Faculdades Anhanguera

Analise e Desenvolvimento de Sistemas

Projeto Integrado II

Felipe Passos

PROJETO INTEGRADO III

Tutor: Jobson Yonaha Gimenez

Uberlândia, 10 de maio de 2022

**ATIVIDADES**

A UML (Linguagem de Modelagem Unificada) é uma linguagem-padrão para a elaboração da estrutura de projetos de software. Ela pode ser empregada para a visualização, a especificação, a construção e a documentação de artefatos que façam uso de sistemas complexos de software. A UML possui diagramas estruturais e comportamentais. Sobre esse assunto, responda às questões a seguir:

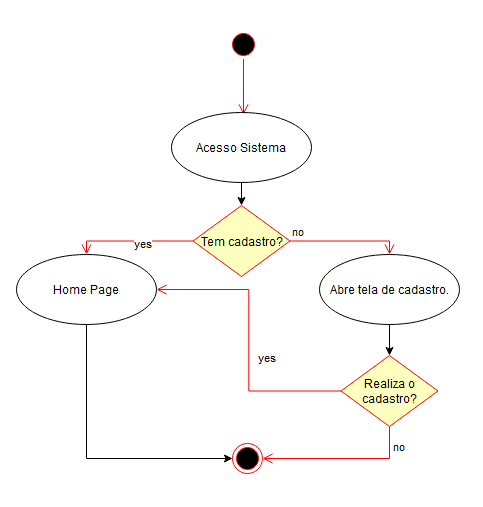
• Descreva a diferença entre os diagramas estruturais e comportamentais;

• Escolha um diagrama comportamental e um diagrama estrutural e descreva as suas principais características e em qual momento do projeto eles se encaixam.

1. Um diagrama estrutural, como o próprio nome implica, trata das características estruturais de um sistema, banco de dados ou aplicação. Ele é utilizado para descrever características como tabelas, tipos de dados, relacionamentos, objetos, classes e todas os demais elementos do sistema a ser criado. Ele é fundamental, inclusive, para se definir um projeto pois a partir dele, aliado aos requisitos e à atuação do PO e GP, um time Scrum pode elencar corretamente as entregas e definir os tempos de desenvolvimento corretamente. Já o diagrama comportamental trata, principalmente, das características relacionadas à usabilidade e emprego da aplicação, ele é muito importante para serem criados os protótipos e case-tests pelos times de UI e Q&A, pois sabendo, com clareza, quais os objetivos e empregos que serão dados ao produto poderemos definir testes e interfaces funcionais e garantir a qualidade das entregas. Em vista disso, ambos os diagramas são essenciais para um desenvolvimento sólido e de qualidade, sendo tanto ágil quanto eficiente no processo de desenvolvimento e evitando a criação de soluções pouco aderentes ou condizentes com o produto esperado.

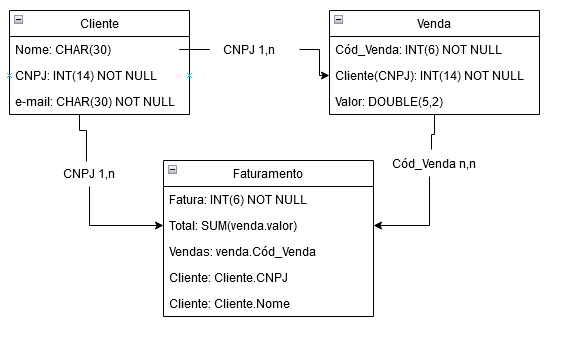
2.

a) Diagrama comportamental:



Aqui temos um diagrama comportamental, onde mostra o comportamento esperado de um sistema no processo de login. Aqui podemos observar a partir de um modelo BPM qual o fluxo do processo de login em um sistema onde, caso o usuário possua cadastro será direcionado à home page, do contrário ele direciona o usuário para a tela de cadastro e, caso o usuário realize o cadastro, ele envia o usuário para a home page, caso não realize o cadastro ele encerra o fluxo.

b) Diagrama estrutural:



Acima temos um diagrama de classes que demonstra o relacionamento entre tabelas de um sistema de vendas onde, a partir dele, é possível vermos que teremos que ter uma tabela com o cadastro de clientes, uma com os dados das vendas e, a partir delas, criaremos as faturas utilizando o CNPJ e a soma dos valores de vendas.

Para o design de interfaces, devemos nos preocupar com diversos aspectos. Imagine que você desenvolverá a página inicial do site de um programa para o Ministério da Educação. Ele poderá ser acessado tanto na web quanto nos dispositivos móveis. Esse programa fará o mapeamento de faculdades com cursos de tecnologia que existem no Brasil.

a) Para o desenho da interface podemos utilizar um esboço em papel e caneta e, a partir dele, definimos através de um programa de prototipagem de interface as telas que teremos que construir, entre eles temos os mais conhecidos FIGMA, Framer e AdobeXD.

b) Observar a usabilidade e eficácia da página, sempre nos concentrando em manter uma interface limpa e intuitiva além de manter uma boa fluência entre as transições observando uma boa performance de navegação e a fácil identificação do usuário em cada transição.

c) Temos que nos atentar que a página deverá ser responsiva pois poderá ser usada tanto na web quanto mobile, ser intuitiva e requer poucos cliques para executar as ações. Particularmente eu dividiria o projeto em uma home-page com a tela de login, a partir do login abriria um mapa do Brasil com os estados e um contador com a quantidade de faculdades em cada estado, ao clicar no estado um pop-up com as cidades onde há faculdades seria exibido e, a partir dele, ao clicar na cidade listaria as faculdades e seus detalhes.

d) Observando a praticidade, para criar os protótipos eu escolheria uma ferramenta que me permitisse tanto criar o design quanto o protótipo na mesma ferramenta, dessa maneira seria possível, desse modo durante o processo de design já seria possível testar alguns elementos da aplicação e criar o design e o protótipo em um mesmo momento.

Leia o fragmento, interprete o código Python em questão e assinale alternativa correta.

a) Descreva como o método recebe o valor:

O valor será enviado a partir do argumento ‘valor’ definido na chamada da função remove(), neste caso o valor, como não está tipado, poderá levar qualquer argumento que seja apontado, ou seja, uma variável, string, lista ou dicionário.

b) Não sendo o método, como será a busca? Lembre-se do head.

A busca será realizada comparando o conteúdo enviado como valor com a primeira linha de uma tabela a partir do método head. herdado da lib pandas e navegando para o próximo caso o valor seja o procurado, neste caso caindo em um loop que realizará as alterações descritas no item C)

c) Quais alterações ocorrerão na lista?

Limpará os itens da lista que possuam o valor apontado em ‘valor’, substituindo seus conteúdos por None (pode ser entendido também por null ou vazio) e navegando os itens que não coincidam com o valor indicado. Desta maneira, ao final do loop, teremos uma lista com apenas um valor que será o cabeçalho e os demais serão removidos.

Qual a diferença entre Bancos de dados relacionais e não relacionais?

Cite as principais características e diferenças entre cada um deles.

A principal diferença entre os bancos relacionais (SQL) e não relacionais (noSQL) está em sua estrutura básica onde, enquanto o banco relacional é composto e orientado por tabelas cujos dados são vetorizados na horizontal e vertical, nos bancos não relacionais a orientação se dá por documentos, onde podemos entender que cada conjunto de dados é composto por um dicionário ou array.

Outra característica importante se dá em suas aplicações e performances, bancos relacionais são, geralmente, os mais recomendados quando utilizados para dados bem estruturados e padronizados como financeiros, contábeis ou estoques, por exemplo. Enquanto isso, os bancos noSQL são fortemente recomendados para dados não estruturados e pouco padronizados, como documentos, artigos ou textos, por exemplo. Além da questão estrutural, ambos tem performances distintas pois, a partir de suas características tendem à performar de maneira diferente em determinadas aplicações, sendo muito importante, nesse caso, mensurar e entender muito bem os dados que serão tratados para optar corretamente entre uma ou outra opção e não ter problemas de performance na aplicação.