**1) Elabore uma resenha, justificando a importância da estimativa de esforço em projetos de software, apontando os aspectos facilitadores nesse processo.**

A utilização de uma estimativa especifica para o Teste dentro do ciclo de desenvolvimento de software tem como objetivo evitar que o mesmo seja colocado em segundo plano, em função dos possíveis atrasos nas demais fases do processo.

Estimar é uma atividade que deve ser realizada frequentemente, para que através de sucessivas melhorias seja possível obter o nível de exatidão esperado. A estimativa do esforço do Teste de Software é muito importante para a avaliação dos custos do projeto. Algumas variáveis como maturidade do processo de teste e testabilidade do software também influenciam na estimativa do projeto.

Quando o desenvolvimento não segue o plano e, por alguma razão, começam a ocorrer atrasos dentro dos prazos estabelecidos, isso pode afetar toda a empresa, aumentando os custos operacionais e até mesmo causando prejuízo.

Estimar com certeza os prazos para a finalização de tarefas dentro do projeto de software é impossível. É normal que algumas entregas não sejam cumpridas, no entanto, é preciso manter o máximo possível todas as atividades dentro do estimado, a fim de evitar:

* Desconhecimento de todos os requisitos envolvidos no processo;
* Falta de conhecimento por parte dos desenvolvedores em como metrificar processos de software;
* Estimativas incorretas que acabam por prejudicar o [cronograma](https://gaea.com.br/9-principais-erros-no-cronograma-de-desenvolvimento-de-aplicacoes/) geral das atividades;

**Aspectos facilitadores:**

* Medir o tamanho da funcionalidade solicitada pelo usuário antes de realizar a codificação da mesma para mensurar o tamanho e custo;
* Medir os projetos de inovação e manutenção de software, sem considerar a tecnologia utilizada, para gerenciar e acompanhar a evolução do projeto;
* Medir o tamanho da funcionalidade entregue ao usuário, após o projeto finalizado, para comparar o que foi estimado em termos de tamanho e custo com o realizado.

**2) Pesquise e liste, pelo menos, cinco técnicas de estimativas. Descreva, de forma simplificada, sobre cada uma delas.**

**Técnica Delphi**

A técnica Delphi tem como principal diferencial impedir que a opinião de um dos membros da equipe acabe por influenciar as respostas de todos os outros profissionais.

Isso acontece, geralmente, quando alguém da equipe é um profissional renomado. Ao dar sua opinião sobre o tempo necessário para uma atividade, os outros colaboradores podem ficar intimidados para discordar.

Nela, é preciso criar um questionário simples com cada atividade que faça parte do projeto para que o profissional descreva o prazo necessário para a sua conclusão. Depois disso, todos colocam suas respostas em um recipiente e um facilitador que não está participando da estimativa analisa a resposta.

Assim, nenhum dos envolvidos pode precisar de quem é cada resposta, não sendo influenciado por ninguém. Caso não haja um consenso sobre o prazo necessário, pode-se realizar novas rodadas.

**Planning Poker**

A sua utilização é muito simples e pode ser realizada para qualquer projeto, independentemente do tamanho. Essa técnica tem necessidade de um baralho, sendo possível encontrar esse item para venda em algumas lojas e até mesmo ganhá-lo como brinde em algumas convenções de desenvolvimento.

Os números das cartas aqui são diferentes de um baralho normal, baseados na sequência Fibonacci, ou seja: 1, 2, 3, 5, 8, 13, 21, 34 e 55. Acima disso, a tarefa é grande demais é precisa ser dividida em itens menores para poder ser estimada com mais precisão.

O primeiro passo é escolher qual processo será estimado. São necessários no mínimo dois desenvolvedores, que vão ler os requisitos da atividade, então, cada um escolherá uma carta que apresente quantas horas ele acha que levará para concluir aquela tarefa.

Ambos mostram sua carta ao mesmo tempo. Caso tenham estimado de forma diferente, os dois devem explicar por que escolheram aquele tempo e, depois, refazer a passo acima, escolhendo novamente uma carta, com base em seu conhecimento e de acordo com as explicações do colega.

Em uma situação em que a estimativa continue diferente, é escolhido o maior prazo entre as cartas apresentadas pelos desenvolvedores que estão participando do jogo.

Todas as estimativas são anotadas junto a cada uma das atividades e, ao final, são somadas para apresentar qual o prazo total estimado para aquele projeto de desenvolvimento, sendo tal estimativa repassada ao cliente para a aprovação.

**Pomodoro**

Utiliza alguns dos elementos apresentados no Planning Poker. Aqui, a sequência Fibonacci também é aplicada, porém de uma maneira diferente.

Essa técnica pode ser usada por um único desenvolvedor ou em equipe. O primeiro passo é escolher qual atividade será estimada, e todos aqueles que estiverem participando da estimativa devem ler seus requisitos.

Ao final da leitura, os envolvidos iniciarão as suas apostas de estimativa baseada em um número de pomodoros. O pomodoro é um período composto por 25 minutos de trabalho e 5 minutos de intervalo, ou seja, meia hora de atividade.

As estimativas devem seguir a sequência Fibonacci, a tarefa pode ser estimada em 1, 2, 3, 5, 8, 13, 21, 34 ou 55. Acima disso, ela é muito grande e deve ser dividida em duas ou mais para que a estimativa fique o mais próximo da realidade.

**Ponto de função**

A análise de ponto de função não é uma técnica para estimar o tempo de desenvolvimento especificamente, mas sim para ter uma ideia geral do tamanho do software, informação essencial para a estimativa.

Os pontos de função são baseados naquilo que se espera que o sistema seja capaz de fazer, e não como ele deverá ser desenvolvido, isso com base nos requisitos lógicos levantados junto ao cliente pela [equipe](https://gaea.com.br/como-reduzir-rotatividade-da-equipe-de-ti-na-sua-empresa/).

Essa é uma técnica bem complexa e capaz de estimar com precisão o tamanho de um sistema. Cada funcionalidade, na visão do usuário, recebe uma determinada pontuação de acordo com a sua complexidade.

Ao final, tem-se uma ideia geral do tamanho do software que deverá ser [construído](https://gaea.com.br/construindo-aplicacoes-escalaveis-com-cerberon-entenda-mais/) e, com isso, pode-se aplicar as técnicas de estimativa para se construir um prazo geral para conclusão de todas as atividades do projeto.

**Pontos por caso de uso**

Essa é uma metodologia clássica de estimativa em software e já existe desde 1993, quando Gustav Karner a idealizou com base nos casos de uso criados com auxílio de UML, Unified Modeling Language.

Essa linguagem de modelagem é utilizada em projetos de software que investem muito tempo na confecção de documentação. É utilizada para demonstrar de maneira gráfica todas as interações dentro do sistema.

Ou seja, cada caso de uso é uma tarefa, e o desenvolvedor deve atribuir uma pontuação de acordo com a sua complexidade. Criar um cadastro é uma atividade simples, já desenvolver uma integração com outro sistema pode ser mais complexo.

Cada ponto deve ser equivalente a uma unidade de tempo, podendo ser descrito em horas de trabalho. Sendo assim, quanto mais complexa uma tarefa, maior será a quantidade de pontos atribuídos e o prazo para a sua conclusão.

**3) Explique cada etapa do processo a seguir.**



Coletar os requerimentos iniciais: Documentar o que o cliente quer, e coletar os requisitos

Estimar o tamanho do produto: Estimar final do produto dado módulos, programas, interfaces, linhas de código, sub-rotinas ou classes.

Estimar o esforço: Estimar o esforço do projeto deve estimar quantos homens/horas (medida do esforço para o desenvolvimento) são necessários para se construir o produto.

Produzir o cronograma: Deve indicar quando cada atividade começa e termina.

Estimar o custo: Equipe financeira, estimará o custo para produção do software, e quanto pedirá ao cliente.

Aprovar as estimativas: Analisar as estimativas conforme técnicas e experiências, e aprovar ou não.

Desenvolver o produto: Programar o produto.

**4) Apresente exemplo de um produto (ou serviço) que você pretende trabalhar ao longo da disciplina.**

Um sistema de confecção, onde se tem o administrador do sistema, o(s) confeccionista(s), e o(s) cliente(s).