

Unidade 01

**Introdução a Processo de Desenvolvimento de
*Software***

Introdução da Unidade

Com a existência de uma grande demanda por desenvolvimento de *softwares* nas empresas, é necessário compreender e organizar os passos que devem ser seguidos para o desenvolvimento de um *software*. Seguindo essa visão, nesta unidade vamos compreender e discutir os principais conceitos que corroboram no processo de desenvolvimento de um *software*.

Na unidade 01, vamos estudar os conceitos gerais sobre o processo de desenvolvimento de um *software*, além de entender como ele é organizado para que ao fim do seu ciclo atenda às necessidades do usuário. Para tal, estudaremos conceitos como: processos, riscos e técnicas de desenvolvimento de *software*.

Na aula 02, ainda usando conceitos da unidade 01, entraremos em mais detalhes sobre os processos que são utilizados, a fim de aprofundar nossos conhecimentos nas ferramentas que possibilitam o desenvolvimento de *software* com qualidade, onde o aluno poderá saber quais são os seus processos e como eles afetam o modo de desenvolver um *software*.

Objetivos

- Ter uma visão geral do processo de desenvolvimento de *software*;
- Entender como os processos de desenvolvimento de *software* afetam a experiência do usuário final.

Conteúdo programático

Aula 01 – Conceitos gerais do processo de desenvolvimento de *software*.

Aula 02 – Gerenciamento do Desenvolvimento.



Você poderá, também, **assistir as videoaulas** em seu celular, basta apontar a câmera para os **QR Codes** distribuídos neste conteúdo.



APRESENTAÇÃO

Utilize o QRcode para assistir!





MINICURRÍCULO

Utilize o QRcode para assistir!



Referências

COPYRIGHT IBM CORP. **Diretriz Diagrama de Comunicação**, 2006. Disponível em: https://www.cin.ufpe.br/~gta/rup-vc/core.base_rup/guidances/guidelines/communication_diagram_FFEA1B5.html?nodeId=3c470376. Acesso em: 26 fev. 2021.

OMG. **Business Process Model and Notation (BPMN)**, 2011. Disponível em: <https://www.omg.org/spec/BPMN/2.0/PDF>. Acesso em: 1 mar. 2021.

PRESSMAN, Roger S. **Engenharia de Software**. 1. ed. São Paulo: Person, 2011. 1056 p.

RUPPENTHAL, J. E. **Gerenciamento de riscos**. Universidade Federal de Santa Maria / Colégio Técnico Industrial de Santa Maria. 120 p. Santa Maria: Rede e-Tec Brasil, 2013.

UML. **What is UML**. 2005. Disponível em: <https://www.uml.org/what-is-uml.htm>. Acesso em: 25 fev. 2021.

Aula 01 - Conceitos gerais do processo de desenvolvimento de software

Olá estudante, como vai? Bom vamos começar nossos estudos falando sobre o que é processo de desenvolvimento de *softwares*. Você deve estar se perguntando para que tenho que estudar processo de desenvolvimento de *software*? Às vezes visualizamos o desenvolvimento de um *software* somente pela etapa de codificação.



VIDEOAULA 01

Utilize o QRcode para assistir!

Assista o vídeo para entender melhor o assunto.



Porém o processo de desenvolvimento vai além do código, é preciso compreender as necessidades do usuário, realizar a gestão de diversos pontos, a fim de garantir que sua construção se torne algo organizado, bem como, vire rotina para as pessoas que participam deste processo. Como exemplo, já imaginou combinar uma data para entrega de um *software*, e no final essa entrega ser postergada para seis meses depois. Esse é um dos diversos cenários possíveis quando não se utiliza um processo de desenvolvimento de *software* da maneira correta.

Existe uma necessidade de compreender que um *software* deve possuir a capacidade de tornar seu uso amigável para o ser humano (PRESSMAN, 2011), um *software* tem a ousadia de modificar o modo de viver do seu usuário, trazendo diferença em seu dia a dia.

Infelizmente, não existe uma receita mágica que se utilizando dela vamos resolver todos os problemas no desenvolvimento de um *software*, porém existem modelos de processos que utilizados de maneira correta, podem mitigar possíveis erros durante o seu desenvolvimento.

É necessário entender que o conjunto de um *software* se constitui em todo o seu processo, dentro de um escopo de trabalho a qual está sendo realizado, tais como: tempo, qualidade, recursos, esforço e custo. Todos esses pontos bem gerenciados resultam em um *software* bem-sucedido e que vai de encontro com as necessidades do cliente.

Dessa forma, possuir clareza que no processo de desenvolvimento de *software* existem várias pessoas que participam direta e indiretamente deste processo. O termo que é utilizado para descrever essas pessoas ou partes integrantes desse processo é *stakeholders*, são eles: usuário final, dono do produto, equipe de desenvolvimento e sócios.

Os *stakeholders* são todos aqueles que de algum modo podem influenciar dentro do processo de desenvolvimento de um *software*. Muitas vezes se faz necessário ser flexível com os *stakeholders*, até porque as necessidades deles mudam durante o processo de desenvolvimento.



VIDEOAULA 02

Utilize o QRcode para assistir!

Assista o vídeo para entender melhor o assunto.

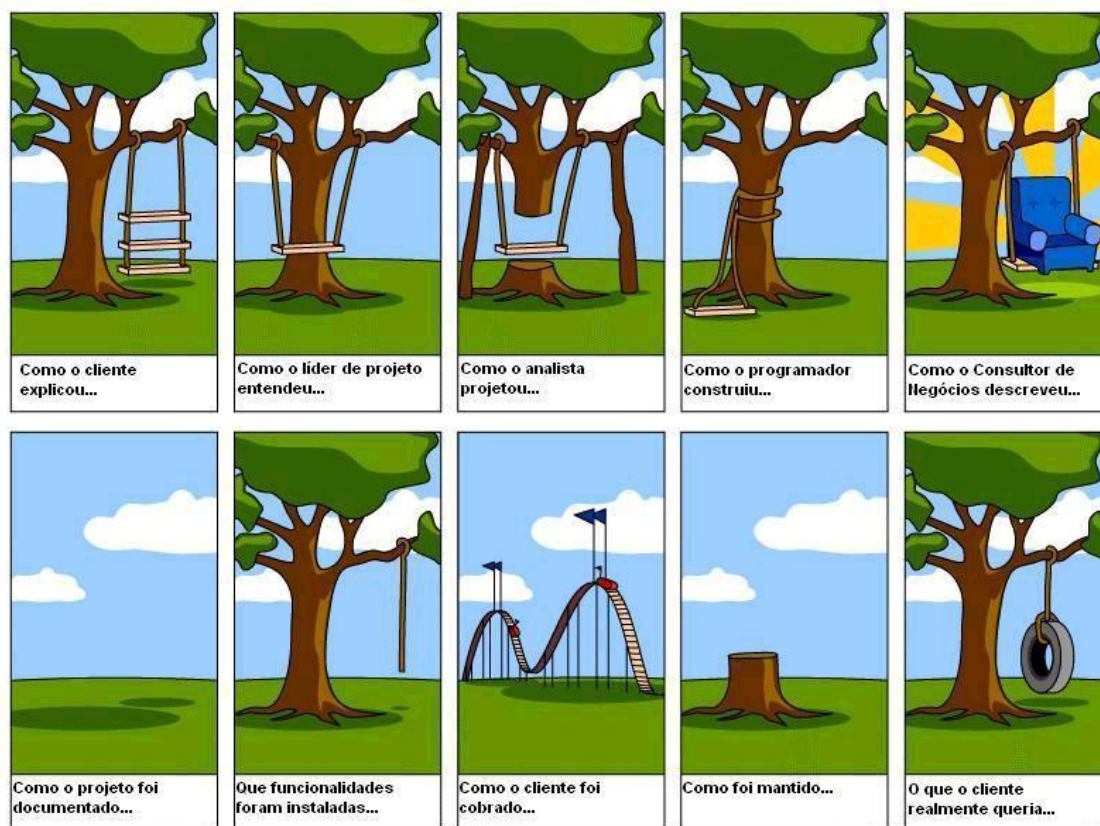


Para bem desenvolver um *software* é necessário compreender as dores do cliente, e assim conseguir identificar suas necessidades. Deste modo, conseguir entregar um *software* que realmente solucione um problema real, para isso é essencial que aconteça uma comunicação clara e eficaz entre todos os *stakeholders*.

Uma equipe bem integrada faz com que o desenvolvimento de um *software* ocorra de maneira orgânica e ao mesmo tempo organizada. O desenvolvimento pode ser comparado com a prática de um esporte em grupo, como exemplo o futebol; no qual cada jogador da equipe tem sua posição fixa, porém o objetivo de todos é fazer o gol.

Sendo assim o processo de desenvolver um *software* deve ser realizado em conjunto e nunca separado, buscando sempre mitigar riscos que possam acontecer no decorrer do seu desenvolvimento. Dessa forma, a comunicação com todos os *stakeholders* se torna extremamente importante, pois todos são “jogadores” do mesmo time e assim buscam o mesmo objetivo final.

Figura 01 - Processo de desenvolvimento errado



Fonte: disponível em: <<https://player.slideplayer.com.br/10/2889400/data/images/img2.jpg>>. Acesso em: 8 maio 2021.

Como podemos observar, a Figura 01 descreve como funciona um processo de desenvolvimento em que a comunicação e a flexibilidade entre as partes interessadas não foram vivenciadas de forma clara, a imagem deixa evidente que houve muitos ruídos na comunicação e que o resultado da entrega, foi muito diferente da imaginada pelo cliente. Deste modo, é preciso lembrar do exemplo do time de futebol, ou seja, todos têm em sua mente de forma clara o objetivo final, para que dessa forma todas as partes consigam desenvolver o projeto de maneira correta.

Desenvolver uma boa comunicação ou seguir processos já desenhados para o desenvolvimento de um *software*, mitigam riscos e auxiliam para que todo o processo discorra de maneira harmoniosa, porém é um mito pensar que seguindo um padrão já definido o desenvolvimento desse *software* vai ser garantido, conforme cita PRESSMAN (2011) existem alguns mitos administrativos e profissionais sobre o processo de desenvolvimento de *softwares*.

Um *software* possui um ciclo de vida, cada autor descreve esse ciclo de maneira diferente, porém todos eles vão abordar a necessidade de possuir um processo de desenvolvimento que torne possível prolongar o tempo de vida de um *software*.

Para facilitar a compreensão das necessidades do cliente e assim desenvolver um *software* com qualidade que atenda as metas das partes envolvidas, é que surge a necessidade de modelar um *software* com a visão de negócios.

Sendo assim, é preciso abstrair as dores do cliente de modo que seja possível descrever esse processo na visão de negócio, pensando em custos, riscos, tempo e recursos. Além disso, muitas vezes se observa a necessidade de descrever todo o processo de maneira visível para todas as partes interessadas: investidores, cliente e equipe de desenvolvimento.



VIDEOAULA 03

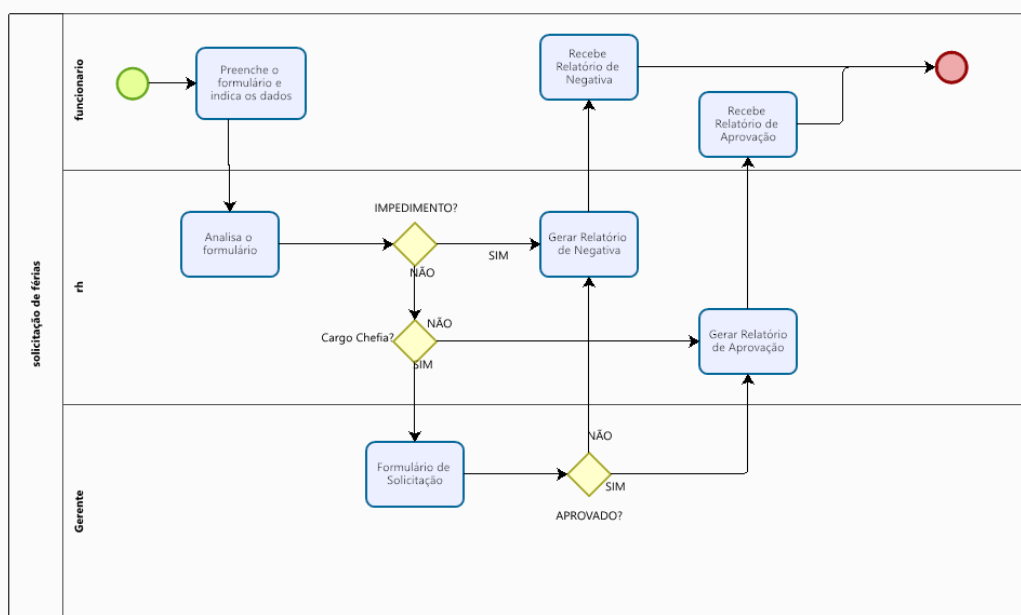
Utilize o QRcode para assistir!

Assista o vídeo que abordará mais conceitos sobre a aula.



Com essa necessidade é que surge o BPMN (*Business Process Model and Notation*), com esse processo é possível abstrair a visão de negócio da empresa que solicitou o *software*, tornando-se mais claro as regras de negócio, casos de uso, problemas vividos, transações, departamentos e pontos de decisão. A Figura 02 demonstra uma notação utilizando o BPMN que exemplifica um processo de retirada de férias, com essa estrutura podemos visualizar de forma geral o funcionamento desse fluxo dentro de uma empresa, portanto, temos descritas as suas regras de negócio.

Figura 02 - Exemplo notação BPMN



Powered by
bizagi
Modeler

Fonte: o autor (2021)

Para entendermos melhor, vamos explicar os componentes presentes no diagrama do BPMN, são eles:

- a) Os Pontos de início e fim do processo, são no formato de círculos.

Figura 03 - Pontos de início e fim BPMN



Fonte: o autor (2021)

- b) As Tarefas do processo, são no formato de quadrados ou retângulos.

Figura 04 -Tarefa BPMN



Fonte: o autor (2021)

- c) O Fluxo de sequência, são no formato de flechas que indicam ligações entre os componentes do fluxo no processo.

Figura 05 - Fluxo de sequência BPMN



Fonte: o autor (2021)

- d) Os Pontos de decisão ou *gateway's*, são no formato de um quadrilátero ou diamante.

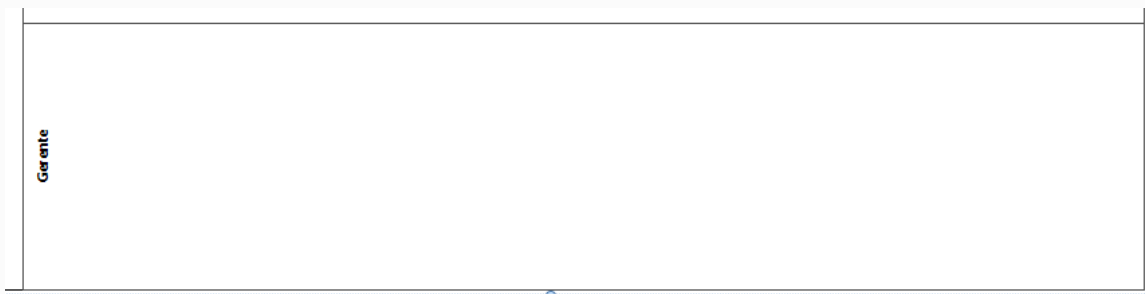
Figura 06 - Pontos de decisão BPMN



Fonte: o autor (2021)

- e) As Raias do processo, são os campos maiores onde estão divididos os departamentos/personagens do processo.

Figura 07 - Raia BPMN



Fonte: o autor (2021)

Com os elementos do BPMN descrevendo os processos de negócio de uma organização facilita a abstração por parte de todos os *stakeholders*, assim, tornando clara as regras de negócio, e observando quais personagens/atores devem tomar decisões e em qual etapa isso deve ocorrer. Desta forma, é possível compreender com clareza quem utiliza o sistema, qual é o problema a ser resolvido, e assim definir um escopo inicial para que o processo de desenvolvimento seja um sucesso.

INDICAÇÃO DE ACESSO

Acesse o site oficial do BPMN. Disponível em: <<https://www.bpmn.org/>>. Acesso em: 8 maio 2021.

Somente após a abstração desses pontos de negócio é que será possível estabelecer vínculos com as partes interessadas e assim definir metas e o objetivo final do *software* que deve ser desenvolvido.

Buscando obter esta visão de negócio e organizar o processo de desenvolvimento de *software* é que surge o UML (*Unified Modeling Language*) e o RUP (*Rational Unified Process*), nas próximas aulas vamos estudar mais a fundo esses recursos e como eles podem auxiliar no dia a dia o processo de desenvolvimento de *software*, a fim de buscar realizar este processo com sucesso.

Aula 02 - Gerenciamento do Desenvolvimento

Olá aluno, pronto para a segunda aula desta unidade? Nesta aula vamos apresentar o gerenciamento do processo de desenvolvimento e a UML (*Unified Modeling Language*), que assim como o BPMN surge com a necessidade de facilitar a compreensão sobre o *software* e seus processos, portanto, documenta o funcionamento, o desenvolvimento do *software* e facilita atividades futuras.



VIDEOAULA 01

Utilize o QRcode para assistir!

Assista a videoaula na qual iremos abordar mais aspectos do desenvolvimento do *software*.



Gerenciar o desenvolvimento de um *software* é visto como um desafio a ser vencido, porém existem pontos a serem considerados como:

- a) Gerenciamento de risco;
- b) Gerenciamento de recursos;
- c) Gerenciamento de tempo.

No gerenciamento de risco é preciso identificar o que ameaça o desenvolvimento do *software* e o que pode tornar-se um ameaça futura, após a sua entrega. Existem riscos que não podem ser retirados do escopo do projeto, porém podem ser mitigados.

Ao ser identificado um risco, é importante ponderar qual é o real índice desse risco dentro do processo, é de alto índice, médio ou baixo, para assim focar em produzir medidas que diminuam ou que até mesmo retirem do escopo do projeto esse risco, sendo assim fortalecendo um processo de desenvolvimento seguro.

Deste modo, é necessário mapear quais são as atividades necessárias para realizar, de modo que seja possível reduzir a quantidade de riscos que um *software* pode passar, ou seja, o *software* vai estar menos exposto aos riscos.

Figura 01 - Nível de exposição

$$\text{Nível de exposição} = \frac{\text{Risco}}{\text{Medidas de controle adotadas}}$$

Fonte: Ruppenthal (2013)

Conforme demonstrado na Figura 01, o nível de exposição está relacionado com os riscos, divididos pelas quantidades de medidas de controle adotadas no projeto, quanto mais medidas de controle forem adotadas menor será o nível de exposição ao risco do *software*, garantindo maior segurança durante todo o seu processo de desenvolvimento.

Muitas vezes o gerenciamento de recursos é visto como muito burocrático e que mata a vida de um *software* antes mesmo dela iniciar, pois, ele precisa avaliar os recursos financeiros e humanos; quantidade de dinheiro necessário para realizar o desenvolvimento; quantidade de pessoas nas equipes e como elas serão divididos; infraestrutura existente e ambiente de desenvolvimento.

Podemos dividir os recursos em dois grandes grupos, sendo eles:

- a) Recursos de trabalho, pessoas e equipamentos necessários para discorrer as atividades;
- b) Recursos materiais, são aqueles de uso contínuo ou de necessidade momentânea no projeto.

Tanto um quanto outro dependem de pessoas e dinheiro, dessa forma é importante desenvolver na equipe um propósito que seja claro, um objetivo transversal que vai além de simplesmente desenvolver um *software*.

No gerenciamento de tempo, encontramos diversas perspectivas, porém é preciso compreender antes de tudo a necessidade do cliente, e montar um cronograma. Algumas metodologias defendem o desenvolvimento com um cronograma de tarefas e no final uma entrega única, já outras dizem que o melhor caminho é a entrega de iterações.

Com a entrega de iterações o processo de desenvolvimento é dividido em pequenas partes, que são entregues ao longo do seu desenvolvimento, assim fornecendo ao usuário uma experiência do *software* antes mesmo de estar finalizado, em outras palavras entrega de valor de forma incremental.

Com o intuito de diminuir a curva de aprendizagem da equipe de desenvolvimento, é que se utilizam de algumas ferramentas como o UML e BPMN, pois com esses recursos, a capacidade de conhecimento sobre as necessidades do cliente se torna mais claras para todos da equipe.



VIDEOAULA 02

Utilize o QRcode para assistir!

Assista o vídeo para entender melhor o assunto.



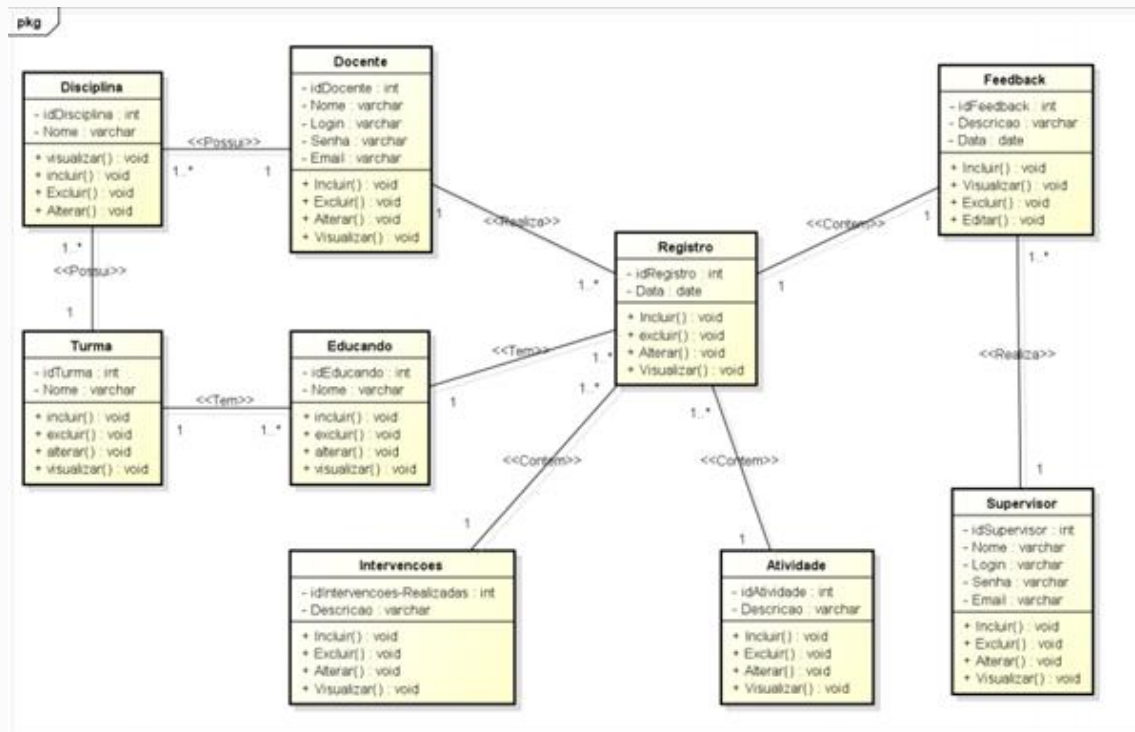
A UML possui atualmente 13 diagramas que podem ser utilizados para analisar e descrever como o *software* irá funcionar, e são divididos em três grandes grupos, sendo eles:

- a) **Diagramas de estrutura:** são os diagramas responsáveis por capturar a estrutura estática da aplicação;
- b) **Diagramas de comportamento:** são os diagramas responsáveis por modelar os tipos de comportamento do usuário com o sistema;
- c) **Diagramas de interação:** são derivados do diagrama de comportamento de uma forma mais geral.

A utilização correta dos diagramas resulta em um processo de desenvolvimento mais claro e eficaz entre todas as partes envolvidas, evitando conflitos ou até mesmo a entrega de um *software* que não atenda às necessidades do cliente.

Para facilitar a sua compreensão, vamos demonstrar nesta unidade alguns dos principais diagramas da UML utilizados no mercado de tecnologia da informação, são eles: os diagramas de classe, caso de uso e comunicação.

Figura 02 - Exemplo diagrama de classes



Fonte: o autor (2021)

Na Figura 02 temos a visualização do diagrama de classe, ele pertence ao grupo dos diagramas estruturais e demonstra de forma estrutural as classes que compõem um sistema. Uma classe representa um conjunto de objetos que possuem comportamentos (métodos/funções) e características (atributos). A classe é dividida em 3 partes: nome da classe (topo), atributos da classe (meio) e métodos da classe (inferior).

Figura 03 - Exemplo diagrama de caso de uso



Fonte: o autor (2021)

Na figura 03 podemos observar um diagrama de caso de uso, este diagrama faz parte do grupo de diagramas comportamentais. O diagrama de caso de uso é de grande auxílio para a compreensão dos requisitos do sistema, ajudando a especificar, visualizar e documentar as características e serviços desejados pelo usuário dentro do *software*. Por apresentar uma

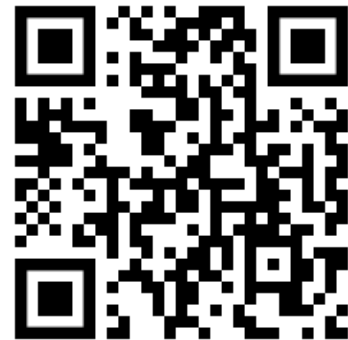
linguagem informal, bem como uma visão geral do comportamento do *software* esse diagrama pode ser apresentado durante as reuniões iniciais com os clientes como forma de ilustrar o comportamento do sistema, o seu comportamento é ilustrado dentro de uma elipse o que chamamos de caso de uso, que por sua vez interagem com um ator. Na Figura 03 esse ator é **chamado de corretor** e representa o papel do usuário que utiliza de alguma maneira os serviços e funções do sistema.



VIDEOAULA 03

Utilize o QRcode para assistir!

Agora, assista à videoaula que abordará esse assunto com mais amplitude.



Observando a figura 03, pode-se perceber que o ator **corretor** interage com o caso de uso **gerenciar_agenda**. De modo prático, a leitura deste caso de uso ocorre da seguinte maneira: o corretor gerencia uma agenda. Nota-se que em volta do caso de uso há um retângulo, é isso que chamamos de fronteira do sistema, tudo que está dentro desta fronteira faz parte do sistema, e o que está fora interage com o sistema de alguma forma.

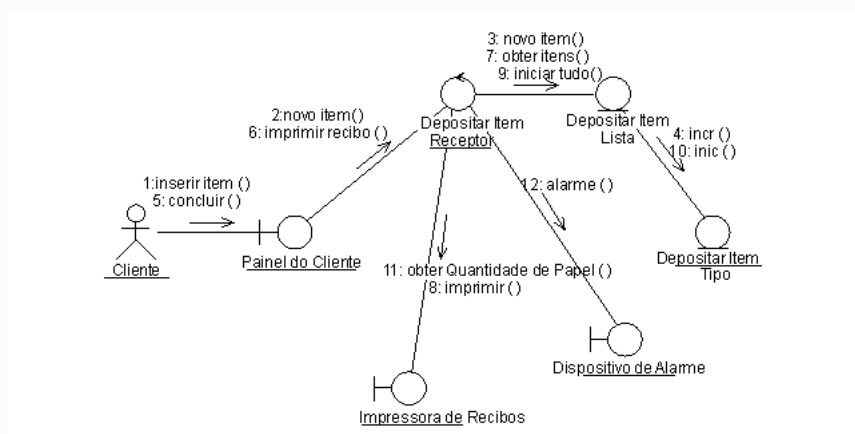
INDICAÇÃO DE VÍDEO

Assista ao vídeo abaixo para saber mais sobre a UML. Disponível em:

<https://www.youtube.com/watch?v=C3xYBT3o_5k>. Acesso em: 8 maio 2021.

Os casos de uso são fortemente utilizados para conseguir obter os requisitos do sistema, suas funcionalidades, aspectos de serviços, comportamento e como esses comportamentos interagem com os atores.

Figura 04 - Exemplo diagrama de comunicação



Fonte: IBM Corp (2006)

Na Figura 04 podemos observar o diagrama de comunicação, esse diagrama é classificado como sendo um diagrama de interação, ele descreve como o usuário/ator interage com o sistema e quais são as opções que ele encontra. Neste diagrama também está descrito a comunicação entre classes do sistema, sendo assim ele exibe um fluxo do funcionamento, dessa forma ele possui um ator e círculos que descrevem classes do *software*, as setas indicam os fluxos que o sistema pode percorrer, elas são numeradas e exibem as ações que podem ser realizadas em cada classe/etapa que o sistema se encontra.

A utilização de notações UML são recursos que podem ser utilizados durante todo o processo de desenvolvimento de software, pois torna visual os comportamentos dos sistemas e descreve as interações do *software* com o homem.

Vale ressaltar que todo o processo de *software* bem-sucedido concede ao usuário final uma mudança em sua vida, sendo assim podemos afirmar que um *software* bem desenvolvido impacta diretamente na vida do seu usuário.

Encerramento

Chegamos ao final da nossa primeira unidade, na qual você teve contato com os primeiros conceitos e ferramentas do processo de desenvolvimento. Com certeza, muito do que foi mostrado aqui você já tinha uma pequena noção. Espero que você aluno, tenha aprendido um pouco mais sobre todo o processo de desenvolvimento de *software*.

No mundo dos *softwares* há muitos modelos de desenvolvimento que podem ser melhores ou piores, no final tudo depende da demanda e do produto que deve ser entregue. Na aula 01, visualizamos alguns conceitos e como funciona o relacionamento de comunicação entre as partes de um projeto, bem como conhecemos um pouco sobre o BPMN.

Na aula 02, foi discutido e apresentado alguns recursos como a UML, que auxiliam na compreensão do projeto a ser desenvolvido pelo time, do mesmo modo as necessidades do cliente. Conhecemos também os tipos de diagramas, e como esse recurso pode ajudar no processo de desenvolvimento de *software*.

Juntando as duas aulas teremos um embasamento básico sobre processo de desenvolvimento de *software*. Caro aluno, pense um pouco como os conhecimentos desta unidade podem auxiliar você em processos do dia a dia. Até a próxima.

Bom trabalho!

Esperamos que este guia o tenha ajudado compreender a organização e o funcionamento de seu curso. Outras questões importantes relacionadas ao curso serão disponibilizadas pela coordenação.
Grande abraço e sucesso!

