essa escolha foi embasada no modelo de negócio proposto de uma Startup, onde ainda é necessario modelar o negócio para aumentar sua ecalabilidade e utilzando o Scrum, as atividades serão divididas em partes menores e solucionadas dentro das sprints que vão de duas a 4 semanas sempre contando com as revisões por todos os participantes da equipe e uma contante revisão e acompanhamento das entregas. Por ser um novo negocio o XP é ideal pois com essa metodologia os produtos são entregues logo no inicio e melhorias vão sendo incrementadas com os feedbacks que são retornado dos clientes.

O **Scrum** é um método ágil que foi concebido como um estilo de gerenciamento de projetos em empresas de linha de produção sequenciais. A ideia era de que se fosse utilizado equipes pequenas e com várias disciplinas os resultados eram melhores. E assim é o Scrum as equipes são divididas e as tarefas também e tudo é controlado e constantemente revisado em busca de entregas rápidas e com qualidade em um processo iterativo e incremental.



**Scrum** é uma estrutura leve que ajuda pessoas, equipes e organizações a gerar valor por meio de soluções adaptativas para problemas complexos.

Em sintese,o Scrum requer um Scrum Master para promover um ambiente onde:

1- **Product Owner**: Ordena o trabalho de um problema complexo em um Product Backlog.

2- **Time Scrum**: Transforma uma seleção do trabalho em um incremento de valor durante uma Sprint.

3- **O Time Scrum e seus stakeholders**: Inspecionam os resultados e se ajustam para o próximo Sprint.

4- **Repita**

O **framework Scrum** é propositalmente incompleto, apenas definindo as partes necessárias para implementar a teoria Scrum, pois é baseado na inteligência coletiva das pessoas que o utilizam. Em vez de fornecer às pessoas instruções detalhadas, as regras do Scrum orientam seus relacionamentos e interações.

Vários processos, técnicas e métodos podem ser empregados na estrutura. Scrum envolve práticas existentes ou as torna desnecessárias. O Scrum torna visível a eficácia relativa do gerenciamento, ambiente e técnicas de trabalho atuais, para que melhorias possam ser feitas.

**Características:**

É um modelo ágil para a gestão de projetos, o qual tem, na reunião regular dos seus desenvolvedores para criações específicas, sua prática mais destacada. Suas práticas guardam semelhança com as próprias do XP, mas possuem nomes e graus de importância diferentes nos dois contextos.

**Product Backlog:** Trata-se da lista que contém todas as funcionalidades desejadas para o produto. O Scrum defende que tal lista não precisa estar completa logo na primeira vez em que é feita. *“Pode-se iniciar com as funcionalidades mais evidentes [...] para depois, à medida que o projeto avançar, tratar novas funcionalidades que forem sendo descobertas” (WAZLAVICK, 2013, p.56)*.

**Sprint Backlog:** Lista de tarefas que a equipe deverá executar naquele Sprint.

Tais tarefas são selecionadas do Product Backlog, com base nas prioridades definidas pelo Product Owner.

**Sprint:** O Scrum divide o processo de efetiva construção do software em ciclos regulares, que variam de duas a quatro semanas (ou em períodos maiores, a depender da complexidade das tarefas). Trata-se do momento em que a equipe se compromete a desenvolver as funcionalidades previamente definidas e colocadas no Sprint Backlog. Se alguma funcionalidade nova for descoberta, ela deverá ser tratada na Sprint seguinte. Cabe ao Product Owner manter o Sprint Backlog atualizado, apontando as tarefas já concluídas e aquelas a serem concluídas.

**Valores:**

O uso bem-sucedido do Scrum depende das pessoas se tornarem mais proficientes em viver cinco valores:

**Compromisso, Foco, Abertura, Respeito e Coragem**

O Time Scrum se compromete a atingir seus objetivos e apoiar uns aos outros. Seu foco principal é o trabalho da Sprint para fazer o melhor progresso possível em direção a essas metas. O Time Scrum e seus stakeholders são abertos sobre o trabalho e os desafios. Os membros da Equipe Scrum se respeitam para serem pessoas capazes e independentes, e são respeitados como tal pelas pessoas com quem trabalham. Os membros do Time Scrum têm a coragem de fazer a coisa certa, de trabalhar em problemas difíceis.

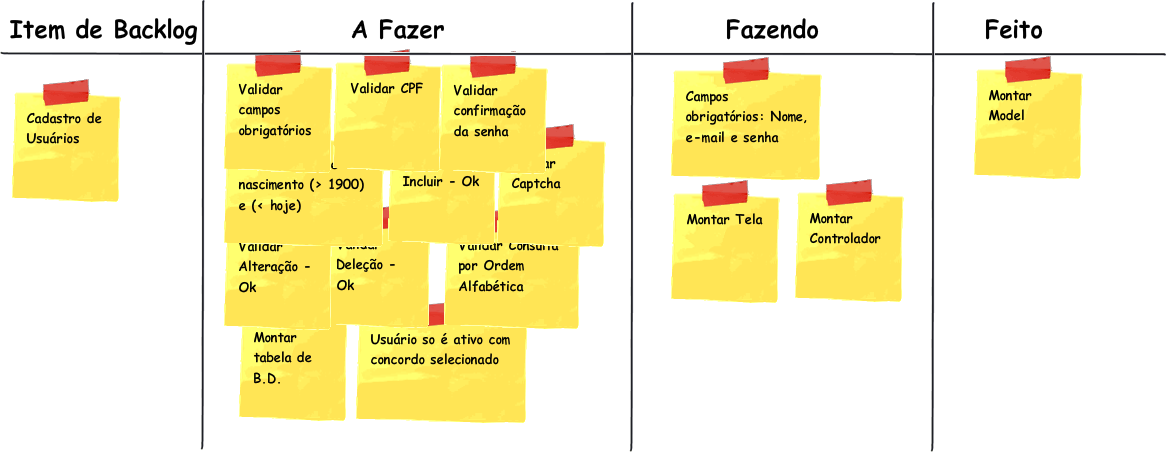
Esses valores orientam o Time Scrum em relação ao seu trabalho, ações e comportamento. As decisões que são tomadas, os passos dados e a forma como o Scrum é usado devem reforçar esses valores, não diminuí-los ou miná-los. Os membros do Time Scrum aprendem e exploram os valores enquanto trabalham com os eventos e artefatos do Scrum. Quando esses valores são incorporados pelo Time Scrum e pelas pessoas com quem trabalham, os pilares empíricos do Scrum de transparência, inspeção e adaptação ganham vida, construindo confiança.

**Práticas:**

Uma das praticas do Scrum mais interessantes é o quadro Srum:

O quadro Scrum é o meio pelo qual a equipe realiza a gestão visual das atividades do projeto.

Uma curiosidade é que por meio de uma análise mais detalhada do guia do framework Scrum revelará que o quadro não faz parte, oficialmente, da metodologia, porém sua adoção foi feita em larga escala pelas equipes e, aparentemente, esse fato não altera sua importância.



As divisões do quadro (daí ele conter desenho com formato matricial) são bem simples:

Uma coluna identifica a estória e as três colunas seguintes representam as tarefas relacionadas a esta estória que estão: a fazer (to do), em execução (doing) e feita (done).

Com essa disposição, cada linha do quadro representa uma estória e suas respectivas tarefas, que são, na verdade, extraídas do backlog do produto e que foram selecionadas para uma determinada Sprint.

Os papéis coloridos colados no quadro são post-its. Eles representam uma determinada característica ou estado daquela tarefa e contêm sua identificação resumida. O exemplo clássico é a do post-it vermelho, que pode representar uma tarefa de correção de defeitos em um código.

A critério da equipe, uma determinada cor pode identificar um determinado membro da equipe ao invés de utilizar papeis e materiais fisicos temos ferramentas computacionais colaborativas que facilitam bastante o controle de um projeto, e a uma delas é o **Trello**, cujo acesso é gratuito.

**Organização do time:**

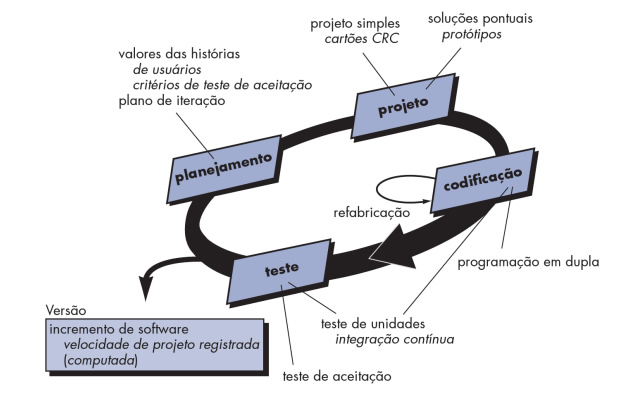
**Scrum Master**: Trata-se de um facilitador do projeto, um agente com amplo conhecimento do modelo e que preza pela sua manutenção durante todas as etapas do projeto. Deve atuar como moderador ao evitar que a equipe assuma tarefas além da sua capacidade de executá-las.

**Product Owner:** É a pessoa responsável pelo projeto propriamente dito. Ele tem a missão de indicar os requisitos mais importantes a serem tratados nos sprints.

**Scrum Team:** É a equipe de desenvolvimento, composta normalmente por um grupo de seis a dez pessoas. A exemplo do Extreme Programming, não há divisão entre desenvolvedor, analista e projetista (WAZLAVICK, 2013).

**O XP** **- Extreme Programming**

É uma metodologia ágil adequada para equipes pequenas e médias, onde os requisitos mudam com certa frequência e é indicada para projetos onde é necessário partes pré-prontas no início do projeto e que vão sendo refinadas com as sugestões e novas solicitações dos usuários.



**Características:**

O XP é uma metodologia adequada para empreitadas em que os requisitos mudam com certa constância, além de ser ajustado para equipes pequenas.

**Valores:**

Um trecho que demostra claramente os valores do XP dizendo que um produto é

Como uma obra de arte:

*"Escrever uma redação, um artigo ou um livro é uma atividade puramente intelectual que se caracteriza pela necessidade de sucessivas revisões e correções até que a obra adquira sua forma final. [...] Quando um pintor cria um novo quadro, é comum começar com alguns esboços, evoluir para uma representação mais próxima do formato final, fazer acertos, retoques e afins até que a obra esteja concluída. (TELLES, 2004. p. 39)"*

**Práticas:**

Para atingir seus objetivos, o Extreme Programming apoia-se em quatro pilares, comumente chamados de valores:

**Feedback:** Sua aplicação pretende alcançar a troca de informações entre cliente e equipe. Assim, quando o cliente aprende com o sistema que utiliza e reavalia suas necessidades, ele gera feedback para sua equipe de desenvolvimento.

**Comunicação:** O seu ideal é estabelecer contato proveitoso entre equipe e cliente, evitando que os desenvolvedores realizem o trabalho de forma especulativa, seguindo sempre um planejmanento com o foco nos requisitos especificados.

**Simplicidade:** Sua formulação serve para orientar a equipe a desenvolver apenas o que for suficiente para atender a necessidade do cliente, sem “sobrecarregar” o projeto com funções quase sempre inúteis.

**Coragem:** Ela será necessária no impulsionamento da equipe para manter sempre o cliente presente, para propor melhorias no que já foi testado e colocado em funcionamento e, de modo geral, para sempre levar adiante as práticas da metodologia.

**Organização do time:**

**Gerente do projeto:** Trata-se do responsável pelos assuntos administrativos do projeto e do relacionamento com o cliente. Duas de suas funções mais importantes incluem o estabelecimento de contato entre a equipe e o cliente e o cuidado para que a equipe fique livre de pressões externas e se dedique integralmente ao desenvolvimento do produto.

**Coach:** este é o nome do responsável técnico pelo projeto, que deve ser tecnicamente bem preparado e experiente. Se, por exemplo, uma nova tecnologia fica disponível no mercado, é função dele sugerir seu uso no produto em desenvolvimento.

**Analista de teste:** ajuda o cliente a escrever os testes de aceitação e fornece feedback para a equipe interna de modo que as correções no sistema possam ser feitas. Ao contrário do que é feito nas metodologias tradicionais, no desenvolvimento ágil em geral (e no XP em particular), o cliente participa ativamente dos testes no produto.

**Redator técnico:** ajuda a equipe de desenvolvimento a documentar o projeto, permitindo que os desenvolvedores estejam plenamente focados na construção do produto propriamente dito. À propósito, a metodologia recomenda que o projeto deve ser claro o suficiente para permitir uma documentação mínima.

**Desenvolvedor:** No XP, não há divisão entre as especialidades de engenheiro de requisitos, projetista e o desenvolvedor propriamente dito.

B) O segundo ponto a ser abordado no relatório, será a escolha de um modelo de maturidade.

Por tanto, sua equipe deverá descrever qual é o melhor modelo de maturidade para a empresa, além de comentar suas características. Neste ponto do relatório, discuta sobre até qual nível de maturidade pretende-se implantar na companhia, além de explicar quais serão os processos implantados a partir desta escolha. Lembre-se de ressaltar o motivo da escolha realizada.

Inicialmente destaco aqui alguns motivos importantes por se adotar um modelo de maturidade para os processos de software na T-Shirt:

Sommerville (2015) afirma que o mapeamento de processos é uma ferramenta gerencial que visa identificar a sequência na qual as atividades são executadas. Em termos práticos, o mapeamento pode gerar mais benefícios, como:

**Compreensão dos processos:** permite entender todas as partes que os compõem.

**Análise dos processos:** ao mapeá-los, é possível fazer uma reflexão sobre as atividades que os compõem.

**Melhoria dos processos:** enxergando detalhadamente as atividades que os compõem, eles podem ser otimizados, ajustados ou substituídos, a fim de que sejam melhorados.

**Padronização dos processos:** as atividades que possuem níveis de qualidade ajustados com as políticas da empresa dentro dos processos podem vir a se tornar padrão.

**Documentação dos processos:** ao mapear as atividades que compõem os processos, é possível documentá-los, se assim a equipe não tiver feito nas atividades que precedem a fase de desenvolvimento do projeto.

Agora abordando os pricipios de Maturidade de Processos:

**O QUE SÃO MODELOS DE MATURIDADE DE PROCESSOS?**

Os modelos de maturidade são ótimas ferramentas que permitem o conhecimento profundo dos processos, da equipe de desenvolvimento, das dificuldades, entre outros.

Temos os modelos de maturidade CMM e CMMI os quais são metodologias que visam, por meio de sua estrutura, posicionar a empresa em um nível de maturidade e, através de técnicas, planejar uma evolução em termos organizacionais a fim de promover melhoria em seus processos.

Um ponto importante sobre a aplicação de uma metodologia de maturidade é que os modelos de maturidade são ótimas ferramentas que permitem o conhecimento profundo dos processos, da equipe de desenvolvimento, das dificuldades, entre outros. No entanto, colocar tais metodologias em prática exige comprometimento e persistência, pois, em muitos casos, a resistência de colaboradores que nunca tiveram contato com as normas e diretrizes podem criar certo bloqueio em sua adoção.

Agora com o olhar voltado para a capacidade dos processos, temos o framework CMMI (Capability Maturity Model Integration). Também desenvolvido pela SEI, em 2012 foi lançada a sua última versão (1.3).

O principal objetivo dessa ferramenta é analisar a capacidade da maturidade do processo .

A estrutura do CCMI é dívida em cinco níveis, os quais são utilizados para determinar a capacidade e a maturidade dos processos:

**Nível 0 (Incompleto):** os processos não são executados em sua totalidade ou são parcialmente executados.

**Nível 1 (Executado):** os processos conseguem satisfazer metas específicas, e a organização como um todo reconhece que existem processos definidos.

**Nível 2 (Gerenciado):** nessa fase, além dos processos serem executados, existe também um monitoramento e um planejamento deles.

**Nível 3 (Definido):** o processo serve como boa prática e se torna padrão dentro da organização.

**Nível 4 (Gerenciado):** são utilizadas medições e técnicas de estatística para análise dos processos, o que gera, consequentemente, melhorias.

**Nível 5 (Otimizado):** o foco é a melhoria contínua para a busca de melhores resultados.

**NORMAS ISO DE QUALIDADE DE PROCESSOS**

Temos também a **ISO 9001:2015**, cujo foco está em utilizar metodologias que permitam aos membros da organização conhecer os processos, os microprocessos e as interações entre os componentes em sua totalidade. Dessa forma, ao se conhecer bem os processos, a chance de erros e falhas é minimizada, proporcionando a melhoria dos processos e consequentes benefícios.

A **ISO 9001:2015** possui uma característica fundamental quanto à visão sistêmica que deve reger as atividades de desenvolvimento. Ou seja, os profissionais devem conhecer os processos, e esse conhecimento deve ser bem consistente entre os membros da organização como um todo.

A **ISO 9001** é um sistema de qualidade que visa à garantia de otimização dos processos. Essa norma também é conhecida nos meios profissionais como **SQA (Sistema de Gestão da Qualidade)**.

Para os gestores de projetos de desenvolvimento de software é um poderoso instrumento de correção e de melhoria de processos.

A ISO 9001 é estruturada por meio de oito princípios da qualidade, conforme pode ser observado a seguir:

**Foco no cliente - Liderança - Envolvimento das pessoas - Abordagem do processo - Abordagem sistêmica para a gestão - Melhoria contínua - Abordagem para tomada de decisão - Benefícios**;

**MELHORIA DE PROCESSOS INDIVIDUAIS E DE EQUIPE (PSP)**

**Personal Software Process**

Agora voltando o olhar para o desenvolvimento das pessoas temos a metodologia para melhoria de processos individuais e de equipe (PSP).

Trata-se de técnicas bem fundamentadas e voltadas às pessoas (desenvolvedores), pois quem, de fato, consegue abstrair informações e codificá-las são as pessoas.

Koscianski e Soares (2006) define que, o PSP é composto por quatro níveis de competência conforme pode ser observado a seguir:

**PSP 0 – Processo atual:** aqui deve ser estabelecida a base da competência, que inclui a determinação dos métodos de medição, o que medir e a que momento (desenvolvimento, compilação e teste), permitindo, assim, o desenvolvimento de relatórios para análise de falhas.

**PSP 1 – Estimativa de tamanho:** nessa fase existem apenas dois objetivos: o planejamento do tempo e das tarefas. Trata-se de uma fase de planejamento pessoal, na qual se deve ter uma estimativa de quantas tarefas estão atribuídas ao colaborador e o tempo de desenvolvimento delas.

**PSP 2 – Revisão:** o enfoque dessa fase é o processo de administração da qualidade pessoal. Nela a visão técnica da tecnologia, por meio da qual está se desenvolvendo, deve observar a aplicação dos processos estabelecidos e os resultados positivos e negativos.

**PSP 3 – Desenvolvimento cíclico:** trata-se de um processo pessoal cíclico, no qual a correta aplicação dos processos, que atendem as demandas e a qualidade, seja processada em cascata, a fim de se obterem os melhores resultados.

**APLICAÇÃO A TSHIRT**

O modelo de maturidade que pode ser aplicado a TShirt é o modelo de melhoria de processo de software brasileiro (MPS.BR), ferramenta muito interessante, pois permite compreender a potencialidade do método de forma que possa ser aplicado em organizações brasileiras.

**MELHORIA DE PROCESSOS DE SOFTWARE BRASILEIRO (MPS.BR)**

Aqui no Brasil, nós possuímos uma ferramenta que possibilita a melhoria dos processos de softwares brasileiros. Porém, isso não significa que as suas normas sejam melhores ou piores que outras, pois estas seguem padrões internacionais de qualidade de processos. Aqui no País, a entidade que faz a gestão do MPS.BR é a Softex, responsável por apoiar a cultura da qualidade de software e por contribuir com a melhoria contínua dos processos/produto de software.

**A SOFTEX**

" A Softex – Organização Social Civil de Interesse Público (OSCIP) atua há 24 anos em prol do fomento da Transformação Digital Brasileira, criando, promovendo e executando iniciativas no âmbito nacional e internacional nas áreas de tecnologia e inovação.

Com sede em Brasília/DF e em Manaus/AM, possui 6 Áreas, 21 Agentes Regionais,

1 aceleradora exclusiva e mais 19 parceiras e 34 Instituições de Ciência e Tecnologia. Beneficia a aceleração de mais de 5 mil startups e cerca de 6 mil empresas.

Atua em todos os estados do Brasil e em 20 países nos setores Industrial, Agronegócio, Comércio e Serviços e Logística nas verticais Gás e Óleo, Telecomunicações, Saúde, Aeroespacial, Automotivo, Segurança."

Segundo Rocha (2005), a construção das técnicas constituintes ao MPS.BR é composta pelas NBR ISO/IEC:

Para a compreensão das siglas que compõem o MPS.BR, acompanhe os tópicos a seguir:

**Modelo de Referência (MR):** contém informações a forma como a organização deve conduzir os seus processos para atingir os resultados. Ainda é possível, nesse mesmo documento, determinar o nível de maturidade e da capacidade dos processos descritos na NBR ISO/IEC 12207.

**Modelo de Avaliação (MA):** trata-se do processo no qual são determinados os parâmetros e requisitos para se aferir a qualidade do desenvolvimento. É baseado na norma ISO/IEC 15504.

**Modelo de Negócio (MN):** determina o nível de maturidade dos processos, que podem ser: (A) Otimizado, (B) Gerenciado, (C) Definido, (D) Largamente definido, (E) Parcialmente definido, (F) Gerenciado e (G) Parcialmente gerenciado.

Vale ressaltar que cada modelo apresenta o seu formulário específico: MR (Guia Geral e Guia de Aquisição), MA (Guia de Avaliação) e MN (Documentos do projeto).

Segundo Rocha (2005), a construção das técnicas constituintes ao MPS.BR é composta pelas NBR ISO/IEC:

**NBR ISO/IEC 12207** para processo de ciclo de vida de software.

**ISO/IEC 15504** para avaliação de processo.

**ISO/IEC 15504-5** para modelo de avaliação de processo de software.

Na prática, como o MBS.BR deve ser iniciado no nível G, para o qual o manual determina 28 GPR, que são os objetivos que a equipe deve alcançar para atingir de maturidade.

Tarefa 2:

Desta forma, sua tarefa é apresentar um relatório contendo algumas informações importantes a respeito de sistemas distribuídos, de acordo com os tópicos listados a seguir:

1. Faça um resumo a respeito das principais vantagens ao se implementar um sistema

distribuído ao invés de um sistema centralizado;

**O QUE É UM SISTEMA DISTRIBUÍDO?**

*“Um sistema distribuído é um conjunto de computadores que são interligados via rede, porém, para o usuário final das aplicações que são executadas através deles, aparenta ser um sistema único, como uma única máquina ou um único software (TANENBAUM; STEEN, 2017).”*

**

Mesmo que não saiba, hoje mesmo você já deve ter acessado um sistema distribuído!

Ao abrir o navegador de sua preferência e acessar uma página de internet, você está usando um sistema distribuído.

Essa simples ação rotineira em nosso dia a dia, por meio de um smartphone ou computador, utiliza um sistema distribuído.

Um de seus principais aspectos é que os computadores que fazem parte de sistemas distribuídos têm o funcionamento independente, ou seja, cada um age por si próprio e, muitas vezes, os sistemas e hardwares dessas máquinas são totalmente diferentes, mas aparentam ser uma coisa só para o usuário.

Esse conceito faz com que o sistema não dependa apenas de uma máquina, pois toda a carga do ambiente estará distribuída entre várias máquinas. Assim, os sistemas distribuídos ocultam o conjunto de máquinas que o fazem funcionar, aparentando ser algo único.

A principal motivação para construir e utilizar sistemas distribuídos é proveniente do desejo de compartilhar recursos. O termo *“recurso”* é bastante abstrato, mas caracteriza o conjunto de elementos que podem ser compartilhados de maneira útil em um sistema de computadores interligados em rede.

Ele abrange desde componentes de hardware, como discos e impressoras, até entidades definidas pelo software, como arquivos, bancos de dados e objetos de dados de todos os tipos.

**COMPARTILHAMENTO DE RECURSOS**

O compartilhamento de recursos refere-se à capacidade do sistema em compartilhar o acesso a quaisquer recursos utilizados por ele entre as máquinas que fazem parte da arquitetura (também chamadas de nós).

Esses recursos são, na maioria das vezes, bancos de dados, links de rede que se conectam à internet, serviços de autenticação, entre outros.

**CONFIABILIDADE**

Está relacionada ao funcionamento adequado, conforme foi planejado.

A confiabilidade nos sistemas distribuídos é maior que nos sistemas centralizados, no entanto qualquer problema relacionado aos processos ou ao canal de comunicação/transmissão pode surtir efeitos diretos sobre a execução do sistema.

**DESEMPENHO**

Aumentar o desempenho de um sistema também é um objetivo dos sistemas distribuídos. Se fizermos uma comparação, os sistemas distribuídos, na maioria dos casos, apresentam melhor desempenho do que os sistemas centralizados. Isto ocorre porque, em um sistema distribuído, temos múltiplas instâncias, tanto de hardware quanto de software, para realizar o processamento necessário.

**Vantagens de um sistema distribuído, se comparado a um sistema centralizado.**

A primeira vantagem que podemos observar diz respeito à economia.

Um sistema de informação centralizado de grande porte necessita de um mainframe para que ele possa funcionar com qualidade.

Quando utilizamos um sistema distribuído em substituição a um sistema centralizado, podemos substituir o mainframe por vários micro computadores. Feita essa distribuição de processamento entre vários computadores de menor porte, é acrescentado um poder computacional de processamento de baixo custo, deixando claro que em algumas situações, a substituição de um mainframe por vários microcomputadores de menor porte, é mais viável.

A segunda vantagem diz respeito à velocidade, por maior que seja a velocidade de processamento de um mainframe, ele nunca conseguirá alcançar o poder de processamento de N microcomputadores interligados, como se fosse um único sistema.

A terceira vantagem está relacionada à confiança oferecida pelo sistema.

Se um dos microcomputadores que fazem parte de um sistema distribuído deixar de funcionar, o sistema como um todo continuará em operação, deixando de funcionar apenas alguns módulos que eram de responsabilidade do computador defeituoso.

A quarta vantagem diz respeito ao poder de crescimento do sistema.

Quando utilizamos um sistema de informação distribuído, podemos aumentar o seu poder computacional sem um custo exorbitante. Se compararmos a um sistema de informação centralizado, para aumentar o poder computacional deste sistema, será necessário substituir o mainframe utilizado ou mesmo duplicá-lo, tornando esse tipo de operação economicamente inviável.

Por último, algumas aplicações necessitam por natureza serem inerentemente distribuídas, proporcionando aos sistemas distribuídos vantagens sobre os sistemas centralizados.

**RESUMO DAS VANTAGENS DE UM SISTEMA DISTRIBUIDO:**

• Melhor relação custo/benéfico;

• Maior capacidade de processamento;

• Maior domínio de aplicações;

• Maior confiabilidade;

• Maior disponibilidade;

• Crescimento gradativo de sua capacidade de processamento.

2. Defina Middleware e explique qual o seu papel em um Sistema Distribuído;

Os sistemas distribuídos podem ser considerados como uma solução mais robusta em resposta aos sistemas puramente de rede, isso graças ao componente conhecido como middleware. Ele é um dos fatores principais para o bom funcionamento de aplicações distribuídas.

*"Middleware é uma camada oculta, um software, o qual se encontra entre os sistemas operacionais e os aplicativos (programas criados) que são executados neles. Portanto, é uma camada central, que permite o gerenciamento e a comunicação de dados para o funcionamento de aplicativos distribuídos. O middleware funciona como uma camada de tradução para interligar o sistema operacional com os programas (COULOURIS et al., 2013)."*

**EXEMPLOS:**

Alguns exemplos são: middlewares de banco de dados e servidores WEB e ferramentas de mensageria.

As mais diversas redes sociais que utilizamos hoje em dia, com conteúdos multimídias, são exemplos de sistemas distribuídos, assim como sites de pesquisas e plataformas de vídeos on line.

Um middlewares liga aplicações com os sistemas operacionais. Essas aplicações e sistemas operacionais estão em diferentes computadores.

Eles interagirão entre si para a execução de um sistema distribuído, e isso só é possível devido à camada do sistema distribuído, chamada middleware.

O middleware facilita aos desenvolvedores de software implementarem comunicação e entrada/saída, de forma que eles possam focar no propósito específico de sua aplicação.

"Middleware é o software de computador que fornece serviços para softwares aplicativos além daqueles disponíveis pelo sistema operacional. Pode ser descrito como "cola de software""

É utilizado para mover ou transportar informações e dados entre programas de diferentes protocolos de comunicação, plataformas e dependências do sistema operacional. É geralmente constituído por módulos dotados com APIs de alto nível que proporcionam a sua integração com aplicações desenvolvidas em diversas linguagens de programação e interfaces de baixo nível que permitem a sua independência relativamente ao dispositivo.

Seu objetivo é mascarar a heterogeneidade e fornecer um modelo de programação mais produtivo para os programadores de aplicativos.

É composto por um conjunto de processos ou objetos em um grupo de computadores, que interagem entre si de forma a implementar comunicação e oferecer suporte para compartilhamento de recursos e aplicativos distribuídos.

O Middleware é a designação genérica utilizada para referir aos sistemas de software que se executam entre as aplicações e os sistemas operacionais.

O objetivo do middleware é facilitar o desenvolvimento de aplicações, tipicamente as distribuídas, assim como facilitar a integração de sistemas legados ou desenvolvidos de forma não integrada automática

**OUTROS EXEMPLOS**:

Qualquer coisa entre o Núcleo e aplicações dos usuários é considerado middleware.

O termo middleware também é usado em outros contextos. Às vezes, o middleware é usado em um sentido semelhante a um driver de software, uma camada de abstração que oculta detalhes sobre dispositivos de hardware ou outro software de um aplicativo.

O sistema operacional Android usa o kernel do Linux em sua essência e também fornece um framework de aplicativos que os desenvolvedores incorporam a seus aplicativos. Além disso, o Android fornece uma camada de middleware, que inclui bibliotecas que fornecem serviços como armazenamento de dados, exibição na tela, multimídia e navegação na web.

Como as bibliotecas de middleware são compiladas na linguagem de máquina, os serviços são executados rapidamente.

As bibliotecas de middleware também implementam funções específicas do dispositivo, portanto, os aplicativos e a estrutura do aplicativo não precisam se preocupar com variações entre vários dispositivos Android.

A camada de middleware do Android também contém a máquina virtual ART e suas principais bibliotecas de aplicativos Java.

O middleware também se refere ao software que separa duas ou mais APIs e fornece serviços como limitação de taxa, autenticação e log.

Às vezes, softwares de mecanismos de jogos como Gamebryo e RenderWare são descritos como middleware, porque fornecem muitos serviços para simplificar o desenvolvimento de jogos.

Em tecnologia de simulação, o middleware é geralmente usado no contexto da arquitetura de alto nível (HLA) que se aplica a muitas simulações distribuídas.

É uma camada de software que fica entre o código do aplicativo e a infraestrutura de tempo de execução.

O middleware geralmente consiste em uma biblioteca de funções e permite que vários aplicativos - simulações ou federados na terminologia HLA - paginem essas funções da biblioteca comum em vez de recriá-las para cada aplicativo.

Os desenvolvedores de redes sem fio podem usar o middleware para enfrentar os desafios associados a uma rede de sensores sem fio (WSN).

A implementação de um aplicativo de middleware permite que os desenvolvedores do WSN integrem sistemas operacionais e hardware com a grande variedade de aplicativos disponíveis no momento.

O sistema operacional QNX oferece middleware para fornecer serviços de multimídia para uso em automóveis, aeronaves e outros ambientes.

Os kits de ferramentas do software de identificação por radiofrequência (RFID) fornecem middleware para filtrar dados brutos e redundantes e ruidosos.

3. Faça um relatório de pelo menos 2 exemplos de plataformas de middlewares mais

utilizadas, explicando suas principais características.

**Entre tantos, são exemplos de Middleware os grupos de tecnologias abaixo:**

**CORBA - Enterprise Serial Bus - Servidores de Aplicação Java EE - Business intelligence - BPM (Business process management) - Mensageria - BAM (Business activity monitoring) - Monitoramento de IT - Ginga**

**CORBA:**

CORBA (abreviado de Common Object Request Broker Architecture) é a arquitetura padrão criada pelo Object Management Group para estabelecer e simplificar a troca de dados entre sistemas distribuídos heterogêneos.

Em face da diversidade de hardware e software que encontramos atualmente, a

CORBA atua de modo que os objetos (componentes dos softwares) possam se comunicar de forma transparente ao usuário, mesmo que para isso seja necessário interoperar com outro software, em outro sistema operacional e em outra ferramenta de desenvolvimento.

CORBA é um dos modelos mais populares de objetos distribuídos.

A arquitetura CORBA define o ORB (Object Request Broker) como um módulo intermediário entre cliente e objeto, sendo responsável em aceitar a requisição do cliente, enviá-la para o objeto competente e, assim que disponível a resposta, entregá-la.

A CORBA utiliza a IDL (Interface Definition Language), uma linguagem baseada em C++ que não possui algoritmos nem variáveis, ou seja, é puramente declarativa, e, portanto, é independente da linguagem de programação utilizada para acessá-la. Há padrão de IDL definido pelo OMG para C, C++, Java, TTCN, COBOL, Smalltalk, Ada, Lisp, C#, Python e IDLscript. Possibilita a interoperabilidade entre os diversos sistemas, visto a separação que é definida entre interface e execução. A interface de cada objeto é definida de forma bastante específica, enquanto a sua execução (código fonte e dados) permanece oculta para o resto do sistema.

Ao contrário dos objetos tradicionais, os objetos em sistemas distribuídos possuem uma característica de dualidade: um estado dinâmico, tipicamente alocado em memória volátil (em tempo de execução), e um estado persistente, que não pode ser destruído após o encerramento do programa que os criou e que pode ser usado para reconstruir o estado dinâmico, devendo ser armazenado em memória não volátil, seja em sistema de arquivos ou banco de dados.

A arquitetura CORBA, para prover a persistência, define o Persistent Object Service (POS) como sendo responsável por armazenar o estado persistente dos objetos, utilizando quatro elementos:

1- Objetos Persistentes (Persistent Object (POs))

2- Gerenciador de Objetos Persistentes (Persistent Objects Manager (POM))

3- Serviços de Persistência de Dados (Persistent Data Services (PDSs))

4- Base de Dados (Datastores)

**ENTERPRISE SERVICE BUS:**

O Enterprise Service Bus (ESB) se refere à arquitetura de construção de software tipicamente implementado em tecnologias encontradas na categoria de produtos de infra-estrutura de middleware.

Normalmente baseado no reconhecimento de padrões, que fornecem uma base de serviços para arquiteturas mais complexas via um driver de evento e padrões baseados em mensagens (BUS).

Um ESB geralmente fornece uma abstração de camadas na implementação de um sistema empresarial de mensagens, que permita integração da arquitetura para explorar o valor das mensagens sem escrever código. Contrariando a clássica integração de aplicações comerciais (EAI).

A base de um enterprise service bus é construída da quebra de funções básicas em partes, que são distribuídas onde for preciso.

ESB não implementa uma arquitetura orientada a serviço (SOA), mas fornece as características para que possa ser implementado. ESB não necessariamente precisa ser implementado usando web services. ESB devem ser baseados em padrões flexíveis, suportando vários meios de transporte. Baseado no EAI melhor que padrões SOA, ele tenta remover o acoplamento entre o serviço chamado e o meio de transporte.

**Arquitetura Orientada a Serviços com a utlização de um EBS**

Um barramento de serviços proporciona desacoplamento entre consumidores e provedores de serviços, já que os consumidores não conhecem diretamente quem está provendo o serviço. Alterações em tais provedores de serviços podem impactar diretamente o ESB, mas em geral são transparentes para os consumidores de serviços.

Tarefa 3:

Neste sentido, sua empresa deve elaborar um documento inicial contento os principais tópicos:

1. O que é a WEB? Introduza os conceitos fundamentais de que é a WEB e qual sua

importância desde que foi criada.

**O que é Web:**

Web é uma palavra inglesa que significa teia ou rede. O significado de web ganhou outro sentido com o aparecimento da internet. A web passou a designar a rede que conecta computadores por todo mundo, a World Wide Web (WWW). Web pode ser uma teia de aranha ou um tecido e também se utiliza para designar uma trama ou intriga.

A web significa um sistema de informações ligadas através de hipermídia (hiperliga-ções em forma de texto, vídeo, som e outras animações digitais) que permitem ao usu-ário acessar uma infinidade de conteúdos através da internet. Para tal é necessário ligação à internet e um navegador (browser) onde são visualizados os conteúdos dis-poníveis. São exemplos de navegadores: Google Chrome, Safari, Mozilla Firefox, In-ternet Explorer, Opera, etc.

A internet revolucionou nossa forma de comunicação e relacionamento social. Trans-formou profundamente o modo como interagimos, seja em nossas famílias ou nos ou-tros grupos sociais que pertencemos. Alterou como vivemos, aprendemos, trabalha-mos, consumimos e nos divertimos. A internet trouxe benefícios na utilização das tec-nologias com fácil acesso ao conhecimento, na colaboração entre as pessoas e orga-nizações, na inclusão social, e na criação de valores.

No início da internet, seu conteúdo era produzido por poucos. Somente aqueles que conheciam as técnicas de programação podiam produzir conteúdo. A maioria dos in-ternautas podiam somente navegar entre as páginas. Nos últimos anos, isso mudou radicalmente, novas ferramentas foram criadas para facilitar a criação de conteúdo e compartilhar de maneiras muito simples, seja por meio de blogs, redes sociais, ou ví-deos no YouTube. Essa interação entre os internautas foi muito enriquecedora para a internet, fortalecendo seu campo de penetração em todos os lugares. Vemos pessoas em todos os cantos do mundo criando blogs ou fotoblogs para manter contatos com seus parentes e amigos mais distantes ou simplesmente compartilhar ideias com o mundo inteiro, sem mais a limitação geográfica de antigamente, podendo simples-mente ser tudo feito de um aparelho celular.

As crianças e os adolescentes são hoje o grupo quem mais se destaca em seu uso. A internet se tornou muito útil em agregar conhecimento e interatividade. É possível encontrar um infinidade de coisas úteis, divertidas e atraentes. E para os mais jovens é um grande ferramenta de interação social.

**Alguns dados nos chama a atenção:** 45% da população brasileira acessa a internet continuamente. Desse grupo, mais de 30% são crianças e adolescentes. São quase 27 milhões de brasileiros abaixo dos 17 anos acessam a internet constantemente, na grande maioria sem orientação de seus pais. O local mais comum de acesso é dentro de nossas casas, seguido de acesso em lan houses e centros comunitários de acesso gratuito, como igrejas ou bibliotecas públicas.

2. Descreva de forma sucinta o processo completo para acesso de sites na WEB. Ou seja,sua tarefa será descrever, passo a passo, todas as etapas que ocorrem nesse processo de comunicação entre a máquina cliente (usuário) e a máquina servidor (site) até o momento em que o site aparece ao usuário.

**Como a Internet funciona?**

Computadores conectados à web são chamados clientes e servidores.

**Clientes:** são os típicos dispositivos conectados à internet dos usuários da web (por exemplo, seu computador conectado ao seu Wi-Fi ou seu telefone conectado à sua rede móvel) e programas de acesso à Web disponíveis nesses dispositivos (geralmente um navegador como Firefox ou Chrome).

**Servidores**: são computadores que armazenam páginas, sites ou aplicativos. Quando o dispositivo de um cliente quer acessar uma página, uma cópia dela é baixada do servidor para a máquina do cliente para ser apresentada no nave-gador web do usuário.

Podemos imaginar que a web é uma estrada. Em um extremo da estrada, temos o cliente, que é como sua casa. No extremo oposto, temos o servidor, que é como uma loja onde você quer comprar algo.

Além do cliente e do servidor, também precisamos ter orquestrado:

**Sua conexão de Internet:** permite que você mande e receba dados na web.

É basicamente como a rua entre sua casa e a loja.

**TCP/IP**: Protocolo de Controle de Transmissão e Protocolo de Internet (Trans-mission Control Protocol e Internet Protocol) são protocolos de comunicação que definem como os dados trafegam pela web. São como os mecanismos de transporte que te permitem ir ao shopping, fazer um pedido e comprar seus produtos. Em nosso exemplo, é como um carro ou uma bicicleta (ou qualquer outra coisa que você possa usar).

**DNS:** Servidor de Nome de Domínio (Domain Name Servers) são como um ca-tálogo de endereços para sites. Quando você digita um endereço web no seu navegador, o navegador procura no servidor de DNS para localizar o endereço real do site, antes que ele possa recuperar o site. O navegador precisa encontrar em qual servidor web a página está hospedada para que ele possa mandar mensagens HTTP ao lugar certo (veja abaixo). Isso é como pesquisar o endere-ço da loja para que você possa entrar em contato.

**HTTP:** Protocolo de Transferência de Hypertexto (Hypertext Transfer Protocol) é um protocolo de aplicação que define uma linguagem para clientes e servidores se comunicarem entre si. É como a linguagem que você usa para encomendar seus produtos.

**Arquivos componentes:** um website é feito de muitos arquivos diferentes, que são como as diferentes partes dos produtos que você comprou na loja. Esses arquivos são divididos em dois tipos principais:

**Arquivos de Código:** os sites são feitos principalmente de HTML, CSS e JavaScript, embora você possa conhecer outras tecnologias mais tarde.

**Recursos:** esse é o nome coletivo para todas as outras coisas que com-põem um site, como imagens, música, vídeos, documentos do Word e PDFs.

**O que acontece quando você digita um endereço da web no seu navegador?**

I.O navegador vai para o servidor de DNS e encontra o endereço verdadeiro de onde o site está hospedado.

II.O navegador manda uma mensagem de requisição HTTP para o servidor, pedindo que envie uma cópia do site ao cliente.

Esta mensagem e todos os outros dados enviados entre o cliente e o servidor são enviados pela sua conexão à internet usando TCP/IP.

III.Se o servidor aprovar a requisição do cliente, o servidor enviará ao cliente uma mensagem "200 OK", que significa "Claro que você pode ver esse site! Aqui está" e então começa a enviar os arquivos do site para o navegador como uma série de pequenos pedaços chamados pacotes de dados.

IV.O navegador monta os pequenos pedaços em um site completo e o mostra a você o site que pediu.

3. HTML5 e CSS3 são as versões mais utilizadas destas duas linguagens de marcação.

Indique de forma resumida quais as principais melhorias e novas possibilidades no

desenvolvimento WEB que surgiram com estas novas versões.

**Novidades em HTML5**

HTML5 foi lançado com o objetivo principal de melhorar a experiência da World Wide Web para os desenvolvedores e os usuários finais.

A maior vantagem que o HTML5 tem sobre seu antecessor é que ele tem suporte de áudio e vídeo de alto nível que não fazia parte das especificações de versão em HTMLs anteriores.

Depois temos o SVG, canvas e outros gráficos vetoriais virtuais são suportados em HTML5, enquanto que em HTML, usando gráficos vetoriais só era possível usando-se em conjunto com diferentes tecnologias como Flash, VML e Silverlight, etc.

O HTML5 usa bancos de dados SQL da Web, cache de aplicativos para armazenamento temporário de dados, enquanto isso, em HTML, apenas o cache do navegador pode ser utilizado para esse fim.

Outra diferença entre HTML e HTML5 vale a pena mencionar é que a primeira não permite que o JavaScript funcione no navegador da Web (ele roda no segmento da interface do navegador), enquanto que o último oferece suporte completo para o JavaScript ser executado em segundo plano Cortesia para a API JS web worker de HTML5).

O HTML5 não é baseado no SGML, e isso permite que ele tenha regras de análise aprimoradas que proporcionam compatibilidade aprimorada.

Em HTML5, MathML inline e SVG podem ser usados ​​no texto enquanto que isto não era possível em HTML.

Alguns dos elementos deprecated que foram deixados completamente são: isindex, noframes, acrônimo, applet, basefont, dir, fonte, quadro, frameset, grande, centro, batida, tt.

O HTML5 suporta novos tipos de controles de formulário, por exemplo: datas e horas, e-mail, número, intervalo, tel, url, pesquisa etc.

Há muitos novos elementos introduzidos em HTML.

Alguns dos mais importantes são: sumário, tempo, de lado, áudio, comando, dados, datalist, detalhes, embed, wbr, figcaption, figura, rodapé, cabeçalho, artigo, hgroup, bdi, Nav, saída, progresso, rp, rt, ruby, seção, fonte, pista e vídeo.

**Novidades do CSS3**

O CSS3 traz muitas propriedades novas.

As principais novidades são:

**Bordas**

border-color

border-image

border-radius

box-shadow

**Fundos**

background-origin

background-clip

background-size

fazer camadas com múltiplas imagens de fundo

**Cor**

cores HSL

cores HSLA

cores RGBA

Opacidade

**Texto**

text-shadow

text-overflow

Ruptura de palavras longas

**Interface**

box-sizing

resize

outline

nav-top, nav-right, nav-bottom, nav-left

**Seletores**

Seletores por atributos

**Modelo de caixa básico**

overflow-x, overflow-y

**Outros**

media queries

criação de múltiplas colunas de texto

propriedades orientadas a discurso ou leitura automática de páginas web

Web Fonts

Tarefa 4:

Então, a tarefa da Startup é levantar informações sobre o desenvolvimento de aplicações móveis.

1. Faça um levantamento das plataformas e linguagens de programação mais utilizadas para desenvolvimento Android disponíveis no mercado.

Relacione quais as vantagens e desvantagens de cada uma.

A primeira coisa que desenvolvedores precisam levar em consideração é fazer o download do Android Studio. Este é um software IDE (Integrated Development Environment), oferecido como um pacote com o Android SDK. Ele nada mais é do que um conjunto de ferramentas usadas para facilitar o desenvolvimento de aplicações. No site do Android Studio, você também encontrará tutoriais e documentos oficiais do Google, além de bibliotecas e códigos gratuitos para aprimorar seus aplicativos.

Na lista abaixo, confira o das linguagens de programação mais indicadas:

**1. Java**

Java é uma linguagem orientada a objetos.

Talvez não seja uma novidade pra você, mas, Java é uma linguagem de programação orientada a objetos ( há a representação do mundo real), o que nos ajuda a melhorar a flexibilidade e reusabilidade do código.

**Vantagens:**

Ao utilizar conceitos de orientação a objetos, nós podemos utilizar classes e seus métodos em outros programas. Isso, nos permite também melhorar a segurança das nossas aplicações, protegendo métodos e atributos com o encapsulamento de classes e controle de acesso do Java.

**Portabilidade**

É possível rodar programas compilados em java na Web, no celular, no seu relógio e até na geladeira, tudo graças a sua máquina virtual (JVM).

**Documentação Extensa**

Por ser uma das linguagens mais utilizadas no mundo, sua documentação possui inúmeras contribuições, e na comunidade JAVA, você nunca vai encontrar um pro-blema pelo qual outra pessoa não tenha passado.

**Java suporta Multithreading**

Java é uma linguagem multithreading, ou seja, mais de uma thread pode ser execu-tada ao mesmo tempo em uma aplicação, o que nos ajuda a obter o máximo de uti-lização da CPU. Thread é a menor unidade de um processo. Vários Threads compar-tilham uma área da memória comum e aumentam a ediciência e o desempenho da aplicação. Esses threads são executados independentemente uns dos outros.

**Java possui o famoso Garbage Collector**

O gerenciamento automático de memória em Java que é gerenciado pela JVM. A função do Garbage Collectior é remover automaticamente todos os objetos aloca-dos na memória e que não estão sendo mais utilizados pela aplicação, o que permite uma alocação de memória mais eficiente.

**Desvantagens:**

Antes de começar a aprender ou trabalhar com qualquer linguagem de programação, é importante saber as desvantagens e seus pontos fracos, para posteriormente não ter dores de cabeça. Vou listar aqui algumas desvantagens e pontos que o java deixa a desejar.

**Java não possui herança Múltipla**

Ao contrário do C++, em JAVA NÃO existe Herança múltipla, situação em que uma subclasse é derivada de duas ou mais superclasses, porém, isso pode ser contornado implementando Interfaces.

**Performance Reduzida**

Java é um excelente consumidor de memória, o que significa que sua performance é mais lenta que outras linguagens nativas como o C ou C++, nessas linguagens cada código tem que ser interpretado a nível de máquina, já no Java existe uma pré-compilação nas aplicações para depois elas serem interpretadas pela JVM.

**Os códigos em Java são prolixos**

Talvez isso seja o que há de mais chato no java depois da sua lentidão, os códigos em Java são prolixos, o que significa que possui muitas palavras e muitas sentenças lon-gas e complexas que são difícies de ler e entender. Isso reduz a legibilidade do código. A linguagem Java apesar de ser mais gerenciável devido a programação orientada a objetos, ao mesmo tempo, compromete tudo isso com códigos excessivamente complexos.

**2. Kotlin**

**Vantagens:**

**JetBrains nos bastidores**

A jetbrains é uma excelente empresa e seus produtos são de qualidade ine-gável. Eles estão diariamente propondo algo novo, empenhados em melhorar a linguagem e aumentar a popularidade do Kotlin e o uso do AndroidStudo, motivando a ainda pequena comunidade de usuários do Kotlin.

**Interoperabilidade**

A interoperabilidade é o objetivo principal de Kotlin.

Desde o início, a intenção do projeto foi usar o conhecimento e a experiência existentes para disponibilizar todas as bibliotecas aos programadores da Kotlin. Os desenvolvedores podem simplesmente escrever módulos no Kotlin que funcionam perfeitamente dentro do código Java existente. Ao emitir o Bytecode, um compilador Kotlin permite que os dois idiomas trabalhem em uníssono no mesmo projeto.

**Sintaxe Limpa e Expressiva**

Kotlin introduz várias melhorias que causam um impacto significativo na legi-bilidade do código, aumentando a produtividade e trazendo mais facilidade para manutenção do código gerado.

Para mencionar alguns:

•Ponto e vírgula são opcionais.

•Chaves “{}” são opcionais nos casos em que não são úteis.

•Getter/Setters são opcionais.

•Tudo é objeto – primitivos são usados nos bastidores automaticamen-te, se necessário.

•Mais consistente e menos verboso – Em algumas situações é possível ter o mesmo comportamento que no Java usando muito menos código.

**Kotlin is null safe por padrão**

As variáveis não são nulas por padrão, permitindo a verificação nula durante o tempo de compilação, o que reduz bastante a ocorrência do temido NullPo-interException durante o tempo de execução.

Se você quiser uma referência anulável, será necessário adicionar um ponto de interrogação ao seu tipo de dados da seguinte forma: var secret: String? nomedocampo = null.

**Coroutines**

É uma ótima maneira de escrever código assíncrono de forma mais natural.

Eles são essencialmente threads mais claras e concisas.

Em Java, você normalmente criaria threads de segundo plano para gerenciar operações intensivas. Isso muitas vezes levou ao aumento da complexidade e da frequência das dores de cabeça. Coroutines também permitem muitas outras possibilidades, como simultaneidade, atores e muito mais.

**Quais as desvantagens de usar o Kotlin?**

**Experimentado e Testado**

Existe uma comunidade extensa e engajada de desenvolvedores construindo todos os tipos de sistemas e aplicativos com Java há mais de 20 anos. O primeiro lançamento oficialmente estável do Kotlin foi no início de 2016, já o Java tem estabilidade e resistiu ao teste do tempo. Não podemos dizer o mesmo para o Kotlin.

**Velocidade de compilação**

Desenvolvedores informaram sobre a flutuação na velocidade de compilação de códigos Kotlin. Em alguns casos, é muito rápido e, em outros casos, é signi-ficativamente mais lento, mas não chegando a ser tão ruim quanto a lingugem Scala.

**3. C / C ++**

**Vantagens:**

**Produção de código o quanto mais eficiente possível.**

Possibilidade em programação de alto e baixo nível;

Alta flexibilidade, portabilidade e consistência;

Adequado para grandes projetos;

Não está sob o domínio de uma empresa (em contraste do Java — Sun ou Visual Basic — Microsoft);

Padronização pela ISO;

Grandes possibilidades para a metaprogramação e programação genérica;

Compatibilidade com C, resultando em vasta base de códigos.

**Desvantagens:**

Compatibilidade com o C, herdou os problemas de entendimento de sintaxe do mesmo;

Os compiladores atuais nem sempre produzem o código mais otimizado, tanto em velocidade quando tamanho do código;

Grande período para o aprendizado;

A biblioteca padrão não cobre áreas importantes da programação, como threads, conexões TCP/IP, interface gráfica e manipulação de sistemas de arquivos, o que implica a necessidade de criação de bibliotecas próprias para tal, que pecam em portabilidade;

Devido à grande flexibilidade no desenvolvimento, é recomendado o uso de padrões de progra-mação mais amplamente que em outras linguagens.

Não possui coleta automática de lixo (se durante o programa uma variável declarada não for mais usada ao longo do programa a linguagem não tem capacidade para eliminá-la continuando esta a ocupar espaço indevidamente em memória);

Não faz uso de classes ou objetos (como no JAVA e C++);

Networking (é uma linguagem que inicialmente não foi pensada para redes).

2. Elabore uma pesquisa sobre as 5 principais tendências de desenvolvimento de

aplicativos Android no mercado atual, como a utilização de inteligência artificial e

machine learning, chatbots, entre outras.

**1. Inteligência Artificial e Aprendizado de Máquina**

Inteligência Artificial é um fenômeno bem conhecido hoje.

Até o final de 2024, o mercado de inteligência artificial estará em US $ 191 bilhões. Conseqüentemente, ele está sendo usado no setor de TI com mais frequência.

Seu uso em aplicativos para dispositivos móveis Android está em voga para melhorar a experiência de aplicativos dos usuários. Essa tecnologia traz várias vantagens para as empresas, pois a combinação de aprendizado de máquina e IA torna os aplicativos capazes de aprender o comportamento dos usuários.

Além disso, ele pode analisar as atividades e comportamentos do usuário rastreando o uso diário e os padrões de pesquisa e, em seguida, sugerir informações importantes sobre as preferências do usuá-rio.

**Tecnologia Beacon**

Beacons atualmente estão sendo muito utilizados por empresas, pois permitem que essas empresas segmentem clientes em potencial de uma maneira única. Essa tecnologia ainda está em sua fase inicial e cresce continuamente.

Basicamente, os beacons estão transmitindo dispositivos que podem se conectar aos dispositivos móveis disponíveis em seu alcance. Em seguida, eles podem ajudar as empresas a enviar notificações sobre ofertas especiais, hotéis próximos e muito mais para esses dispositivos.

Por esse motivo, a integração dessa tecnologia nos aplicativos Android está se tornando muito popular, e empresas e desenvolvedores estão ansiosos para testar seus resultados com o comportamento de compra dos usuários. Para empresas como restaurantes e lojas, essa integração pode ser extremamente proveitosa.

O uso de iBeacons e outros dispositivos semelhantes para marketing deve crescer exponencialmente.

**Chatbots**

Os chabots também estão sendo amplamente utilizados, pois os clientes esperam respostas mais rá-pidas. Os robôs oferecem respostas instantâneas à consulta de um usuário. Além disso, a disponibili-dade de assistência virtual 24/7 é muito conveniente para os usuários. Esses recursos tornam os chatbots a principal tendência de desenvolvimento do Android em 2020.

**Aplicativos Android do provedor de serviços sob demanda**

Como os aplicativos móveis são ferramentas úteis para a aquisição de diferentes tipos de serviços, a tendência dos aplicativos Android sob demanda está aumentando bastante. Embora o principal obje-tivo desses aplicativos seja aprimorar o atendimento ao cliente, aplicativos sob demanda como Uber, DoorDashe Postmates ajude os clientes a encontrar diferentes serviços com facilidade.

A maioria desses aplicativos permite que os usuários obtenham o serviço específico no local, como em casa, no escritório ou em algum lugar na rua. Esses aplicativos tornam as tarefas diárias mais confortá-veis e rápidas.

Esses tipos de aplicativos seguem diferentes conceitos de serviço sob demanda, como entrega de su-permercado, entrega de alimentos, aluguel de carros, serviços de limpeza e muito mais.

**7 - CONCLUSÃO**

**8 - REFERÊNCIAS BIBLIOGRÁFICAS**

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

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

COMO FUNCIONA A INTERNET -<https://nic.br/videos/ver/como-funciona-a-internet-parte-1-o-protocolo-ip/>

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

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

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>

O que é web - https://www.significados.com.br/web/

Introdução Linguagem de Programação - https://pt.wikiversity.org/wiki/Introdu%C3%A7%C3%A3o\_%C3%A0s\_Linguagens\_de\_Programa%C3%A7%C3%A3o