

Code Dimension



# OnPush Strategy

# Code Dimension



## O que é?

- Permite uma verificação de estado mais certa
- Usa a **referência** das inputs (Arrays e Objetos) para comparar os estados
- Notifica o Angular quando eventos no DOM acontecem ou signals são modificados

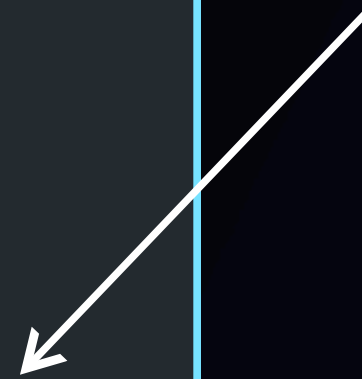
# Code Dimension



## Como usar?

### OnPush Strategy

```
@Component({  
  selector: 'app-root',  
  imports: [Add, List],  
  templateUrl: './app.html',  
  styleUrls: ['./app.css'],  
  changeDetection: ChangeDetectionStrategy.OnPush  
})  
export class App {
```



# Code Dimension



## Quando usar?

- Aplicações com muitos componentes sendo renderizados ao mesmo tempo
- Para garantir o uso correto de Signals



# Code Dimension



## Vantagens

- Ciclos de renderização mais otimizados
- Menor consumo de memória
- Melhor integração com Signals

# Code Dimension



## Desvantagens

- Pequena curva de aprendizado
- Necessita mais da intervenção dos Devs para gerar novas renderizações (se não usar Signals)
- Precisa conhecer Imutabilidade