**Deferrable Views**

também conhecidos como **@defer blocks**, são uma nova funcionalidade introduzida no Angular 17 que permite **adiar o carregamento** de determinadas partes de uma aplicação.

Isso pode ser usado para melhorar o desempenho das aplicações, especialmente aquelas com grandes quantidades de conteúdo.

**Exemplos**

Aqui estão alguns exemplos de como usar **Deferrable Views** para melhorar o desempenho das aplicações:

* **Carregamento de imagens:** Você pode usar **Deferrable Views** para adiar o carregamento de imagens até que elas sejam visíveis na tela. Isso pode ajudar a melhorar o desempenho das aplicações móveis, onde a largura de banda é limitada.
* **Carregamento de dados:** Você pode usar **Deferrable Views** para adiar o carregamento de dados até que eles sejam necessários. Isso pode ajudar a melhorar o desempenho das aplicações que precisam carregar grandes quantidades de dados.
* **Carregamento de componentes pesados:** Você pode usar **Deferrable Views** para adiar o carregamento de componentes pesados até que eles sejam necessários. Isso pode ajudar a melhorar o desempenho das aplicações que usam componentes complexos ou personalizados.

**Considerações**

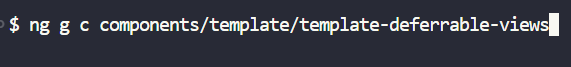
Ao usar **Deferrable Views**, é importante considerar os seguintes fatores:

* **Eficiência:** Os **Deferrable Views** podem melhorar o desempenho das aplicações, mas eles também podem adicionar complexidade e aumentar o tempo de carregamento da primeira página.
* **Usuabilidade:** Os **Deferrable Views** podem afetar a usabilidade das aplicações, pois o conteúdo pode não estar disponível imediatamente.

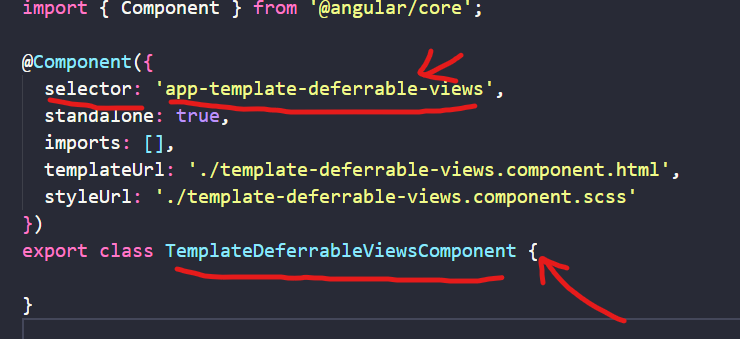
Os **Deferrable Views** são uma nova funcionalidade poderosa que pode ser usada para melhorar o desempenho das aplicações Angular. No entanto, é importante considerar os fatores acima antes de usá-los.

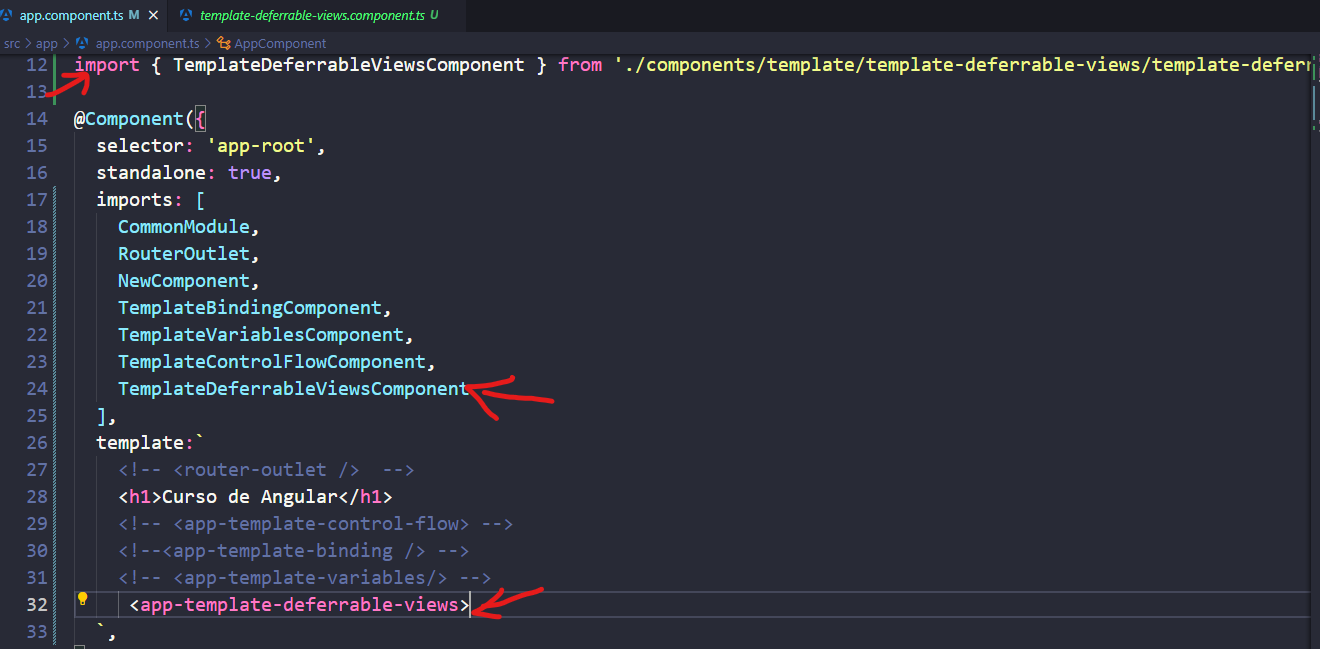
**COMO USAR NO ANGULAR**

Comandono GIT-BASH para criar um novo componente: template-deferrable-views:

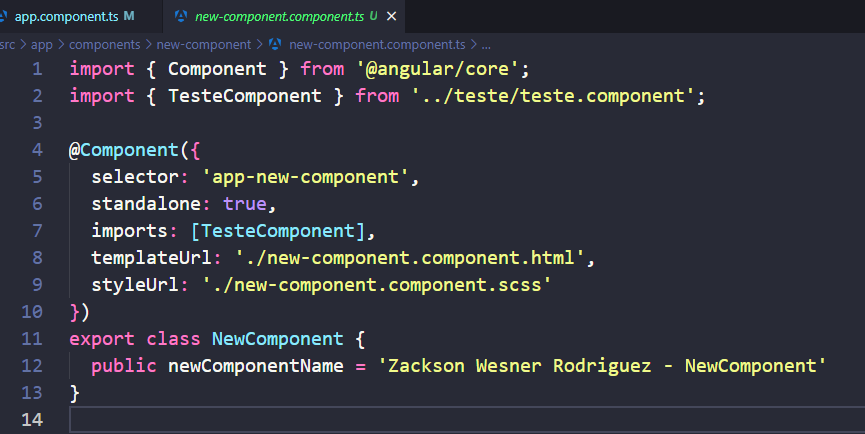


Este novo componente deve ser importado no arquivo “app.components.ts” e também deve ser informado o SELETOR:





Agora vamos importar outro componente para este novo componente recém criado:





Agora o código é no HTML

