

## 44 - Vistas diferidas : @defer (when [expresión lógica]) {}

Otra variante que podemos implementar con las vistas diferidas es utilizando la palabra clave when con una expresión lógica que retorne true o false. La vista se materializará cuando la expresión lógica se verifique verdadera.

Creemos un proyecto y veamos el funcionamiento de esta variante de

vista diferida:

```
ng new proyecto030
```

En el archivo 'app.component.ts' definimos un atributo que almacene el valor false:

```
import { Component } from '@angular/core';
import { RouterOutlet } from '@angular/router';

@Component({
  selector: 'app-root',
  imports: [RouterOutlet],
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  visible = false;
}
```

La la plantilla 'app.component.html' queda con el siguiente código:

```
<p>¿Quién descubrió America?</p>
<input type="button" (click)="visible=true" value="mostrar respuesta"
>
@defer (when visible) {
  <p>Respuesta:Cristóbal Colón</p>
} @placeholder {
  <p>¿?</p>
}
<router-outlet />
```

Como el atributo visible almacena el valor false, el bloque de código no se renderiza. Luego cuando presionamos el botón cambiamos el valor del atributo visible por el valor true, en dicho momento se hace visible el bloque de código definido en el @defer.

Tener en cuenta que en una aplicación más realista en el bloque @defer tendremos una componente, que se cargará con lazy loading una vez que el when se verifique con el valor true.

Podemos probar esta aplicación en la web [aquí](#).

Podemos utilizar también el prefetch, para el caso en que sea una componente su contenido se recupere del servidor inmediatamente y cuando el when se verifique verdadero, la componente se renderize inmediatamente:

```
@defer (when visible;prefetch on idle) {  
<p>Respuesta:Cristóbal Colón</p>  
} @placeholder {  
<p>¿?</p>  
}
```

## Retornar