Transcrição

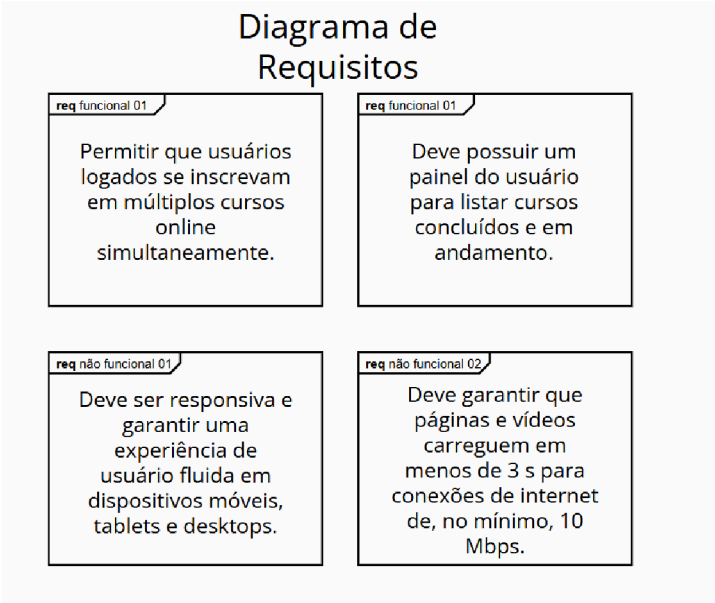
Estávamos analisando a plataforma da Alura, pensando em seu **ciclo de vida**, ainda no estágio de concepção, e mencionamos o uso de uma linguagem gráfica chamada **UML**. Quando iniciamos o processo de concepção de uma plataforma ou software em geral, definimos os **requisitos**. Esses requisitos nos ajudam a estabelecer as **regras de negócio**, que são todas as funcionalidades esperadas do software.

Analisando requisitos

Para modelar bem todo o sistema usando uma linguagem gráfica, o primeiro passo é definir os requisitos e as regras de negócio. Em geral, não possuímos dentro da UML um diagrama específico para elaborar os requisitos.

Por isso, utilizamos um diagrama de blocos para criar uma série de cartões para os requisitos, considerando o caso de uma

plataforma educacional. Dividimos os requisitos em duas categorias: funcionais e não-funcionais.



Os **requisitos funcionais** estão relacionados às funcionalidades que serão implementadas na plataforma. Já os **não-funcionais** estão mais ligados à experiência da pessoa usuária. Por exemplo, a plataforma deve ser responsiva e garantir uma experiência fluida em dispositivos móveis, tablets e desktops. Além disso, deve garantir um carregamento de menos de 3 segundos para uma determinada velocidade de conexão com a internet.

> Em todos as descrições desses requisitos, buscamos ser **concisos e claros**. Além disso, um requisito deve ser **testável ou verificável**.

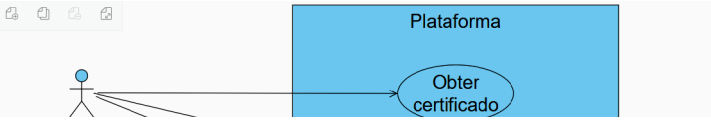
Em outras palavras, devemos conseguir entrar na plataforma e testar se a funcionalidade foi implementada e se o requisito não-funcional está sendo atendido. Um conjunto de requisitos ou regras de negócio opera como um **contrato** entre nós, pessoas desenvolvedoras, e *stakeholders*, que são as partes interessadas no software. Cada requisito é como uma cláusula que devemos cumprir.

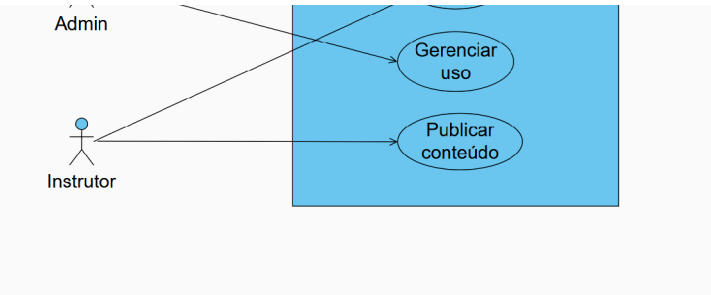
No processo de desenvolvimento, os requisitos ou regras que definimos são nossa bússola para guiar todo o processo de desenvolvimento. A partir deles, podemos construir diagramas da UML para discutir funcionalidades e também as camadas e componentes da plataforma.

Conhecendo diagrama de casos de uso

Vamos começar pelo **diagrama de casos de uso**. Este diagrama é comum na descrição de um software e descreve alguns casos de uso, ou seja, **como a plataforma será utilizada** pelos seus atores.

Vamos conferir um exemplo simples de diagrama de caso de uso da mesma plataforma de cursos online:





Os atores são as **pessoas usuárias**, com **diferentes perfis**. Nessa plataforma, temos a figura de **admin**, da pessoa instrutora e da pessoa estudante. Essas diferentes pessoas realizam tarefas distintas.

Dentro do retângulo à direita onde descrevemos as funcionalidades a que cada ator terá acesso na plataforma, utilizamos uma **linguagem concisa e clara**. Por exemplo, a pessoa estudante poderá obter o certificado, realizar o curso e fazer o login. Todos poderão fazer o login. Além disso, o admin poderá gerenciar o uso da plataforma, enquanto a pessoa instrutora poderá publicar conteúdo.

Para criar um bom diagrama de casos de uso, com detalhes e aspectos do funcionamento de cada caso, incluindo funcionalidades internas, precisamos consultar a documentação oficial da UML para entender como usar os **recursos gráficos** disponíveis. No caso da ferramenta

Visual Paradigm, esses recursos ficam listados no painel lateral esquerdo.

> Cada recurso gráfico tem um significado específico. Por isso, precisamos conhecê-los para poder montar cada diagrama da maneira adequada.

É importante destacar que os requisitos são nossa bússola no processo de desenvolvimento e podem ser entendidos como **cláusulas de um contrato**. No entanto, ao longo da vida de uma plataforma, esses requisitos invariavelmente passarão por **modificações**. Por exemplo, na Alura, agora temos um chatbot inteligente para ajudar na trajetória de aprendizagem. Antes, há dois ou quatro anos, isso não era um requisito da plataforma. Hoje, já é.

> À medida que a tecnologia evolui, precisamos **atender a novos requisitos**, pois estão intimamente relacionados às expectativas das pessoas usuárias e ao dinamismo do mercado em que a solução está inserida

Por isso, é fundamental ter um processo contínuo de controle e discussão dos requisitos, entendendo que eles mudam. A documentação do software é essencial para

entender em que ponto precisamos implementar uma nova funcionalidade ou modificação.

Próximos passos

No diagrama de caso de uso, obtemos uma visão geral dos principais componentes inseridos na solução. Tanto as pessoas desenvolvedoras, quanto as partes interessadas terão uma ideia geral da estrutura da plataforma. A partir daí, podemos decidir sobre a arquitetura, estrutura e componentes a serem utilizados.

Além disso, temos outros diagramas, como o **diagrama de classes**. Vamos explorar mais essa diferença entre os diagramas para entender como podemos usar cada um para representar nossa solução.