Arquitetura de uma aplicação em Angular

GeneXus™

Angular é um framework para construir aplicações web, vamos começar revisando alguns conceitos destas aplicações.

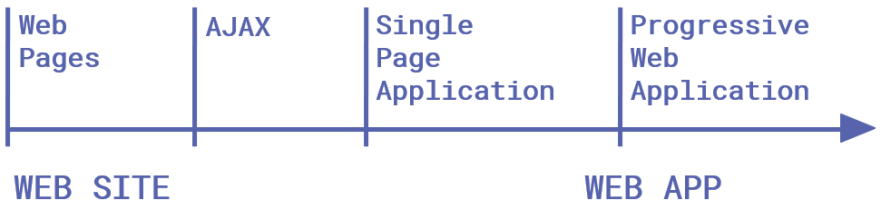
Arquitetura de uma aplicação web

As aplicações web têm dois componentes, o front-end que é executado no navegador web e o back-end que é executado em um servidor web.

O front-end é onde é executada realmente a aplicação e é a parte com a qual o usuário interage, aquela que possui a interface de usuário, é mantido o estado da aplicação e é tratada a lógica de negócios.

O back-end é onde estão os serviços que atendem ao front-end, é validada a lógica de negócios e são obtidos os dados da base de dados.

Evolução das aplicações web



As aplicações web evoluíram desde sua primeira implementação, passando de páginas quase estáticas a aplicações com uma grande riqueza de conteúdo e interatividade.

Para aumentar a interatividade com os controles da página, surgiu a tecnologia AJAX que permitia que algumas coisas fossem resolvidas no cliente sem a necessidade de a solicitação viajar para o servidor.

Em seguida, surgiram as aplicações de página única (SPA). Estas aplicações têm a particularidade de que, em resposta às interações do usuário, a página não é carregada novamente, mas é atualizada apenas a parte necessária, o que proporciona uma experiência de usuário mais fluida.

E mais recentemente surgiram as aplicações web progressivas (PWA), que são aplicações web que permitem o acesso aos recursos de hardware e software da máquina como se fosse uma aplicação nativa e também permitem a sua instalação no desktop ou em dispositivos móveis.

O que é Angular?

E o que é Angular?

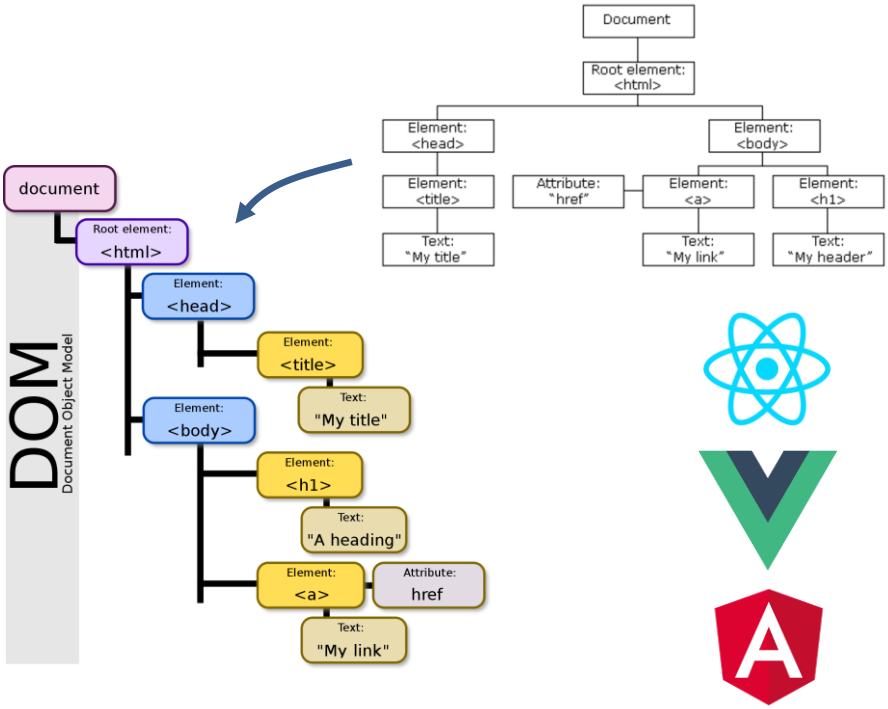
Angular é um framework de desenvolvimento para JavaScript de código aberto criado por Google, que serve para gerar aplicações web de front-end, em particular aplicações de página única e aplicações web progressivas.

Angular é programado com TypeScript, que é um superset de JavaScript que adiciona tipos de dados, entre outras coisas, portanto, finalmente é JavaScript que é executado no navegador.

Este framework se especializa em melhorar a velocidade de desenho da página web no browser, ou seja, a renderização, tornando-o ideal para aplicações de front-end orientadas ao usuário final, as aplicações denominadas “customer-facing”.

DOM: Document Object Model

```
1  <html>
2      <head>
3          <title>"My title"</title>
4      </head>
5      <body>
6          <h1>"A heading"</h1>
7          <a href="My link"></a>
8      </body>
9  </html>
```

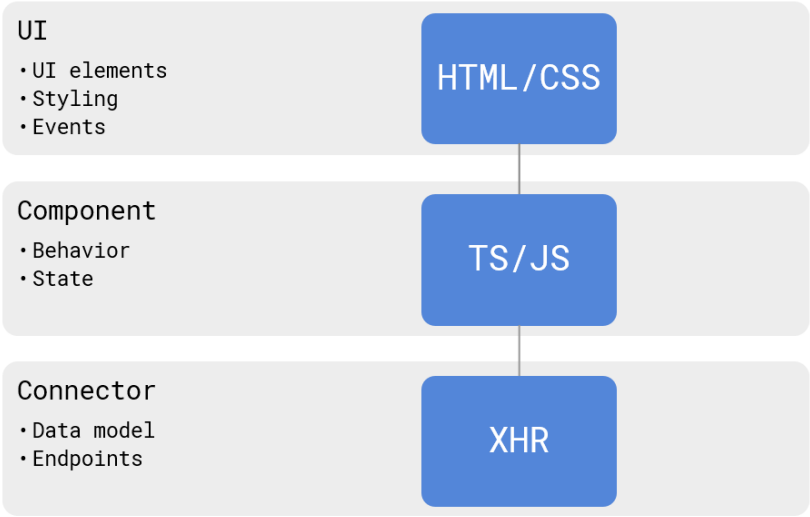


Para conseguir que a tela seja pintada rapidamente, Angular realiza um tratamento eficiente do DOM (Document Object Model) que é a estrutura hierárquica de objetos que gera o navegador quando é carregado um documento e é possível alterar usando Javascript para modificar dinamicamente os conteúdos e aparência da página.

Como as aplicações de página única e as PWA são pesadas e cada vez mais complexas, o tratamento eficiente do DOM é muito importante. Isto se consegue mediante a virtualização do DOM e, em seguida, atualiza o DOM real de forma muito rápida. Existem outros frameworks para front-end web como React ou VueJS que também resolvem o tratamento eficiente do DOM, mas cobrem apenas uma parte do desenvolvimento.

Angular, por outro lado, é uma solução completa que, além da lógica para o gerenciamento do DOM, inclui o gerenciamento de componentes, bibliotecas para navegar a aplicação, gerenciamento do estado dos controles em tela e mecanismos de comunicação entre o cliente e o servidor.

Estrutura de uma aplicação de front-end



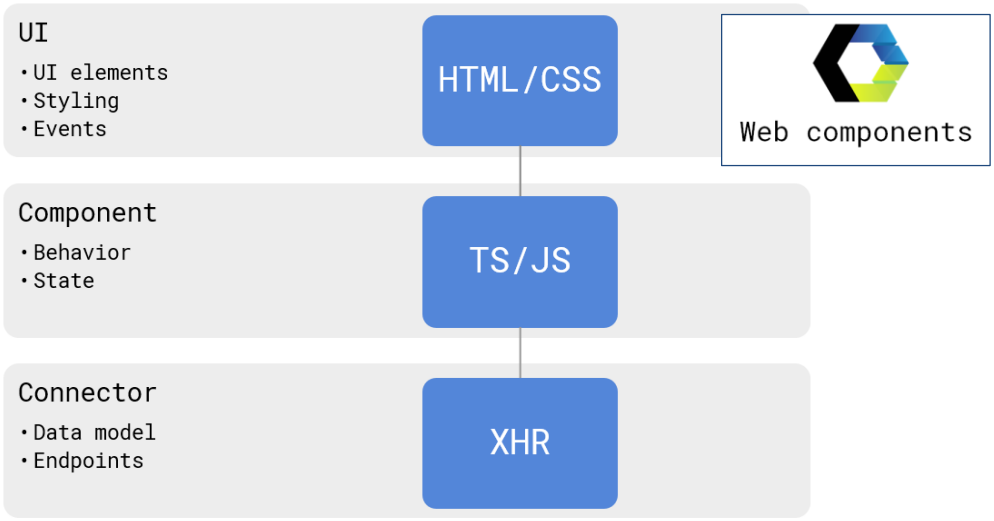
Uma aplicação web de front-end possui as seguintes 3 camadas ou elementos.

A User Interface, que é o que o usuário vê. São os elementos de tela ou controles codificados em html, com as definições de aparência escritas em css (styling). Quando o usuário interage com esta interface de usuário, são gerados eventos que causam alterações no conteúdo da página.

Component - É o código que reage aos eventos dos elementos da User Interface, ou seja, onde é programado o comportamento dos controles da UI. Esta camada também se encarrega de manter o estado de cada controle e seu código é programado em TypeScript ou em JavaScript.

Connector - É o representante do servidor no cliente, o que ele faz é pedir ao servidor dados e em alguns casos pode chegar a persisti-los, para minimizar as comunicações com o servidor.

Como é construído o front-end com o gerador Angular de GeneXus



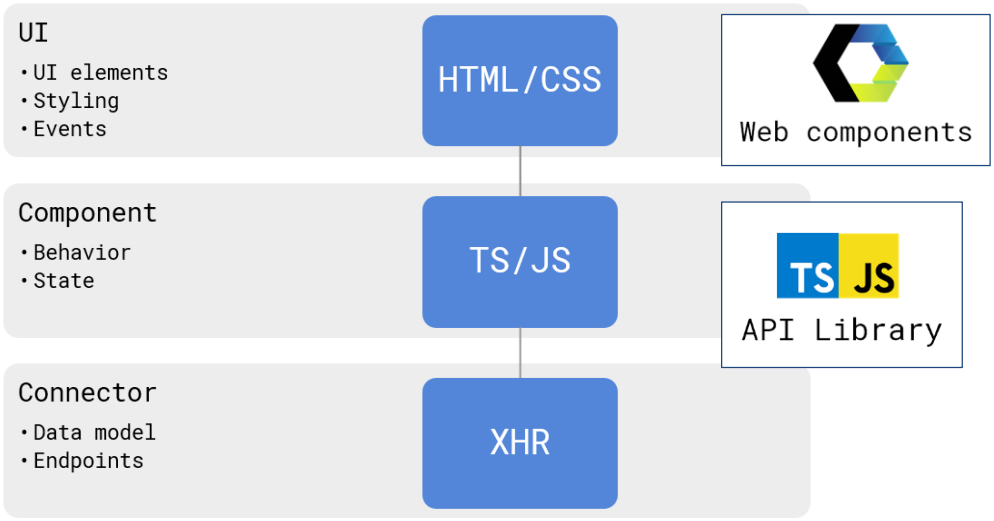
Vejamos agora como o gerador Angular de GeneXus constrói cada uma destas partes.

Para gerar a User Interface são utilizados Web Components. Os web components são um padrão de W3C que nos permite encapsular html, javascript e css em tags html definidas por nós.

Em outras palavras, podemos estender o html e criar nossas próprias tags com base em elementos que rodam nativamente em todos os browsers. Por exemplo, podemos criar um botão nosso com seu estilo e seu comportamento, que fica encapsulado em um web component e então podemos usá-lo em outros projetos web ou distribuí-lo para outros usarem.

É uma nova maneira de componentizar na web e tornar os componentes portáteis.

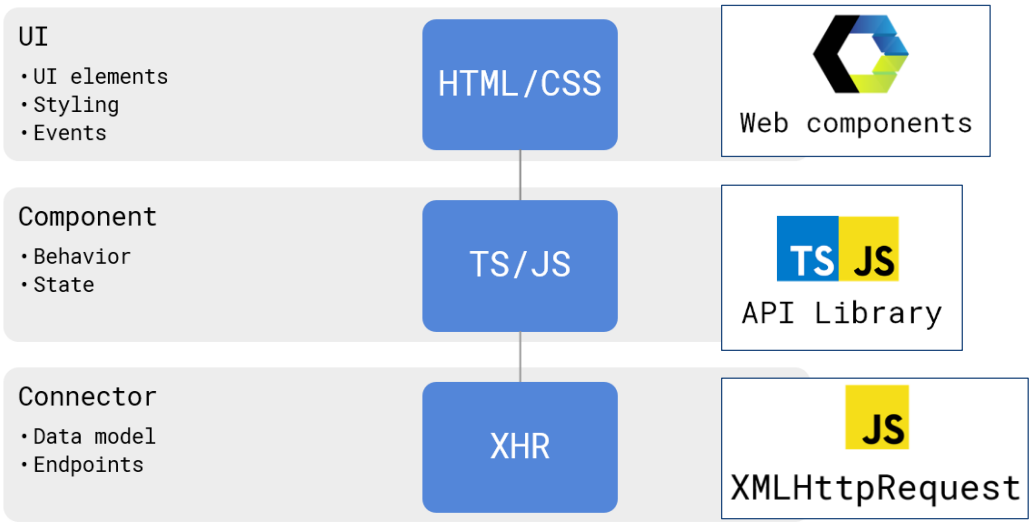
Como é construído o front-end com o gerador Angular de GeneXus



Para implementar a camada de comportamento, utiliza-se a GeneXus API Library.

Estas APIs são funções de uso geral, como as de manipulação de caracteres, funções de data, etc., e que eram geradas no código específico de cada gerador, agora foram implementadas em TypeScript e JavaScript para poderem ser reutilizadas em todos os geradores, entre eles Angular.

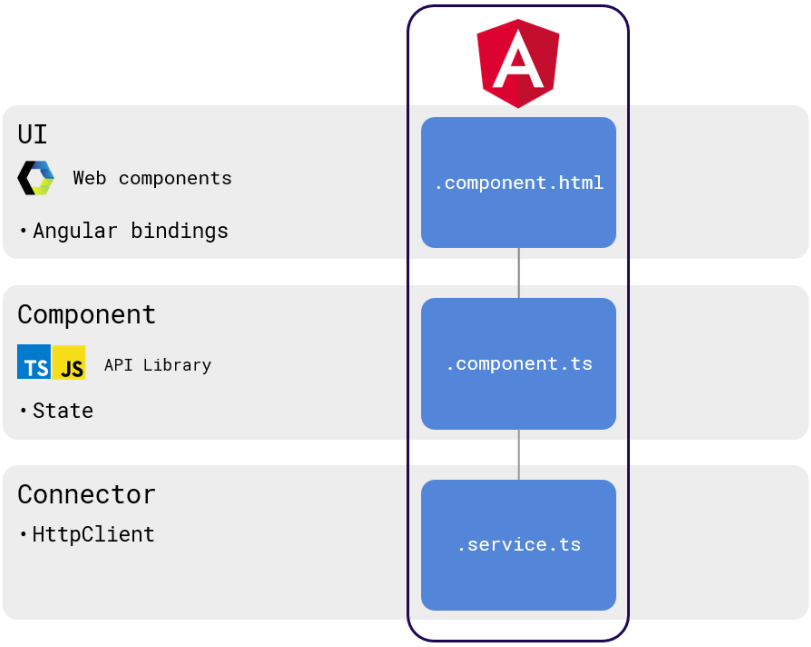
Como é construído o front-end com o gerador Angular de GeneXus



Para o conector é utilizado o objeto XMLHttpRequest, que é um objeto JavaScript projetado pela Microsoft e adotado por Mozilla, Apple e Google.
Atualmente é um padrão W3C e permite:

- Atualizar uma página web sem recarregar a página
- Solicitar dados ao servidor e receber dados do servidor, depois de ter sido carregada a página
- Enviar dados ao servidor, em segundo plano

Componente angular da aplicación e arquivos gerados



O bloco de construção básico de uma aplicação Angular é denominado Componente. Cada componente consiste em 3 partes que mapeiam as camadas de uma aplicação front-end e são representadas em 3 arquivos que são criados pelo gerador Angular.

As definições de user interface são armazenadas em um arquivo cujo nome termina com .component.html. Este arquivo é gerado por GeneXus e contém os web components que definem o template html e as definições do estilo da User Interface. Também são incluídas as associações dos elementos de UI com os dados correspondentes da base de dados.

O componente que define o comportamento e estado dos controles é armazenado em um arquivo que termina com o nome .component.ts. Ali estão programados o comportamento e o estado dos controles em TypeScript, utilizando a GeneXus API Library.

O conector que se encarrega de comunicar-se com os serviços REST do servidor web é gerado em um arquivo que termina com o nome .service.ts e GeneXus usa o HttpClient para implementar estas comunicações, utilizando o XMLHttpRequest.

Para cada objeto panel que for gerado, serão criados estes 3 arquivos cujos nomes serão compostos pelo nome do objeto que será executado, seguido das terminações que estão em tela.

Cenários de uso do gerador Angular

- Desenvolvimento de uma aplicação front-end web do tipo Customer-facing
- Quando sabemos que vamos precisar de uma aplicação móvel nativa
- Converter aplicações móveis nativas desenvolvidas em GeneXus em uma Web app
- Desenvolver back-end em GeneXus para front-end desenvolvidos com React, VueJs ou Angular

Uma pergunta que podemos nos fazer é, quando devemos escolher o Angular para gerar nossa aplicação web.

Um cenário possível é quando precisamos desenvolver uma aplicação de front-end web voltado para o cliente final (customer-facing). Como vimos, Angular foi projetado para ter um bom desempenho nestes casos de interfaces de usuário com requisitos de muita interatividade.

Outro caso em que o desenvolvimento em Angular é indicado, é quando temos a certeza de que após o front-end web, precisaremos de um front-end mobile nativo. Neste caso, os painéis que criamos para a aplicação Angular poderão ser reutilizados para gerar a aplicação Android ou iOS, definindo os layouts correspondentes aos dispositivos que vamos utilizar.

Também é aplicável o uso de Angular quando já desenvolvemos uma aplicação nativa para Android ou iOS com GeneXus e precisamos de um front-end web. Este caso é semelhante ao anterior, pois podem ser utilizados os mesmos objetos, mudando o desenho do layout para web e então gerando a mesma aplicação que tínhamos em Angular.

Mais um caso em que é útil o uso de Angular como gerador é quando queremos integrar aplicações codificadas manualmente em React, VueJs ou Angular, com serviços ou aplicações desenvolvidas em GeneXus. Isto permite que se você tiver uma equipe de desenvolvedores especializados no uso destes frameworks, eles possam continuar construindo aplicações em sua ferramenta, mas toda a parte do back-end pode ser desenvolvida em GeneXus e integrá-la ao front-end.

Nos vídeos a seguir veremos como podemos desenvolver aplicações web com o gerador Angular em GeneXus.

training.genexus.com
wiki.genexus.com
training.genexus.com/certifications