

LABORATÓRIO DE SOFTWARE E PROJETOS

MODELAGEM CONCEITUAL DE SOFTWARE

Autor: Me. Rodrigo Ramos Nogueira

Revisor: Felipe Oviedo Frosi

INICIAR

introdução

Introdução

Prezado(a) aluno(a), o processo de desenvolvimento de um sistema é complexo e envolve diversas metodologias e ferramentas. Esta unidade proporcionará, a você, um arcabouço de ferramentas para garantir a qualidade do sistema desenvolvido. O primeiro elemento a ser estudado são os protótipos, uma maneira de ter uma visão geral de seu produto antes mesmo de desenvolvê-lo. Os mapas mentais são elementos que poderão ser utilizados em diversas etapas do desenvolvimento de *software*, com o objetivo de organizar. Por fim, utilizando o UML, você aprenderá como projetar o seu código sem digitar nenhuma linha de código.



Design de Protótipo



O processo de desenvolvimento de sistema envolve um conjunto de técnicas de modelagem e projeto de sistemas. Desde a sua ideia até a concepção de um *software* em si, uma estratégia que pode contribuir com o projeto de software é o desenvolvimento de protótipos.

Mas, afinal, o que é um protótipo? Em sua definição do dicionário, um protótipo é primeiro tipo criado, algo original. Isso quer dizer que um protótipo é a primeira versão de algo, a ideia de seu criador sendo colocada em prática pela primeira vez, seja um projeto da engenharia mecânica, ou um carro, ou um avião e, é claro, o projeto de um *software* .

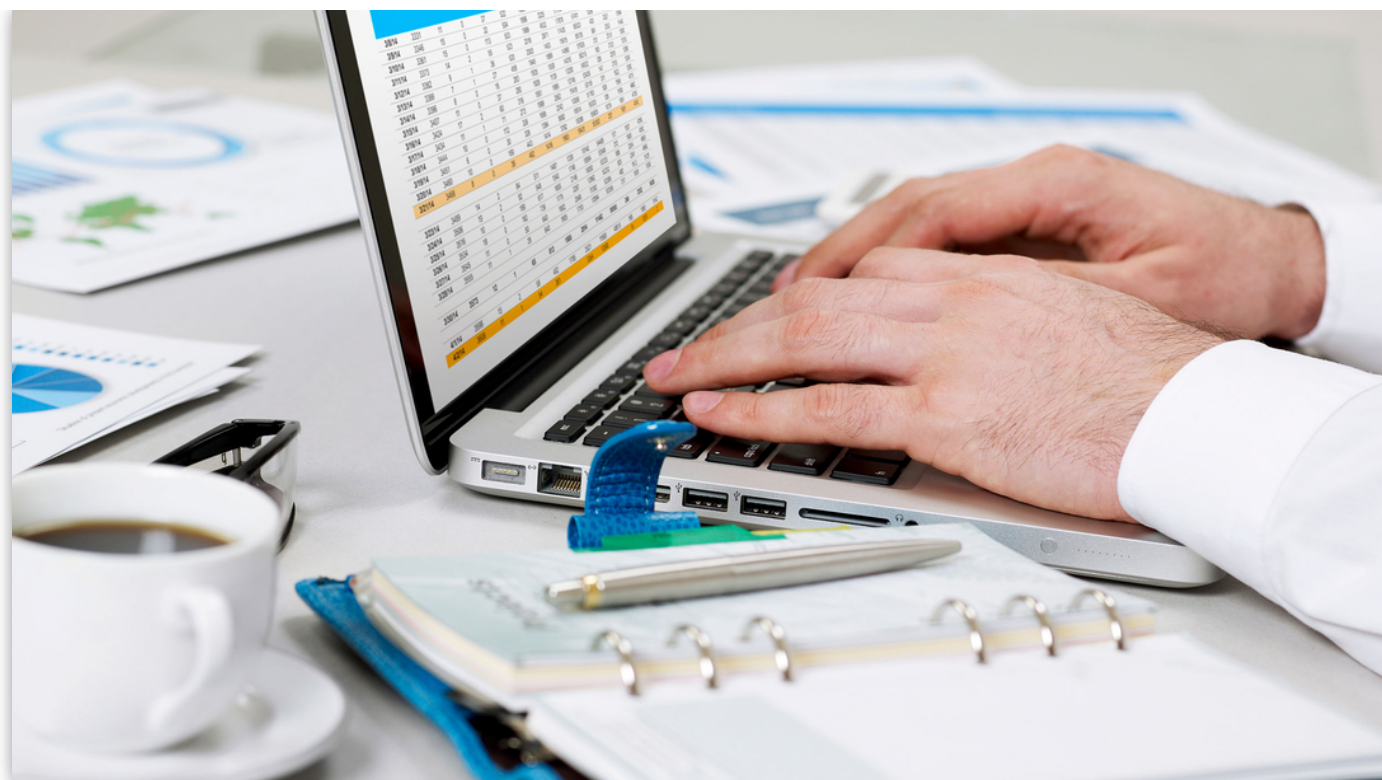


Figura 3.1 - Protótipo
Fonte: mearicon / 123RF.

#PraCegoVer: Ao centro de uma mesa há um notebook aberto e na tela deste há uma planilha e no teclado duas mãos digitando. Em volta do notebook há uma xícara branca de café, folhas com desenhos de gráficos e uma agenda de anotações com uma caneta em cima desta agenda.

No contexto do desenvolvimento de *software*, a criação de protótipos tem, como objetivo, representar os requisitos levantados na forma de telas do sistema. Segundo Spinola (2019), os protótipos são uma maneira de representar a versão inicial de um sistema para experimentação, devendo suportar os requisitos levantados.

O objetivo do protótipo é permitir que se possa visualizar o produto de acordo com os requisitos levantados.

Protótipos podem fornecer grande quantidade de conhecimentos sobre a interação do usuário em vários níveis. Eles nos permitem testar a viabilidade e usabilidade do projeto antes mesmo de escrever o código. Descobertas inesperadas e inovações ocorrem neste processo (PINHEIRO, 2016, on-line).

A ideia da criação de protótipos vai além da validação dos requisitos e de atender a real necessidade do cliente, sem a necessidade de alterações no sistema. A possibilidade de visualizar o *software* permite o auxílio em outras etapas, por exemplo, na modelagem de banco de dados.

O desenvolvimento de um protótipo, além de ser aliado à engenharia de software na garantia dos requisitos, é um forte recurso de gestão de projetos. Isso acontece porque a criação de um protótipo pode evitar esforços na etapa de desenvolvimento de *software* . Para compreender melhor, pense no exemplo clássico de quando o cliente pede para “adicionar só um campo”.

Ao apresentar diretamente o sistema desenvolvido (pronto) para o cliente, pode acontecer de que essa simples alteração envolva o tempo de alterações no banco de dados (remoção e adição de campos e tabelas, codificação do *software* , bem como uma revisão de documentação do sistema). Isso pode ser evitado, com a apresentação do protótipo, em que, na etapa de projeto de sistema, o cliente poderá sugerir alterações, sem impacto em um *software* .

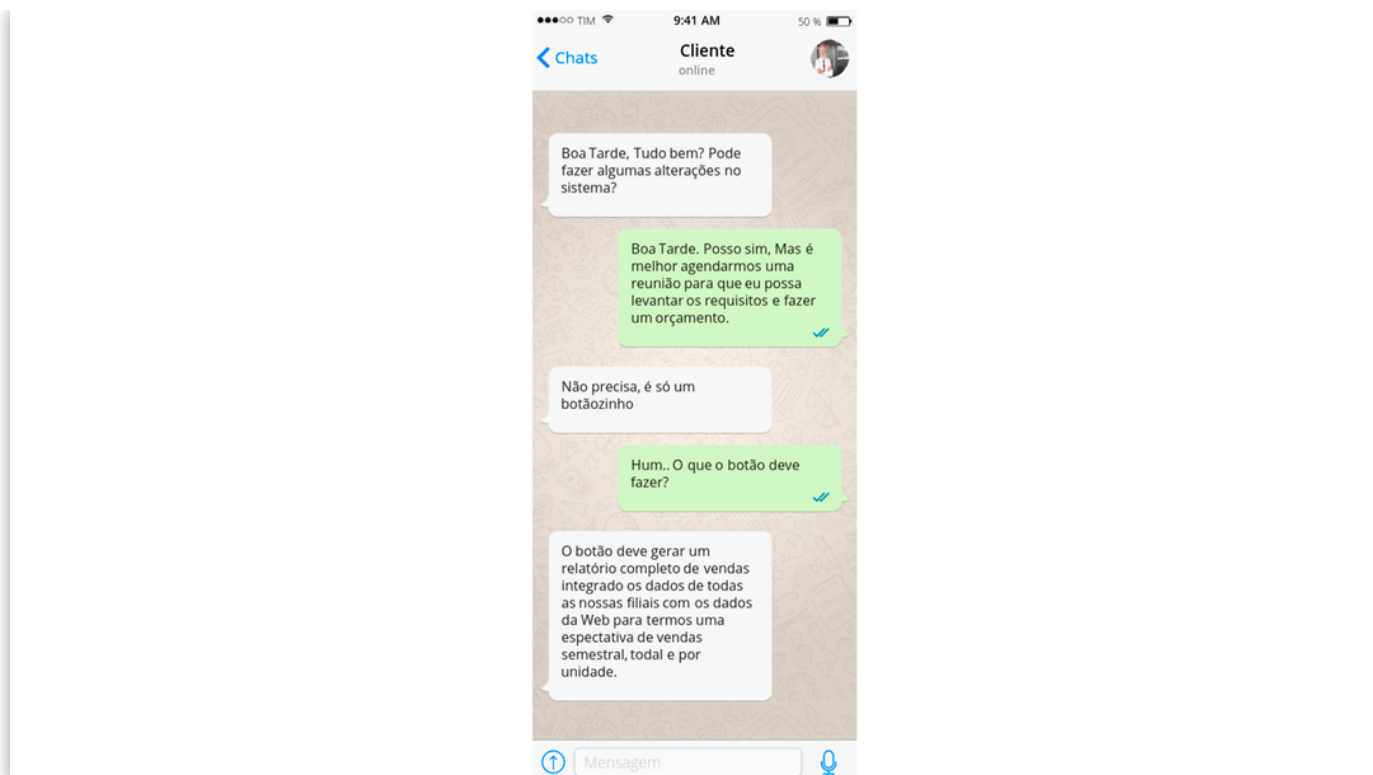


Figura 3.2 - Charge - "Só um Botãozinho"

Fonte: Elaborada pelo autor.

#PraCegoVer: Print de uma janela de WhatsApp. Conversa entre cliente e produtor.

Cliente: Boa tarde, tudo bem? Pode fazer algumas alterações no sistema?

Produtor: Boa tarde. Posso sim. mas é melhor agendarmos uma reunião para que eu possa levantar os requisitos e fazer um orçamento.

Cliente: Não precisa, é só um botãozinho.

Produtor: Hum...o que o botão deve fazer?

Cliente: O botão deve gerar um relatório completo de vendas integrado os dados de todas as nossas filiais com os dados da web para termos uma expectativa de vendas semestral total e por unidade.

Tendo em vista compreender como um protótipo pode ser criado, as próximas seções explicarão os três níveis de protótipo de *software* : baixa

fidelidade, média fidelidade e alta fidelidade.

Saiba mais

Quando falamos em prototipação de telas, é comum vir à tona o termo *User Experience* - UX, que é a experiência do usuário. Como o nosso conteúdo tem ênfase na gestão de projetos e engenharia de software, não nos aprofundaremos nesse tema, mas é importante termos conhecimento sobre ele. O termo UX significa *User Experience* (experiência do usuário) e foi cunhado por Don Norman, em meados de 1990, em seu livro *O Design do Dia a Dia*. O autor conceituou seis princípios de design da oscilação de produtos que satisfazem o consumidor e tornam este menos frustrado, com relação à experiência com o produto. Seus seis princípios do design são pontuados em: visibilidade, *feedback*, restrições, mapeamento, consistência e *affordance*, que se baseiam em ter funções mais visíveis, de fácil descoberta, possuir retorno de informação, a partir de uma ação executada, evitar erros, restringir as possibilidades de escolha e direcioná-las a serem mais eficazes para o processo.

Ademais, deve-se controlar ações de execução e a relação entre estas, traçar um movimento de ação e mapear o resultado obtido, juntamente com esses objetivos, além de manter um padrão de interface e

operações similares expostas com elementos similares. Esses elementos possibilitam um entendimento claro de como utilizar e interagir com o produto. No contexto da experiência do usuário, os protótipos têm um papel importante, pois permitem descobrir e avaliar a experiência do usuário antes mesmo do sistema desenvolvido. Desse modo, além de garantir que os requisitos foram atingidos também, é proporcionada uma experiência satisfatória aos usuários do sistema. Para saber mais sobre UX, acesse o *link* a seguir.

Fonte: Barbosa (2019).

ACESSAR

praticar

Vamos Praticar

O processo de construção de um sistema é composto por diversas etapas. Comumente, esse processo tem início na entrevista com o cliente, transformando-se em um documento de requisitos, em diagramas de caso de uso e, por fim, no *software* em si. Durante esse processo, protótipos são desenvolvidos. Em relação ao assunto de protótipos, assinale a alternativa correta.

- ☐ a) Os protótipos são desenvolvidos depois que o sistema já está pronto.

- ☐ **b)** Os protótipos são criados com finalidade de validar o que será desenvolvido.
- ☐ **c)** Os protótipos têm serventia apenas para validar os requisitos levantados.
- ☐ **d)** Os protótipos não são utilizados no processo de projeto de *software* .
- ☐ **e)** Os protótipos servem apenas para o teste de *software* .

Prototipação na Prática

Os protótipos, dentro do projeto de um software, são a criação e uma prévia de como serão as telas do sistema. A criação de protótipos permite a validação dos requisitos, bem como obter uma visão geral de como será o *software* em sua versão final.

Protótipo de Baixa Fidelidade

O termo fidelidade está relacionado a quanto o protótipo desenhado é fiel à versão final da tela no *software*. O protótipo de baixo nível é a ideia inicial de como será a tela do sistema que, geralmente, é feita por um desenho à mão da tela de um sistema.

Os protótipos de baixa fidelidade, também chamados de rascunhos ou sketches, são concebidos ainda na fase inicial, durante a concepção do sistema. Desenhadas geralmente à mão utilizando lápis, borracha e papel, essas representações são feitas de maneira rápida e superficial, apenas margeando a ideia do projeto e definindo superficialmente sua interação com o usuário, não se

preocupando ainda com elementos de layout, cores, disposições, etc. (NASCIMENTO, [s.d.], on-line).

Utilizando como referência o exemplo de uma tela de cadastro de aluno, a Figura 3.3 mostra o exemplo de um protótipo de baixa fidelidade, feito à mão, a maneira mais utilizada durante o desenvolvimento desse tipo de protótipo.

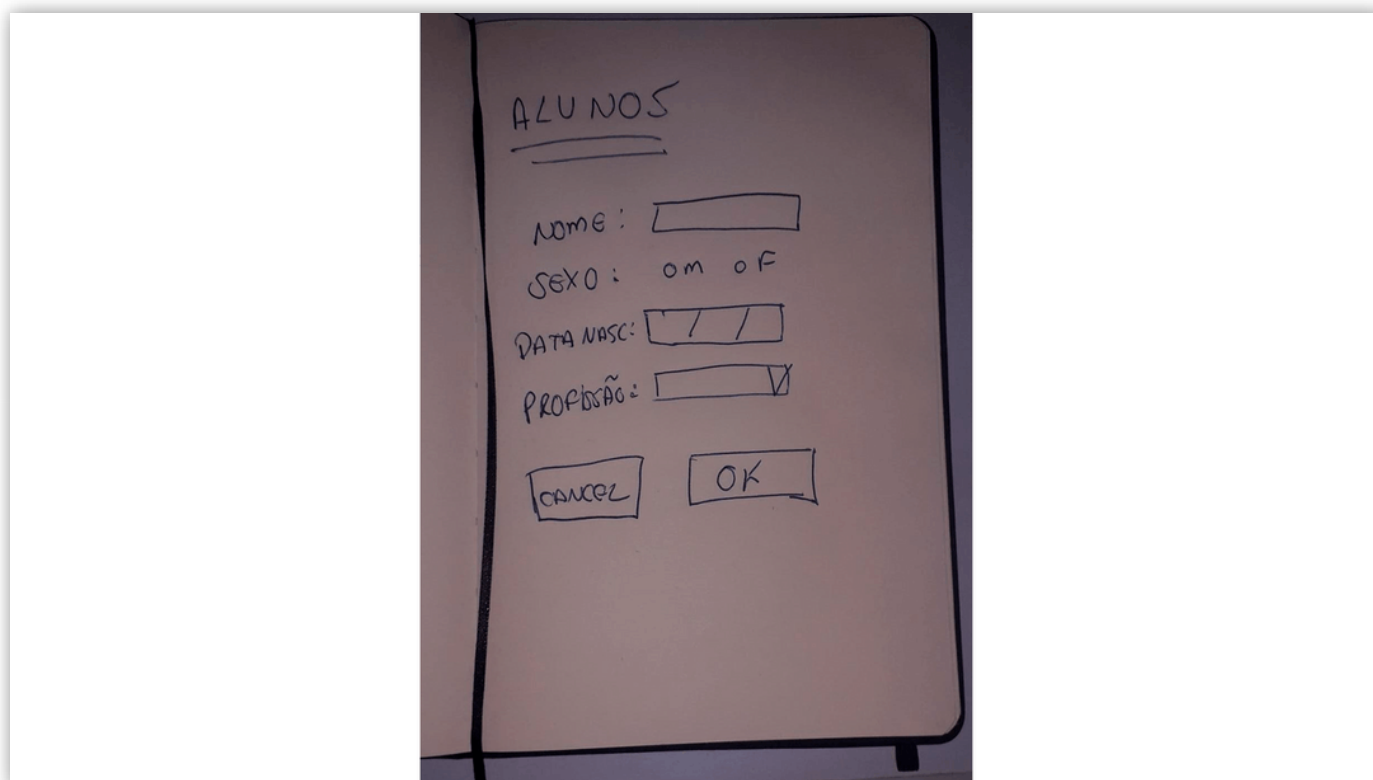


Figura 3.3 - Protótipo de Baixa Fidelidade

Fonte: Elaborada pelo autor.

#PraCegoVer: Uma folha de papel com a escrita alunos em letra maiúscula e sublinhada. Abaixo a palavra nome e um quadrado ao lado para preenchimento.

Abaixo a palavra sexo e opções para marcar M e F. Abaixo a palavra data de nascimento e um quadrado ao lado para preenchimento. Abaixo a palavra profissão e ao lado um quadrado para preenchimento. Abaixo dois retângulos um ao lado do outro. Dentro do primeiro retângulo a palavra cancel e no segundo a palavra OK.

O protótipo de baixa fidelidade permite que se faça realmente um rascunho

do sistema, que pode ser discutido em conjunto com a equipe de *software*, bem como com o cliente final. Para criar esse tipo de protótipo, ainda podem ser utilizados alguns softwares visuais, como o *Pencil Project* ou, até mesmo, o *Paint* , mas, por finalidade de ser objetivo, a melhor maneira de criar esse tipo de protótipo é com lápis e caneta.

Protótipo de Média Fidelidade



Os protótipos de média fidelidade podem ser considerados uma evolução do protótipo de baixa fidelidade. Segundo Nascimento ([s.d.], *on-line*), a partir da média fidelidade, os protótipos passam a apresentar a estrutura e o conteúdo da interface, já contendo o *layout* básico do projeto.

Esse tipo de protótipo pode ser chamado de *Wireframe* ou *Mockups* . No protótipo de média fidelidade, os campos aparecem com maior detalhamento (fidelidade), como os campos de data, janela, imagens, dentre outros.

Para a criação desse tipo de protótipo, existem diversas ferramentas. A ferramenta mais conhecida é o *Balsamiq Mockups Wireframes* . Utilizando o *Balsamiq* , vejamos um exemplo da evolução do protótipo de baixa fidelidade criado anteriormente, mas, agora, como um protótipo de média fidelidade, mostrado pela Figura 3.4.

Protótipo de uma tela web (Página Web https://www.sisacademico.com.br) com os seguintes elementos:

- Barra de endereço com ícones de navegação (voltar, avançar, fechar, casa) e uma barra de busca.
- Formulário com campos para:

- Nome:
- Sexo: ☐ M ☐ F
- Data de Nascimento: / / 
- Profissão: 

Botões de ação:

Figura 3.4 - Protótipo de Média Fidelidade

Fonte: Elaborada pelo autor.

#PraCegoVer: Tela de uma ferramenta de prototipagem. Bem acima da tela tem a opção para colocar o link de uma url. Bem a esquerda tem a palavra Nome e um quadro para colocar o nome. Abaixo a palavra sexo e opção para marcar entre M e F. Abaixo a palavra Data de Nascimento e um campo ao lado para preenchimento. E, ainda ao lado do campo um ícone de calendário. Abaixo a palavra Profissão e ao lado um campo para selecionar uma das profissões cadastradas. Abaixo dois retângulos um ao lado do outro. Dentro do primeiro retângulo a palavra cancel e no segundo a palavra OK.

É válido ressaltar que o *Balsamiq* é apenas uma ferramenta e que, em sua essência, os protótipos de média fidelidade reproduzem o sistema com maiores detalhes, se comparados aos protótipos de baixa fidelidade, ou seja, são mais fiéis ao sistema em sua versão final.

Protótipo de Alta Fidelidade

Um protótipo de alta fidelidade é aquele que reflete exatamente como a tela do sistema será, com exatos detalhes e cores. De um modo simples, um

protótipo de alto nível é como o sistema será.

No que se refere a esse tipo de prototipação, há diversas abordagens, dependendo, principalmente, da tecnologia utilizada para desenvolver o *software*, porque o protótipo pode ser desenvolvido utilizando *softwares* de *design*, como o *Adobe Photoshop*, e, posteriormente convertido para telas na linguagem/tecnologia em que o sistema será desenvolvido.

Por uma questão de gestão de projetos e otimização de tempo, o protótipo de alta fidelidade pode ser desenvolvido diretamente na tecnologia que o sistema funcionará, ou seja, as telas são criadas mesmo sem estar funcionando ou conectadas ao banco de dados.

Para exemplificar, vamos mostrar um protótipo de alto nível, desenvolvido diretamente na tecnologia que o sistema será desenvolvido. A Figura 3.5 mostra uma tela desenvolvida utilizando HTML e CSS.L.

O protótipo apresenta uma interface web com o título "ALUNOS" no topo. Abaixo dele, há quatro campos de entrada: "Nome:" com um campo de texto; "SEXO:" com duas opções de rádio, "M" e "F"; "DATA NASC:" com um campo de texto contendo o formato "dd/mm/aaaa"; e "PROFISSÃO:" com um menu suspenso que mostra "SELECIONE". No rodapé da interface, há dois botões azuis: "CANCEL" e "OK".

Figura 3.5 - Protótipo de Alta Fidelidade

Fonte: Elaborada pelo autor.

#PraCegoVer: Mostra uma tela desenvolvida em HTML e CSS.L. Bem acima e a esquerda está a palavra ALUNOS.

Abaixo a palavra nome e um quadro ao lado para preenchimento.

Abaixo a palavra sexo e duas opções para marcar entre M e F. Abaixo a palavra data de nascimento e um campo ao lado para preenchimento. Abaixo a palavra profissão e ao lado um campo para selecionar uma das profissões cadastradas. Abaixo dois retângulos azuis um ao lado do outro. Dentro do primeiro retângulo a palavra cancel e no segundo a palavra OK.

No desenvolvimento do exemplo, foi utilizada uma tecnologia empregada em um projeto. No caso do HTML e CSS, é permitido desenvolver as telas de maneira independente do funcionamento do sistema.

Saiba mais

Existe um conjunto de ferramentas para a criação de protótipos e para a interação entre as telas. Muito pode ser encontrado com a combinação de *software mockup* e *wireframe* . No entanto, duas opções interessantes são o *Invisionapp* e o *Sketch2code* . O *Sketch2code* permite criar telas HTML, a partir de um protótipo de baixo nível; basta tirar uma foto do protótipo desenhado e fazer upload na plataforma. É claro que esta é uma plataforma de reconhecimento de escrita e desenho, por isso, dependerá muito de como foi grafado.

O *InvisionApp* , por sua vez, permite a interação entre as funcionalidades, ou seja, utilizando as telas no formato de imagens criadas previamente, essa aplicação permite a interatividade entre estas. De modo geral, permite-se mostrar como será a aplicação, possibilitando a navegação antes mesmo de ser desenvolvida.

praticar

Vamos Praticar

A prototipação é o processo de construir telas, com o objetivo de validar os requisitos levantados com a proposta do sistema a ser desenvolvida. Durante uma entrevista com um cliente, um *Product Owner* pegou um bloco de anotações e começou a desenhar os protótipos do sistema. Selecione a alternativa correta, a qual apresenta o nome do protótipo desenhado.

- ☐ **a)** Protótipo de baixa fidelidade.
- ☐ **b)** Protótipo sem fidelidade.
- ☐ **c)** Protótipo de alta fidelidade.
- ☐ **d)** Protótipo de média fidelidade.
- ☐ **e)** Protótipo de autêntica fidelidade.

Mapa Mental

Mapas mentais são estruturas que permitem organizar as ideias por meio de um diagrama que permite organizar tópicos. Os mapas mentais são utilizados pelos estudantes, para organizar o aprendizado, e pelas mais diversas áreas do conhecimento.

Os mapas mentais são diagramas de uso pessoal, uma vez que contêm símbolos e informações organizadas na ordem e hierarquia que o usuário estabelecer. No entanto, os mesmos também podem ser elaborados coletivamente, desde que os símbolos, códigos e palavras-chave sejam de conhecimento de todos (HERMANN, 2005 apud BATISTA et al., 2013, p. 3).

Os mapas mentais permitem conectar diversos conceitos em um único diagrama.

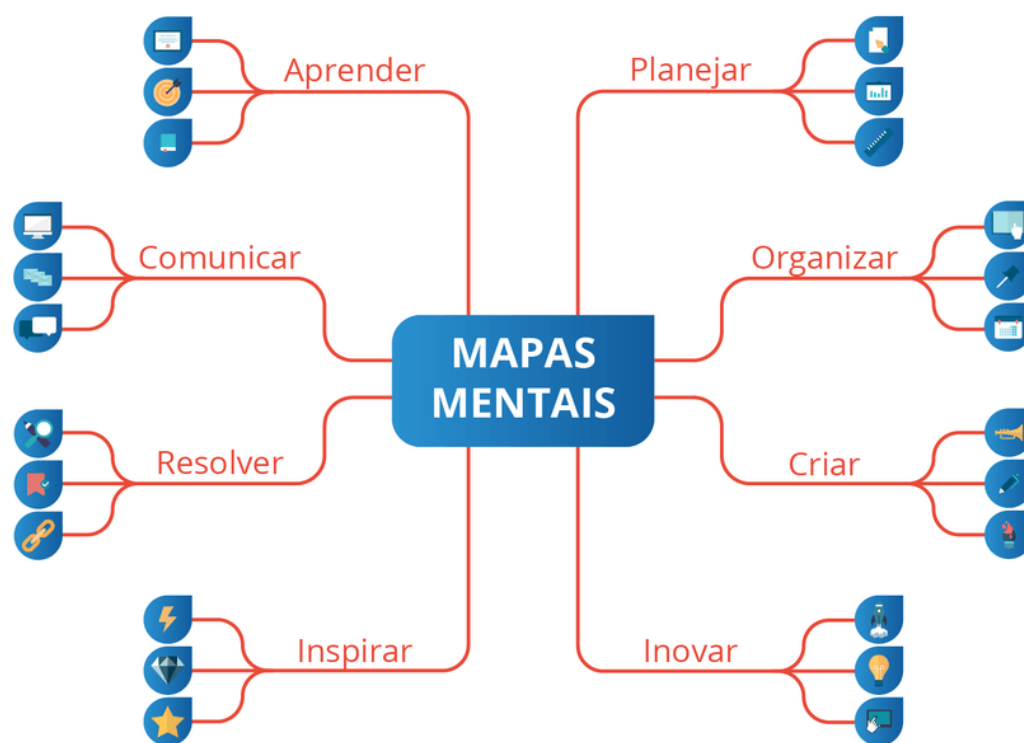


Figura 3.6 - Exemplo de Mapa Mental

Fonte: O Que... ([s.d.], on-line).

#PraCegoVer: Exemplo de mapa mental onde ao centro há um retângulo azul com a escrita “mapas mentais” conectado a quatro linhas para a direita e quatro linhas para a esquerda, uma abaixo da outra, sendo cada linha representada por uma palavra-chave que se conecta com três figuras representativas. Acima e à esquerda tem a primeira linha com a escrita “aprender” que se conecta com três figuras representando uma tela de computador, um alvo com uma flecha no meio e uma folha. Na linha abaixo a escrita “comunicar” conectando-se com um monitor, várias folhas juntas e balões de conversa. Na terceira linha a escrita “resolver” conectado com uma lupa cruzada por um lápis, uma seta e uma corrente. E na última linha à esquerda a escrita “Inspirar” conectado com as figuras de um raio, um diamante e uma estrela. Do lado direito do retângulo azul acima e à direita tem a primeira linha com a escrita “planejar” que se conecta com três figuras representando uma folha com uma caneta, um gráfico de barras e um arquivo. Na linha abaixo a escrita “organizar” conectando-se com um caderno aberto e uma mão indicativa, um pin marcador e um calendário. Na terceira linha a escrita “criar” conectada com uma corneta, uma caneta e um pinga de tinta preta. E na última linha a escrita “inovar” conectada com um foguete,

uma lâmpada e uma tela de monitor com uma mão indicativa.

Segundo Wanderley et al. (2012), esse diagrama é usado na conexão de palavras, ideias e conceitos a um conceito central, sendo utilizado para visualizar, classificar, estruturar conceitos e gerar novas ideias nos mais diversos tipos de setores, não somente para a tecnologia. Os elementos de um mapa mental são ordenados de forma intuitiva, ou seja, sua ramificação será feita de acordo com a compreensão do usuário sobre o tema.

Os principais benefícios em utilizar mapas mentais são: organização, utilização de palavras-chave, associação, agrupamento de ideias, memória visual, criatividade e inovação com simplicidade (BUZAN, 2003 apud WANDERLEY et al. 2012 p. 3).

No contexto de projetos e gestão de projetos em sistemas de informação, você compreenderá o mapa mental como um canivete suíço. Isso acontece porque são diversas as aplicações de um mapa mental nesse contexto. O emprego de mapas mentais é tão dinâmico que a própria metodologia *Scrum* pode ser explicada por meio desse diagrama.



Figura 3.7 - Mapa Mental do Scrum

Fonte: Elaborada pelo autor.

#PraCegoVer: Exemplo de mapa mental do scrum com fundo azul bem clarinho. Acima e ao centro uma figura representando os processos do scrums sendo iniciado por quatro retângulos marrons sobrepostos seguindo para uma sequência de quatro folhas juntas e após 24 horas e 30 dias representados por um loop um em cima do outro segue para uma caixa marrom finalizando o desenho.

Abaixo desta figura tem outra figura. Bem ao centro está um retângulo azul com a escrita scrum conectado por quatro linhas para a esquerda e três linhas para a direita uma em baixo da outra, sendo cada linha representando uma palavra chave.

A ordem das palavras iniciando pela primeira linha da esquerda do scrum é: Princípios, papéis, ferramentas e por último cerimônias.

A ordem das palavras iniciando pela primeira linha da direita do scrum é: metodologias ágeis, origem e artefatos. De artefatos saem outras quanto linhas em cada uma delas, de cima para baixo, escrito: Product Backlog, Sprint

Backlog, Definição do Ponto e por último Incremento.

Para a criação e desenvolvimento de mapas mentais, existem diversas ferramentas no mercado, para a criação de mapas *on-line*. Na construção dos exemplos, foi utilizado o *MindMister*, plataforma *on-line* e colaborativa, a qual pode ser acessada pelo site da *MindMister*.

Vejamos alguns exemplos baseados no que você já estudou.

- **Menu de Navegação:** os mapas mentais são comumente utilizados para projetar a navegação de um sistema, mapeando quais serão as telas existentes e como chegar de uma até a outra.

Veja um exemplo, na figura a seguir, sobre um Mapa Mental de Menu do Sistema.



Figura 3.8 - Mapa Mental Menu do Sistema

Fonte: Elaborada pelo autor.

#PraCegoVer: O fundo da figura é de um azul bem clarinho. Acima e a esquerda da folha tem um retângulo azul escrito Menu Principal. Desse retângulo azul saem quatro linhas. Cada linha tem um nome. A primeira linha

tem o nome Alunos em negrito conectado com outras três linhas cada uma com um nome que são Cadastro, Presença e Nota. A segunda linha tem o nome Professor em negrito conectada com outras duas linhas cada uma com um nome que são Cadastro e Pesquisa. A terceira linha tem o nome Disciplina em negrito conectada com outras duas linhas cada uma com um nome que são Cadastrar e Associar com o Professor. A quarta e última linha tem o nome Relatórios em negrito conectada com outras três linhas cada uma com um nome que são Frequência, Nota e Horas Professor.

- **Integrando Requisitos e Scrum** : outro exemplo da aplicação de mapas mentais é no mapeamento dos requisitos em uma visão pelo *Scrum* . Brod (2012) sugere um exemplo utilizando o Portal Ibero-Americano de Inovação, mostrado pela Figura 3.9, em que se pode ver toda a estrutura de um projeto.

Veja o exemplo, na figura a seguir, de um Mapa Mental com *Scrum* .

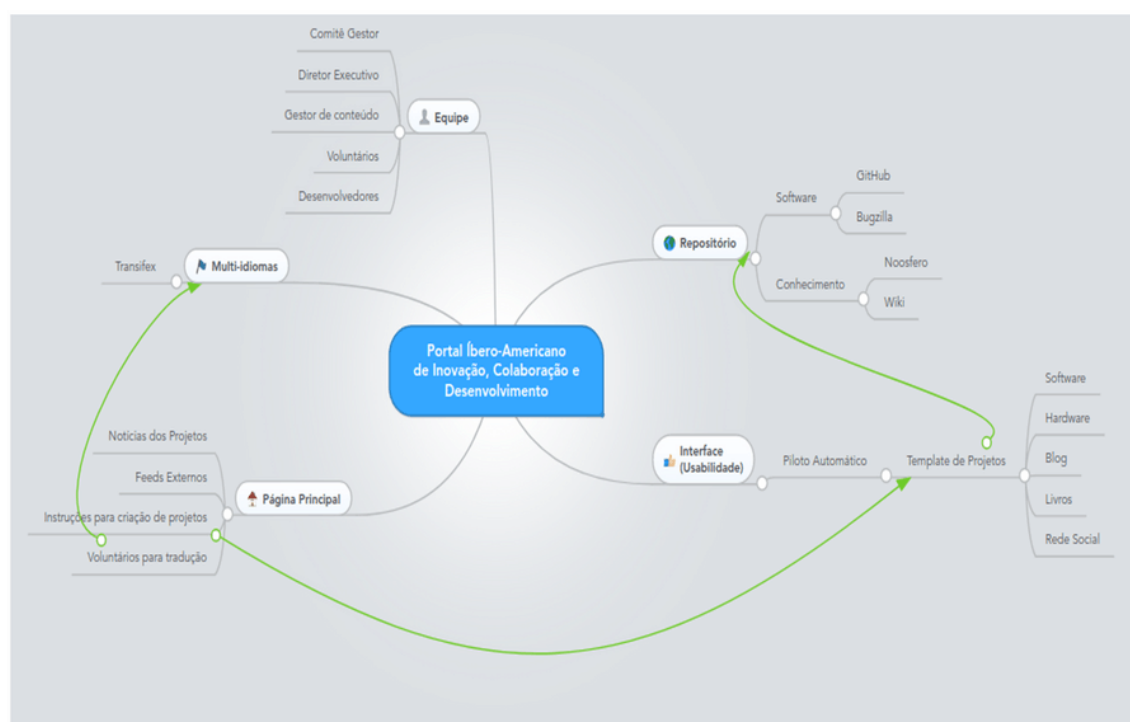


Figura 3.9 - Mapa Mental com Scrum

Fonte: Brod (2012, on-line).

#PraCegoVer: Exemplo de um mapa mental com scrum. A figura tem o fundo

azulado.

Bem ao centro um retângulo azul com a escrita Portal Íbero-Americano de Inovação, Colaboração e Desenvolvimento. Deste retângulo saem três linhas para a direita e duas linhas para a esquerda. A primeira linha a esquerda tem no final dela um retângulo branco que tem escrito dentro dele a palavra Equipe. Deste retângulo saem cinco linhas cada uma delas com um nome que são: Comitê Gestor, Diretor Executivo, Gestor de Conteúdo, Voluntários e Desenvolvedores. A segunda linha a esquerda tem no final dela um retângulo branco que tem escrito dentro dele a palavra Multi-idíomas. Deste retângulo sai uma linha com o nome transifex. A terceira linha a esquerda tem no final dela um retângulo branco que tem escrito dentro dele a palavra Página Principal. Deste retângulo saem quatro linhas cada uma delas com um nome que são: Notícias dos Projetos, Feeds Externos, Instruções para criação de projetos e Voluntários para tradução. A primeira linha a direita tem no final dela um retângulo branco que tem escrito dentro dele a palavra Repositório. Deste retângulo saem duas linhas cada uma delas com um nome que são: Software e Conhecimento. Da linha como nome de Software tem duas linhas com os nomes GitHub e Bugzilla. Da linha como nome Conhecimento saem outra duas como os nomes Noosfero e Wiki. A segunda linha a direita tem no final dela um retângulo branco que tem escrito dentro dele a palavra Interface (usabilidade). Deste retângulo sai uma linha com o nome Piloto Automático. Desta linha sai uma linha como nome Template de Projetos. Desta saem cinco linhas como os nomes Software, Hardware, Blog, Livros e Rede Social. Na figura tem ainda três linhas verdes com flechas nas suas pontas. Uma das linhas verde liga a linha Voluntários da tradução com a linha Multi-Idíomas. A outra linha liga a linha Instruções para criação de projetos com a linha Template de projetos e a última linha verde liga a linha Repositório com a Template de projetos.

Você pode aprender que os mapas mentais são recursos dinâmicos, os quais podem ser utilizados em qualquer etapa do projeto de *software*. Esses mapas podem auxiliar desde a conversa com o cliente e o mapeamento de entidades em um banco de dados, até mesmo no levantamento de requisitos.

praticar

Vamos Praticar

Os mapas mentais são estratégias para a organização de ideias de maneira visual, os quais podem ser aplicados nas mais diversas áreas do conhecimento. No desenvolvimento de *software*, que pode ser composto por diversas metodologias, técnicas e diagramas, os mapas mentais são recursos que podem auxiliar.

Selecione a alternativa correta, sobre qual é a etapa em que o mapa mental é aplicado.

- ☐ **a)** No levantamento de requisitos.
- ☐ **b)** Para idealizar o comportamento das telas.
- ☐ **c)** Na modelagem de banco de dados.
- ☐ **d)** Durante o desenvolvimento do sistema.
- ☐ **e)** Todas as alternativas estão corretas.

Utilizando UML antes de Codificar

A etapa de projeto de um *software* tem, por objetivo, documentar o sistema, de tal maneira a garantir que o *software* a ser desenvolvido reflita as necessidades do cliente levantadas no documento de requisitos.

A UML, por sua vez, ainda que se chame Linguagem de Modelagem Unificada, é um conjunto de diagramas que permitem o mapeamento do sistema. Durante essa etapa de modelagem conceitual de software, abordaremos o principal diagrama com ênfase em programação: o diagrama de classes.

Programação Orientada a Objetos

Antes de iniciarmos com a criação de um diagrama de classes, é importante compreendermos alguns conceitos básicos sobre a orientação a objetos. A programação orientada a objetos é um paradigma de programação de computadores, o qual baseia-se em objetos que pertencem ao mundo real do sistema a ser desenvolvido.

A programação orientada a objetos é composta pelos seguintes elementos

principais:

- **objeto:** representa um objeto do mundo real, lembrando que o mundo real é o escopo do sistema. No exemplo do sistema acadêmico, um objeto pode ser um aluno, o professor ou sala de aula;
- **atributos:** atributos são características de um objeto. Tomando, como exemplo, o objeto “aluno”, são atributos o nome, o endereço e cpf;
- **métodos:** são ações realizadas pelos objetos. Tomando, como exemplo, o objeto “aluno” são métodos: `cadastrar_aluno()`, `editar_aluno()` e `pesquisar_aluno()`;
- **classes:** as classes são definições prévias de uma estrutura de um objeto. Uma analogia é como se a classe fosse a receita de bolo de cada objeto.

Diagrama de Classes

Dando sequência ao projeto de *software*, o diagrama de classes permite projetar todo o comportamento de um sistema orientado a objetos. Ao desenvolver um diagrama de classes, não precisamos determinar qual linguagem de programação será utilizada.

reflita

Reflita

Os diagramas de classes são encontrados, com maior frequência, na modelagem de sistemas orientados a objetos. Um diagrama de classes mostra um conjunto de classes, interfaces e colaborações e seus relacionamentos. Use os diagramas de classes para fazer a modelagem da visão estática do projeto de um sistema. Na maioria dos casos, isso envolve a modelagem do vocabulário do sistema, a modelagem de colaborações ou a modelagem de esquemas. Os diagramas de classes são importantes não só para a visualização, a especificação e a documentação de modelos estruturais, mas também para a construção de sistemas executáveis por intermédio de engenharia direta e reversa.

Fonte: Booch, Rumbaugh e Jacobson (2006, p. 175).

A criação do diagrama de classes é realizada por intermédio de *softwares* UML, por exemplo, o *StarUML*. Durante a criação desse diagrama, é possível pensar

em como será desenvolvido o código, sem que haja a necessidade de codificar, economizando tempo com a equipe. Veja o exemplo de um diagrama de classes.

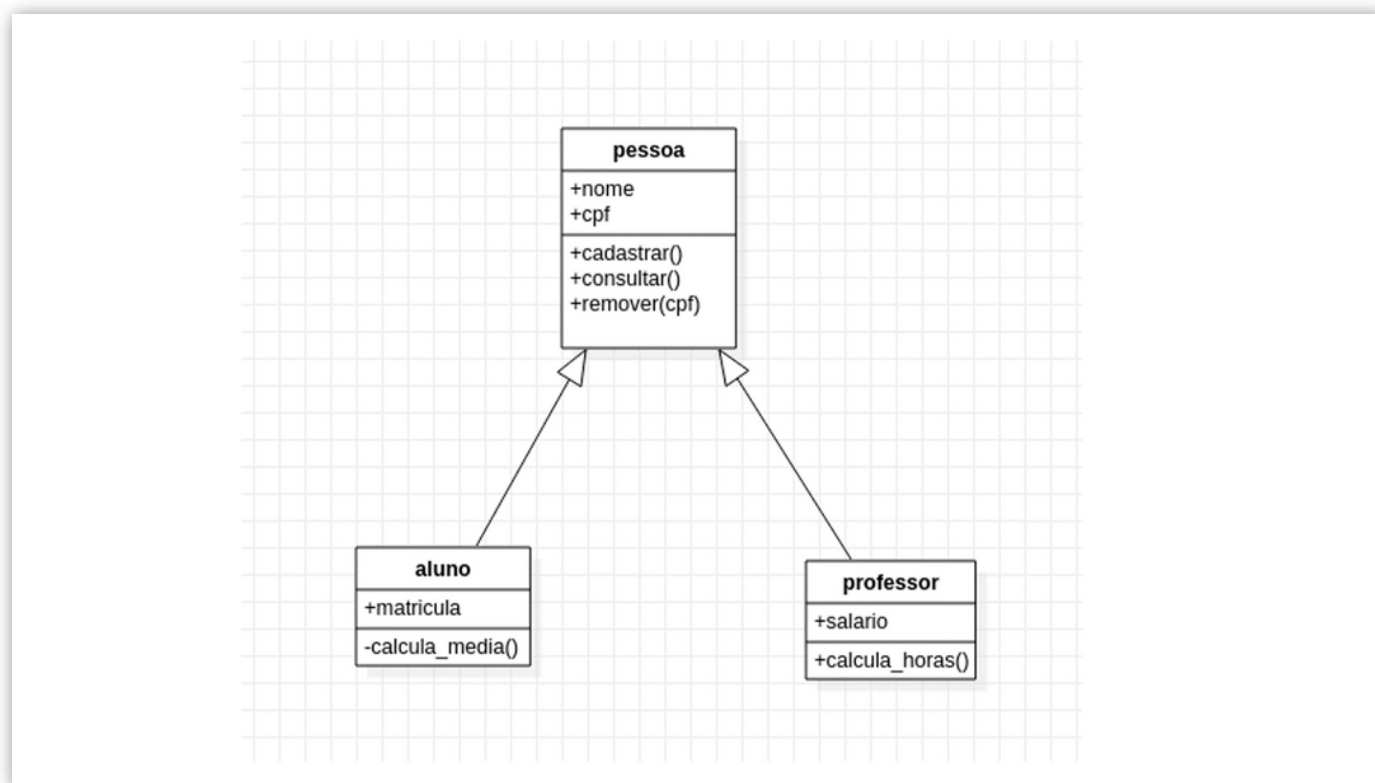


Figura 3.10 - Diagrama de Classes

Fonte: Elaborada pelo autor.

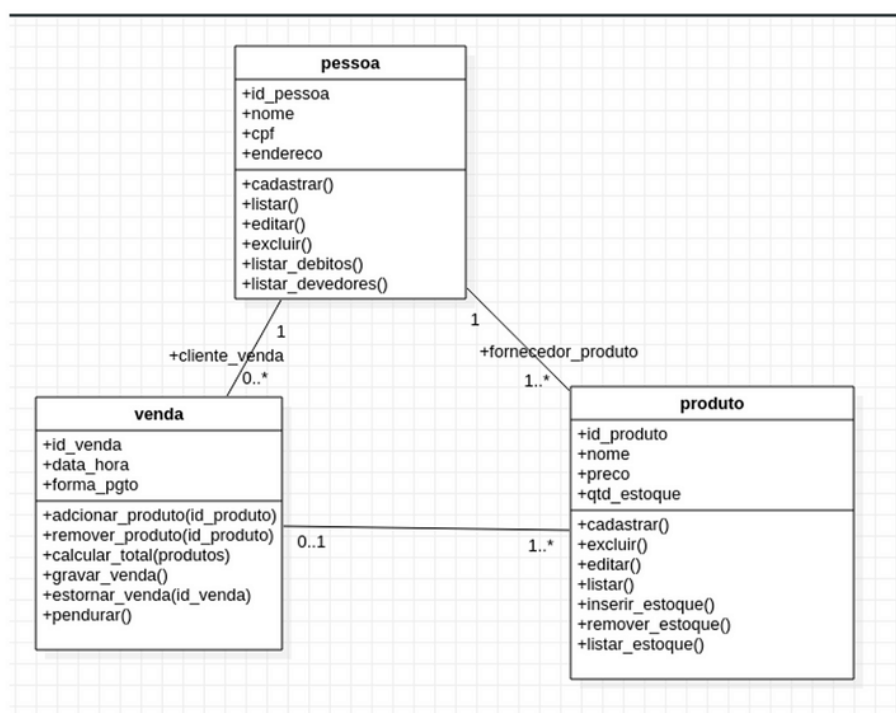
#PraCegoVer: Exemplo de diagrama de classes com o fundo quadriculado cinza.

Três retângulos dispostos. Um retângulo acima e ao centro e dois retângulos abaixo. Cada retângulo abaixo contém uma seta indicativa para o retângulo de cima.

Cada retângulo está dividido com três linhas. O primeiro retângulo tem a escrita pessoa e na linha abaixo as palavras +nome e +cpf e na terceira linha as palavras + cadastrar() +consultar() e +remover (cpf). O segundo retângulo à esquerda e abaixo tem a escrita aluno e na linha abaixo a palavra matrícula e na linha de baixo a palavra calcula_média() O terceiro retângulo à direita e abaixo tem a escrita professor e na linha abaixo a palavra +salário e na última linha deste retângulo a palavra +calcula_horas()

No diagrama acima, note que professor e aluno possuem poucos atributos. Isso acontece pois essas classes são uma herança da classe pessoa. A **herança** é um dos mais importantes recursos para o reuso de código. Isso significa que todos os atributos e métodos de pessoa também pertencem ao aluno e ao professor.

Em um outro exemplo, vejamos um diagrama que representa um sistema de vendas. A classe pessoa representa cliente e fornecedor. Já a venda representa as operações de vendas, bem como existe a classe produto.



*Figura 3.11 - Diagrama de Classes Venda de Produtos
essa é a fonte*

#PraCegoVer: Exemplo de diagrama de classes com o fundo quadriculado cinza. Nesta figura tem três retângulos. Um retângulo no centro acima e dois retângulos abaixo.

Cada um dos retângulos abaixo é ligado por uma linha com um nome. Cada retângulo está dividido com três linhas. O primeiro retângulo tem o nome de pessoa. Neste mesmo retângulo abaixo tem as palavras id da pessoa, Nome, CPF e endereço. Abaixo, mesmo retângulo, tem as palavras: cadastrar, listar, editar, excluir, listar débitos e listar devedores. O segundo retângulo está à

esquerda e abaixo do retângulo com o nome pessoa. Este retângulo tem o nome venda. Neste mesmo retângulo abaixo tem a palavra id venda, data hora e forma de pagamento. Abaixo, mesmo retângulo, tem a palavra adicionar produto e id produto entre parênteses. Remover produto e id produto entre parênteses, calcular total e produtos entre parênteses. Gravar venda, estornar venda e id venda entre parênteses e pendurar. O terceiro retângulo está à direita e abaixo do retângulo com o nome pessoa. Este retângulo tem o nome produto. Neste mesmo retângulo abaixo tem as palavras: id produto, nome, preço, quantidade de estoque. E abaixo, mesmo retângulo, tem as palavras: cadastrar, excluir, editar, listar, inserir estoque, remover estoque e listar estoque. O retângulo com nome de pessoa está ligado ao com nome venda por uma linha que tem o nome de cliente venda. Na ponta desta linha de nome pessoa tem o número um e do outro lado o número zero, dois pontinhos e um asterisco. O retângulo com nome pessoa está ligado ao com nome produto por uma linha que tem o nome fornecedor produto. Na ponta desta linha de nome pessoa tem o número um e do outro lado o número um, dois pontinhos e um asterisco. O retângulo com nome de venda está ligado ao com nome produto por uma linha. Na ponta desta linha de nome venda tem o número zero, dois pontinhos e o número um. E do outro lado o número um, dois pontinhos e um asterisco.

Note que as classes estão conectadas, ou seja, relacionadas e, nesse diagrama, não existe o conceito de herança. Quando abordamos a conexão entre as classes, é um conceito similar aos relacionamentos da modelagem de banco de dados, em que, no modelo de classes, os relacionamentos entre as classes representam como os objetos irão se relacionar quando forem codificados.

Para mapear o comportamento entre os objetos de um sistema, existe o conceito de **multiplicidade**. A multiplicidade indica a quantidade de objetos de uma classe que irá se relacionar com outra, colocando limites (inferior e superior). O conceito de multiplicidade é muito similar ao conceito de cardinalidade.

Vamos analisar a multiplicidade dos relacionamentos. Venda → 1..* Produto significa que cada objeto da classe venda tem de ter pelo menos um objeto da

classe produto associado, podendo ter muitos. Produto → 0..* Venda significa que um objeto produto não precisa estar associado a nenhum objeto da venda, mas pode estar associado a muitos.

Venda → 1 Pessoa, essa multiplicidade significa que um objeto da venda tem de estar associado obrigatoriamente a um cliente (objeto da classe pessoa). Pessoa → 0..* Venda representa que um objeto da classe pessoa pode ser criado sem estar associado à classe Pessoa, mas pode ter muitos objetos associados à venda.

Por fim, o relacionamento entre produto e fornecedor (Pessoa). Pessoa (fornecedor) → 1..* Produto representa que, para se instanciar um objeto do fornecedor, deve-se ter, no mínimo, um produto associado. Produto → 1 Pessoa (fornecedor), representando que cada objeto produto deve ter, obrigatoriamente, um fornecedor.

praticar

Vamos Praticar

O desenvolvimento de *software* envolve um conjunto de metodologias e processos da área da computação, bem como da gestão de projetos. A UML, como uma linguagem para modelagem visual de *software*, oferece um conjunto de diagramas que tem, como objetivo, documentar o desenvolvimento de um sistema, de tal modo que também suporta a gestão de projetos de *software*.

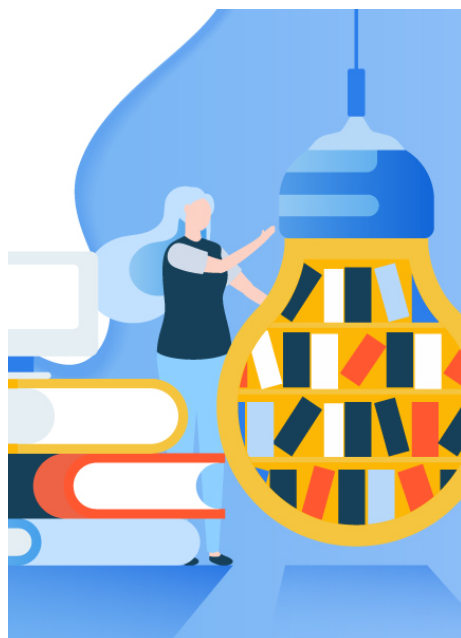
No contexto da UML, selecione a alternativa correta, que contenha o diagrama utilizado para projetar o código orientado a objetos.

- ☐ a) Diagrama de Requisitos.
- ☐ b) Diagrama de Casos de Uso.

- ☐ **c)** Diagrama de Banco de Dados.
 - ☐ **d)** Diagrama de Classes.
 - ☐ **e)** Diagrama relacional.
-

indicações

Material Complementar



LIVRO

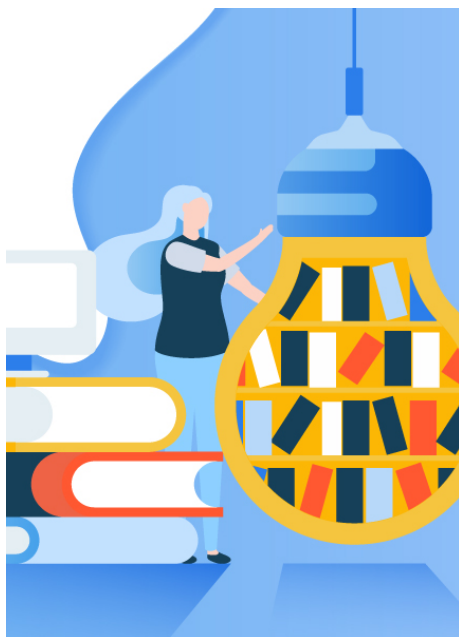
UML: guia do usuário

Grady Booch, James Rumbaugh e Ivar Jacobson

Editora: Elsevier Brasil

ISBN: 8535217843, 9788535217841

Comentário: para aprofundar seus conhecimentos sobre o desenvolvimento de diagrama de classes, é recomendada a leitura do capítulo 8 do livro.



LIVRO

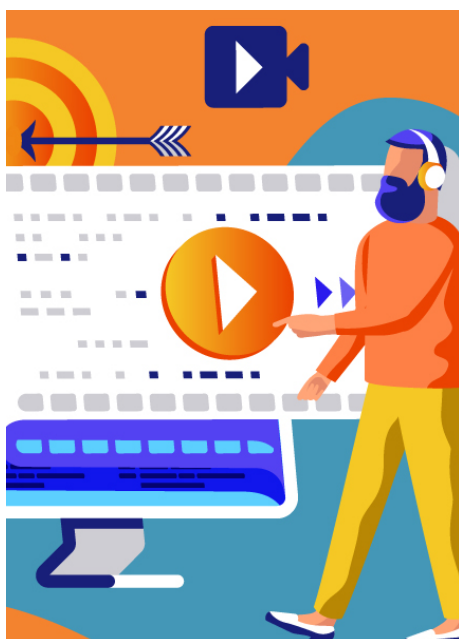
User Experience Design: Como criar produtos digitais com foco nas pessoas

Rogério Pereira

Editora: Casa do Código

ISBN: 8594188676, 9788594188670

Comentário: durante esta unidade, foi abordado o conceito de prototipação e sua aplicação em *User Experience* . Essa sugestão de leitura é para que você possa compreender melhor o conceito de UX e sua importância em diversos setores da tecnologia.



FILME

Os estagiários

Ano: 2013

Comentário: aborda ideias, inovação e trabalho em equipe, trazendo uma visão sobre trabalhar em uma gigante da Internet. Para conhecer mais sobre o filme, acesse o trailer a seguir.

TRAILER

conclusão

Conclusão

Durante esta unidade, você aprendeu novas ferramentas para integrar ao seu projeto de *software* . Por meio do desenvolvimento de protótipos de tela, você poderá incorporar o seu projeto de *software* e garantir que os requisitos sejam atingidos, bem como realize alterações no sistema antes mesmo de codificá-lo.

Os mapas mentais permitirão que você organize o seu projeto em diversos níveis. Utilizando o UML, você poderá desenvolver o projeto orientado a objetos.

referências

Referências Bibliográficas

BARBOSA, C. C. I. **A experiência do cliente está na sua mão** : estudo do aplicativo da marca Renner com base no conceito de ux design. 2019. Trabalho de Conclusão de Curso (Graduação em Comunicação Social - Publicidade e Propaganda) - Universidade Federal de Pampa, Rio Grande do Sul, 2019. Disponível em: <https://bit.ly/38iTBqL> . Acesso em: 7 jan. 2020.

BARCELOS, G. T. et al. Uso educacional de tablets: estudo de caso na formação inicial de professores de Matemática. **RENOTE - Revista Novas Tecnologias na Educação** , v. 11, n. 1, 2013.

BATISTA, Silvia Cristina Freitas et al. **Mapas Mentais com Tecnologias Digitais: Reflexões na Formação Inicial de Professores de Matemática. Challenges 2013: Aprender a Qualquer Hora e em Qualquer Lugar, Learning Anytime Anywhere**, p. 111, 2013..

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML : guia do usuário**. Elsevier Brasil, 2006.

BROD, C. Scrum: Product Backlog e Mapas Mentais. **Brodtec** , 2012. Disponível em: <http://brodtec.com/diario/scrum-product-backlog-e-mapas-mentais> . Acesso em: 8 jan. 2020.

BUZAN, T. **The Mind Map Book** . BBC Active, 2003.

HERMANN, W.; BOVO, V. **Mapas mentais** – enriquecendo inteligências. Campinas, 2005.

NASCIMENTO, T. A importância dos protótipos no desenvolvimento de sistemas. **Thiagonasc** . Disponível em: <http://thiagonasc.com/desenvolvimento-web/a-importancia-dos-prototipos-no-desenvolvimento-de-sistemas> . Acesso em: 20 jan. 2020.

O QUE é Mapa Mental? **Mapa Mental** . Disponível em: <https://www.mapamental.org/dicas/o-que-e-mapa-mental/> . Acesso em: 8 jan. 2020.

PINHEIRO, A. P. da S. UX design introduzido no desenvolvimento de interfaces gráficas. **Repositório** , 2016. Disponível em: <https://repositorio.uniceub.br/jspui/bitstream/235/9445/1/21400530.pdf> . Acesso em: 21 jan. 2020.

SPINOLA, R. Engenharia de Software - Engenharia de Requisitos. **DevMedia** . Disponível em: <https://www.devmedia.com.br/artigo-engenharia-de-software-introducao-a-engenharia-de-requisitos/8034> . Acesso em: 7 jan. 2020.

WANDERLEY, F. et al. Uma Abordagem para o Desenvolvimento de um Sistema Clínico baseado em Arquétipos e Mapas Mentais. In: CONGRESSO BRASILEIRO DE INFORMÁTICA EM SAÚDE. 13., 2012, Pernambuco. **Anais** [...]. Pernambuco: CBIS, 2012. Disponível em: <https://bit.ly/3ctLGtV>. Acesso em: 5 jan. 2020.