

Nome: Arthur Guilherme Esteves Martins
Nome: Tiago Pereira Higa Martins

RA: 21109970
RA: 20201015

1. (2.0) Responda às seguintes questões quanto a modelagem de sistemas:

a. O que é Modelagem de Sistemas? Por que é importante?

Todo software ao ser construído passa por um estágio inicial. Nesse estágio é se determina como será conduzido do zero até a sua implementação. Dito isso, a modelagem faz parte desse processo inicial onde serão realizadas atividades que irão explicar e ilustrar como irá funcionar o software, ou seja, os passos para atender a solução do problema.

É a parte onde se entende a dificuldade, a regra do negócio do cliente para conseguir atender com a solução, assim se verificam através da abstração, requisitos, técnicas e metodologias a serem usadas para atingir o fim proposto.

A sua importância baseia-se compreender melhor o sistema que será desenvolvido, sua estrutura, a visualização. Determinamos a quantidade e as escolhas dos papéis de cada um nessa construção, a escolha dos modelos que atendam especificamente para a necessidade proposta.

As horas de desenvolvimento são baixas já que se tem a documentação do que se precisa fazer, o que precisa ser construído. E não é algo estático, muito pelo contrário, a modelagem é dinâmica e pode ser revista a qualquer momento caso a solução não atenda a regra de negócio. E financeiramente se constrói uma documentação que indica todas as decisões tomadas em conjunto e o que será efetivamente construído e o valor pago. Em caso de mudanças no decorrer, vai mudar as horas empenhadas e consequentemente o valor pago pelo cliente, fora a segurança que isso dá a todos os envolvidos.

b. O que é UML?

Significa em nossa língua a Linguagem de Modelagem Unificada, que é a modelagem de sistemas mais usada. Como é uma modelagem de sistema serve para criar sistemas complexos de software. Por ser uma linguagem padronizada, motivo pelo qual seu uso é global e também por ter sido aprovada como tal. Seu objetivo é realizar a modelagem de Sistemas de forma ordenada seguindo a sua padronização para tal, assim auxiliando desde a abstração, ilustração, comportamento através de sua diagramação.

c. Cite pelo menos um exemplo de cada um dos seguintes tipos de diagrama: Estrutura, Comportamento e Interação da UML.

O de estrutura pode-se citar o primeiro que usamos que é o de classes, onde realizamos o esboço inicial da estrutura do sistema enquanto suas classes. No qual conseguimos entender as regras do negócio, as interfaces e o diagrama que geralmente usamos para a construção dos demais já que ali tem a visão geral do escopo do sistema com suas regras de negócio.

Já o de comportamento será usado o caso de uso, onde as etapas serão divididas e a cada pedaço será entendido com o sistema irá se comportar, entender os

requisitos e ter uma grande parte importante que é o documento por escrito do caso de uso proposto com o passo a passo do sistema e suas interações.

Por fim, o de interação temos o de sequencia que é importante pois ele representa todas as interações que ocorrem entre os objetos de um determinado cenário. Ele é construído com base no caso de uso, ou seja, pela sequencia podemos ver por exemplo todos as interações que um usuário vai ter para efetuar um cadastro no sistema por exemplo, desde clicar no login, colocar dados, sistema verificar dados e todo o restante desse processo.

2. (2.0) Defina os seguintes conceitos:

a. Sobrecarga: sobrecarga é uma característica que permite que uma determinada classe tenha métodos com o mesmo nome, mas assinaturas diferentes. Através da sobrecarga, o método correto é selecionado com base na lista de parâmetros que ele recebe.

b. Polimorfismo: é o princípio pelo qual classes derivadas de uma mesma classe padrão podem invocar métodos que, apesar de possuírem a mesma assinatura, implementam comportamentos distintos para cada uma das classes.

c. Herança: define o comportamento no qual características comuns a duas ou mais classes podem ser definidas em uma classe base e utilizados pela subclasse. Desta forma, a subclasse apresentará as características (atributos e métodos) da classe base, bem como qualquer implementação específica.

d. Classe abstrata: servem como um modelo de como as classes que a herdarem devem se comportar, definindo atributos e métodos. A principal característica da classe abstrata é que ela não pode ser instanciada. Além disso, pode definir métodos obrigatórios para implementação nas subclasses através do “abstract”.