



Transcrição

Quando pensamos em software, geralmente a primeira ideia que nos vem à mente é uma série de linhas de código e módulos nos quais implementamos funcionalidades de uma aplicação.

De fato, desenvolver um software envolve muita programação, mas devemos lembrar que esse software não existirá apenas para nós, pessoas desenvolvedoras. Ele estará inserido em um contexto, podendo ser em uma farmácia, padaria, indústria ou até mesmo no smartphone das **pessoas usuárias**. Atualmente, interagimos constantemente com aplicativos móveis, que também são exemplos de software.

Software é uma solução para as pessoas em geral!

É comum pensar que o desenvolvimento de software está altamente relacionado ao campo das ciências exatas. De fato,

utilizamos uma série de conhecimentos e

software e áreas afins. No entanto, o processo de construir um software não é essencialmente determinístico.

Não conseguimos, *a priori*, inserir uma série de funcionalidades esperadas e aplicar uma teoria matemática, fórmula ou modelo que nos permita obter o software perfeito para aquelas funcionalidades que pretendemos implementar. Portanto, é um processo que envolve **criatividade e** habilidade da pessoa programadora, utilizando uma linguagem, um *framework* ou um conjunto de ferramentas nesse processo de desenvolvimento.

Como lidamos com esse processo de desenvolvimento de um software?

Compreendendo o ciclo de vida

Imagine que recebemos uma demanda para desenvolver um software ERP (*Enterprise Resource Planning* ou Planejamento de Recursos Empresariais, em português) para um mercado ou pequeno negócio no nosso bairro. Como lidamos com esse processo? Começamos implementando diretamente? Não. Vamos considerar um caso concreto

próximo a nós, que é a plataforma da Alura, onde estamos fazendo este curso.

Na plataforma, temos uma série de funcionalidades, incluindo a inteligência artificial da Luri, que nos permite interagir e obter conteúdos extras, além dos artigos, vídeos, cursos e formações disponíveis.

Ao longo dos mais de 10 anos de existência da plataforma no Brasil, começamos com funcionalidades mais básicas, assim como as demandas de conteúdo, que eram mais voltadas para o desenvolvimento web e programação em Java e C, linguagens mais comuns na época do lançamento da plataforma. Atualmente, temos uma variedade de conteúdos diversificados para todos os públicos, o que possibilita uma formação continuada.

Quando pensamos na trajetória de uma plataforma como a Alura ou de um aplicativo móvel que precisa ser atualizado, seja por mudanças no Android ou por vulnerabilidades detectadas, consideramos um **ciclo de vida da aplicação**. Esse ciclo de vida começa na concepção da plataforma, ou seja, nessa solução que vamos construir em forma de software.

O **ciclo de vida** é composto por uma série de estágios pelos quais uma

aplicação passará ao longo de sua vida útil.

O primeiro estágio é a **concepção**, onde não estamos programando de fato, mas, sim, **desenhando a solução**.

Devemos pensar no software sempre como uma solução para outras pessoas. Por isso, no processo de concepção, devemos interagir bastante com as pessoas envolvidas, ou seja, as **partes interessadas**, conhecidas na linguagem de negócios e gestão como **stakeholders**. Muitas vezes, não conversamos diretamente com as pessoas usuárias, mas sim com *stakeholders* e parceiros, para entender o que buscam em termos de solução de software. A partir dessa conversa, definimos os **requisitos**.

O que é um requisito? Vamos pensar no caso da Alura. Ao interagir com a plataforma, podemos nos inscrever em cursos para criar uma primeira aplicação tanto com C# quanto com Python. É possível fazer os cursos dessas duas linguagens de maneira simultânea. Ao final de cada curso, podemos obter o certificado, compartilhar e incluir em nossa pasta de conquistas profissionais.

Tudo isso são requisitos da plataforma. Um dos requisitos é permitir que uma pessoa se

inscreva simultaneamente em diferentes cursos, sem limitar a jornada de aprendizagem. Quando a pessoa conclui um curso, há uma funcionalidade para gerar um certificado que ficará permanentemente disponível em seu perfil.

Além disso, temos requisitos relacionados à interação da pessoa usuária com a plataforma, como **usabilidade** e **navegabilidade**. Por exemplo, ao clicar para postar uma dúvida no fórum, esperamos uma resposta rápida da plataforma. Ninguém quer esperar mais de 10 segundos para carregar uma página. Esse tipo de parâmetro, relacionado ao tempo de resposta e latência da plataforma, é um requisito operacional importante para garantir uma **boa experiência de uso** na Alura ou em qualquer outra aplicação web.

Os requisitos são levantados no início do processo de concepção da plataforma. A partir desses requisitos, que foram definidos em conjunto com as partes interessadas, construímos uma série de **diagramas descritivos**.

Devemos construir diagramas para descrever a solução de forma gráfica. No entanto, em vez de modelar essa solução por meio de desenhos livres que apenas nós compreenderíamos, usamos uma

linguagem unificada que permite que diferentes pessoas desenvolvedoras e parceiras compreendam o que estamos desenhando.

Essa linguagem gráfica padrão é chamada de **UML** (*Unified Modeling Language* ou Linguagem Unificada de Modelagem).

A UML é bem conhecida, começou a ser adotada no final dos anos 90 e é utilizada até hoje. Existem diversas documentações em vários sites, como o da IBM, que orientam sobre como criar diagramas de classe (<https://www.ibm.com/docs/pt-br/dmrt/9.5?topic=diagrams-creating-class>) e diagramas de casos de uso, descrevendo as principais formas de uso da plataforma ou solução que vamos construir.

Além disso, existem uma série de editores de diagramas online, que permitem construir diagramas de forma simples e rápida - sem precisar instalar nada no computador. Basta procurar por "UML ferramentas" em um motor de pesquisa, como o Google.

Próximos passos

A UML é uma linguagem interessante no processo de modelagem para conceber e discutir como a solução em forma de software será implementada. Vamos explorar mais sobre o mundo da modelagem de aplicações na sequência.