

Crear componentes anidados



Uso de componentes

Como directiva



Componente de aplicación o componente anidado

Como objetivo de enrutamiento

Product List

Filter by:

Show Image

Product

Code

Available

Price

5 Star Rating

Leaf Rake

gdn 0011

March 19, 2021

\$19.95

★★★★

Garden Cart

gdn 0023

March 18, 2021

\$32.99

★★★★

Hammer

tbx 0048

May 21, 2021

\$8.90

★★★★★

Saw

tbx 0022

May 15, 2021

\$11.55

★★★★

Video Game Controller

gmg 0042

October 15, 2020

\$35.95

★★★★★

Vista de página completa

```
<body>
  <pm-root></pm-root>
</body>
```


¿Qué hace que un componente se pueda anidar?



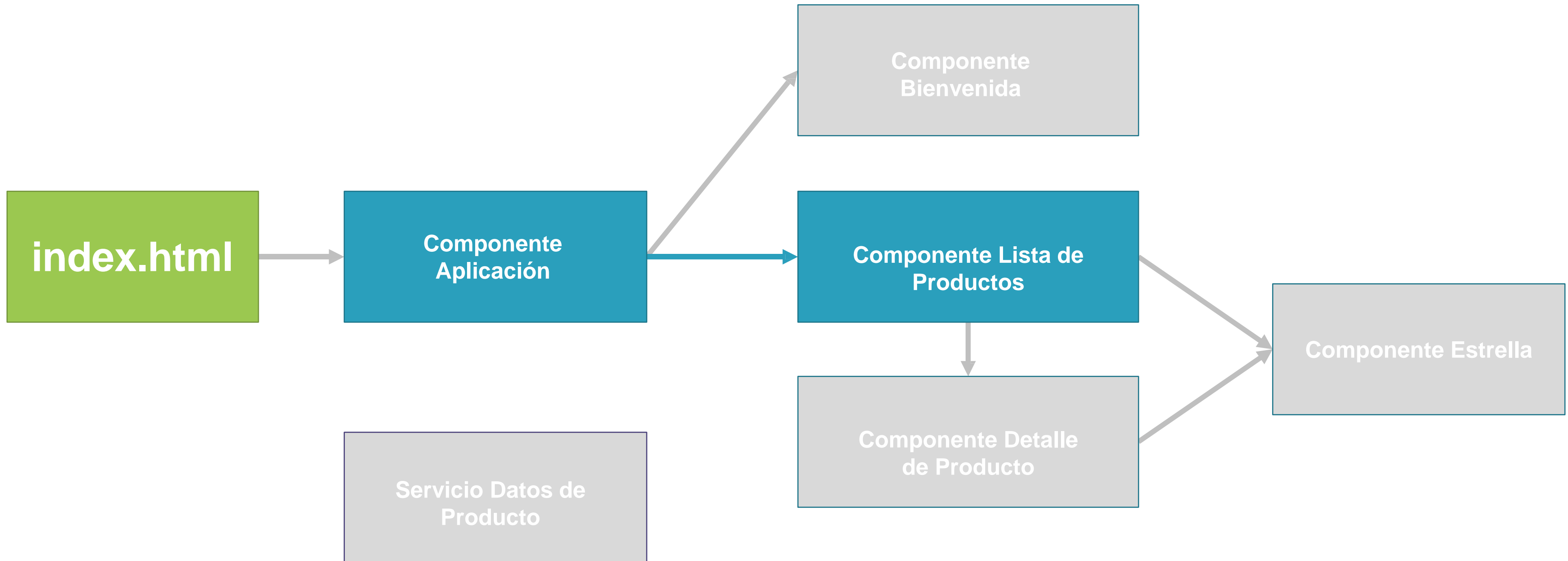
- Su plantilla administra un fragmento de una vista mayor
- Tiene un selector
- Se comunica opcionalmente con su contenedor

Resumen

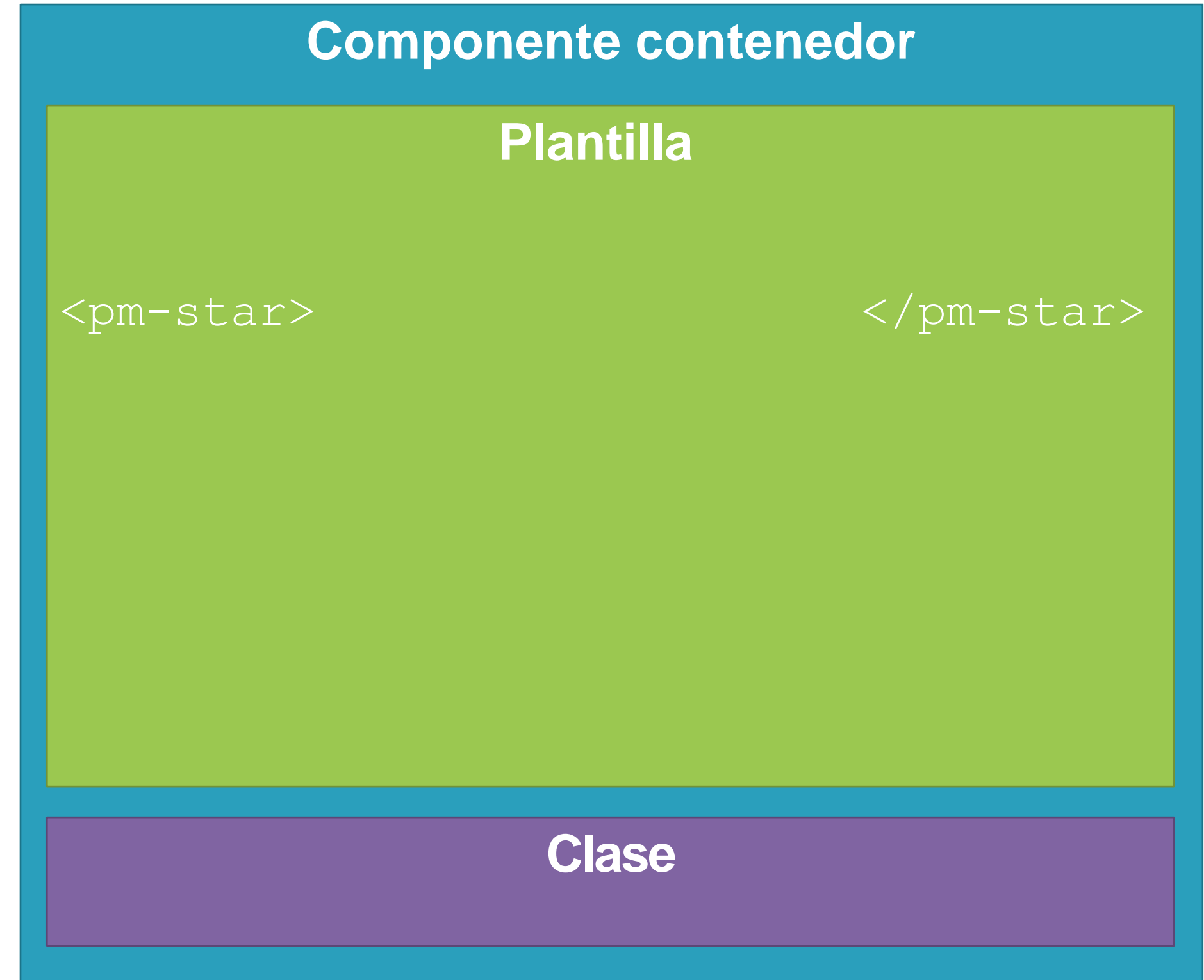


- **Crear un componente anidado**
- **Uso de un componente anidado**
- **Pasar datos a un componente anidado mediante `@Input`**
- **Cómo lanzar un evento desde un componente anidado mediante `@Output`**

Arquitectura de la aplicación



Crear un componente anidado



Vista lista de productos

Product List					
Filter by:	<input type="text"/>				
Show Image	Product	Code	Available	Price	5 Star Rating
	Leaf Rake	gdn 0011	March 19, 2021	\$19.95	3.2
	Garden Cart	gdn 0023	March 18, 2021	\$32.99	4.2
	Hammer	tbx 0048	May 21, 2021	\$8.90	4.8
	Saw	tbx 0022	May 15, 2021	\$11.55	3.7
	Video Game Controller	gmg 0042	October 15, 2020	\$35.95	4.6

Vista lista de productos

Product List					
Filter by:	<input type="text"/>				
Show Image	Product	Code	Available	Price	5 Star Rating
	Leaf Rake	gdn 0011	March 19, 2021	\$19.95	★★★★
	Garden Cart	gdn 0023	March 18, 2021	\$32.99	★★★★★
	Hammer	tbx 0048	May 21, 2021	\$8.90	★★★★★
	Saw	tbx 0022	May 15, 2021	\$11.55	★★★★
	Video Game Controller	gmg 0042	October 15, 2020	\$35.95	★★★★★

Uso de un componente anidado como directiva

product-list.component.ts

```
@Component ({  
  selector: 'pm-products',  
  templateUrl: './product-list.component.html'  
})  
export class ProductListComponent { }
```

product-list.component.html

```
<td>  
  {{ product.starRating }}  
</td>
```

star.component.ts

```
@Component ({  
  selector: 'pm-star',  
  templateUrl: './star.component.html'  
})  
export class StarComponent {  
  rating: number;  
  cropWidth: number;  
}
```

Uso de un componente anidado como directiva

product-list.component.ts

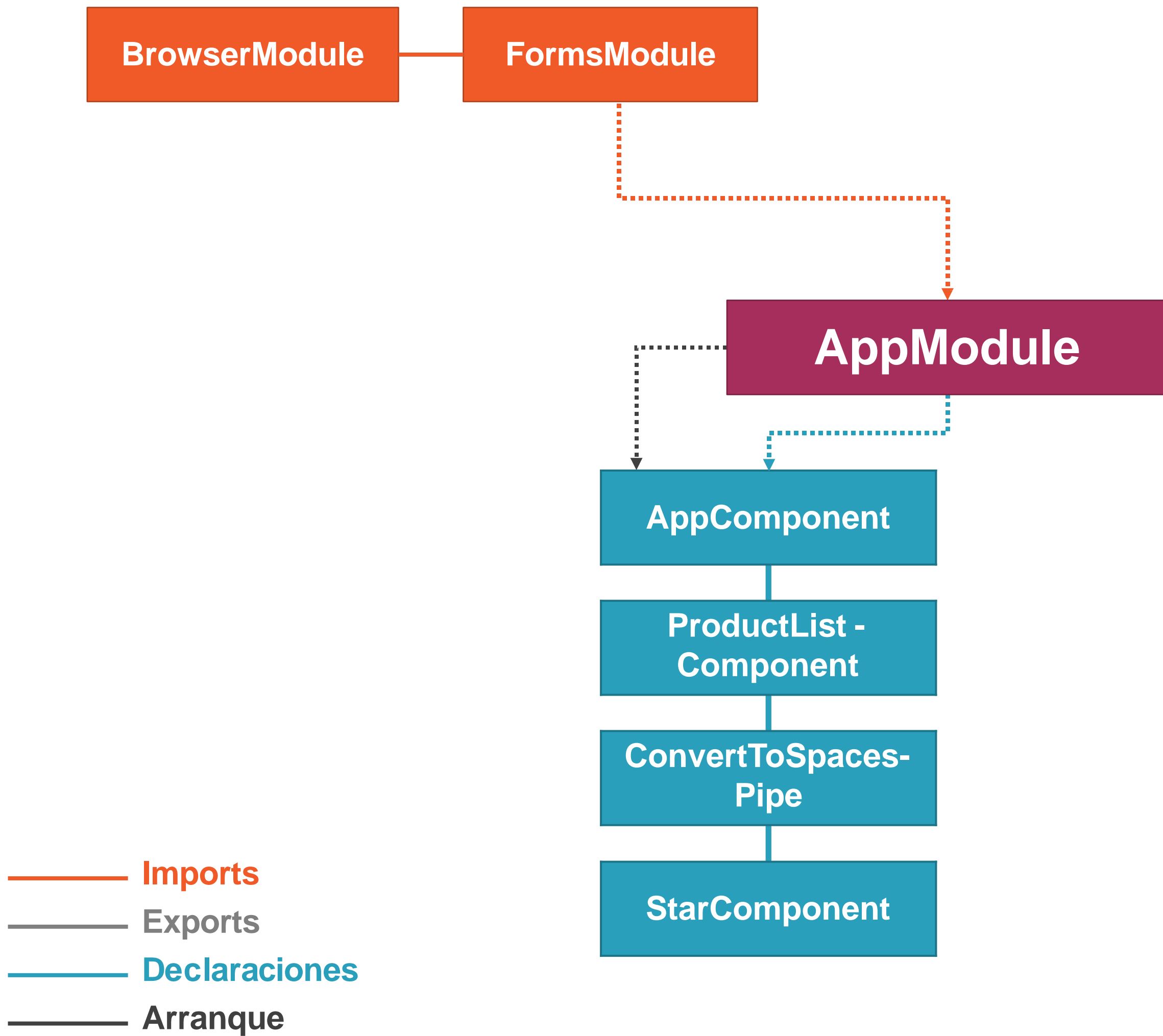
```
@Component ({  
  selector: 'pm-products',  
  templateUrl: './product-list.component.html'  
})  
export class ProductListComponent { }
```

product-list.component.html

```
<td>  
  <pm-star></pm-star>  
</td>
```

star.component.ts

```
@Component ({  
  selector: 'pm-star',  
  templateUrl: './star.component.html'  
})  
export class StarComponent {  
  rating: number;  
  cropWidth: number;  
}
```



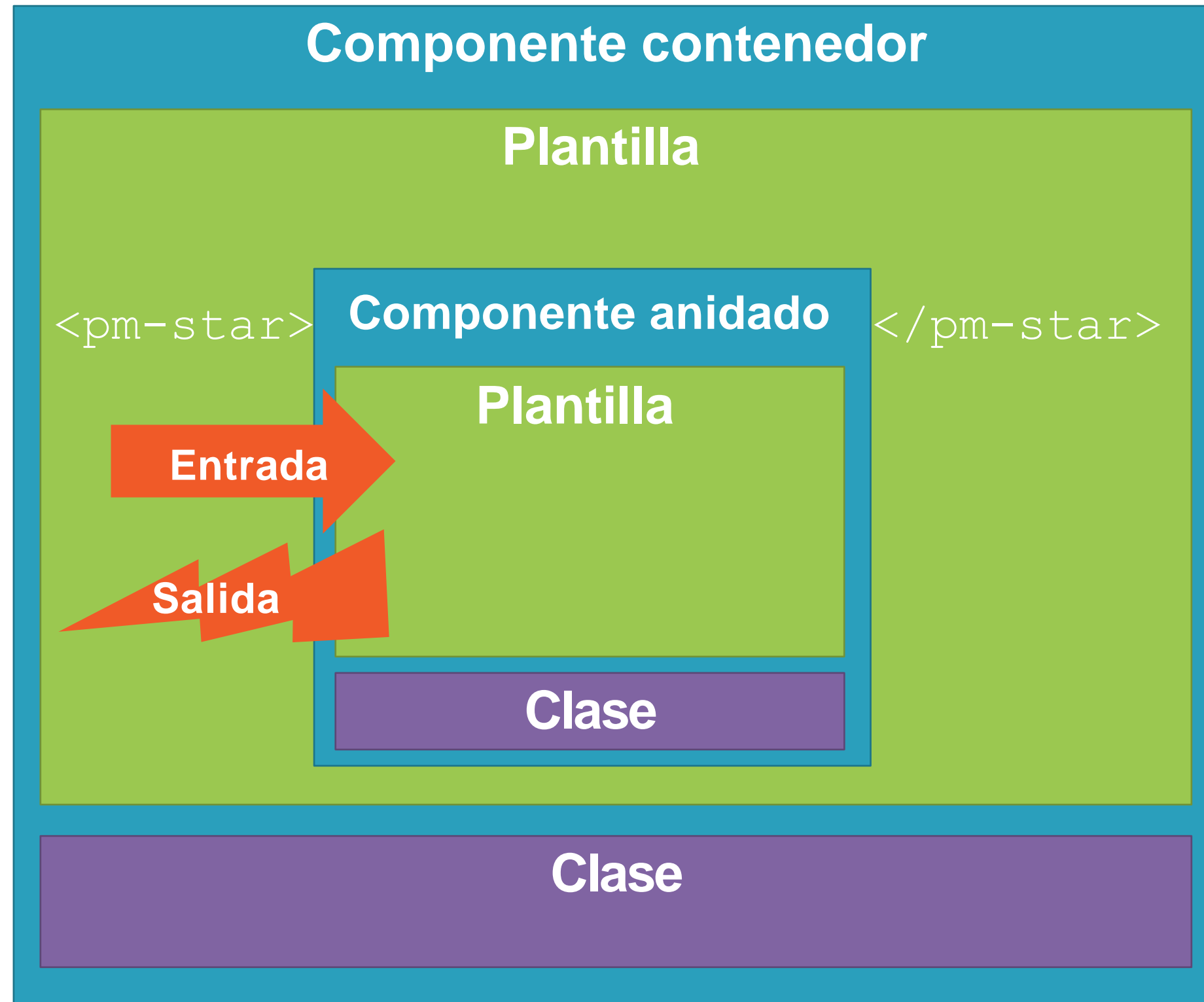
Informar a Angular sobre nuestro componente

app.module.ts

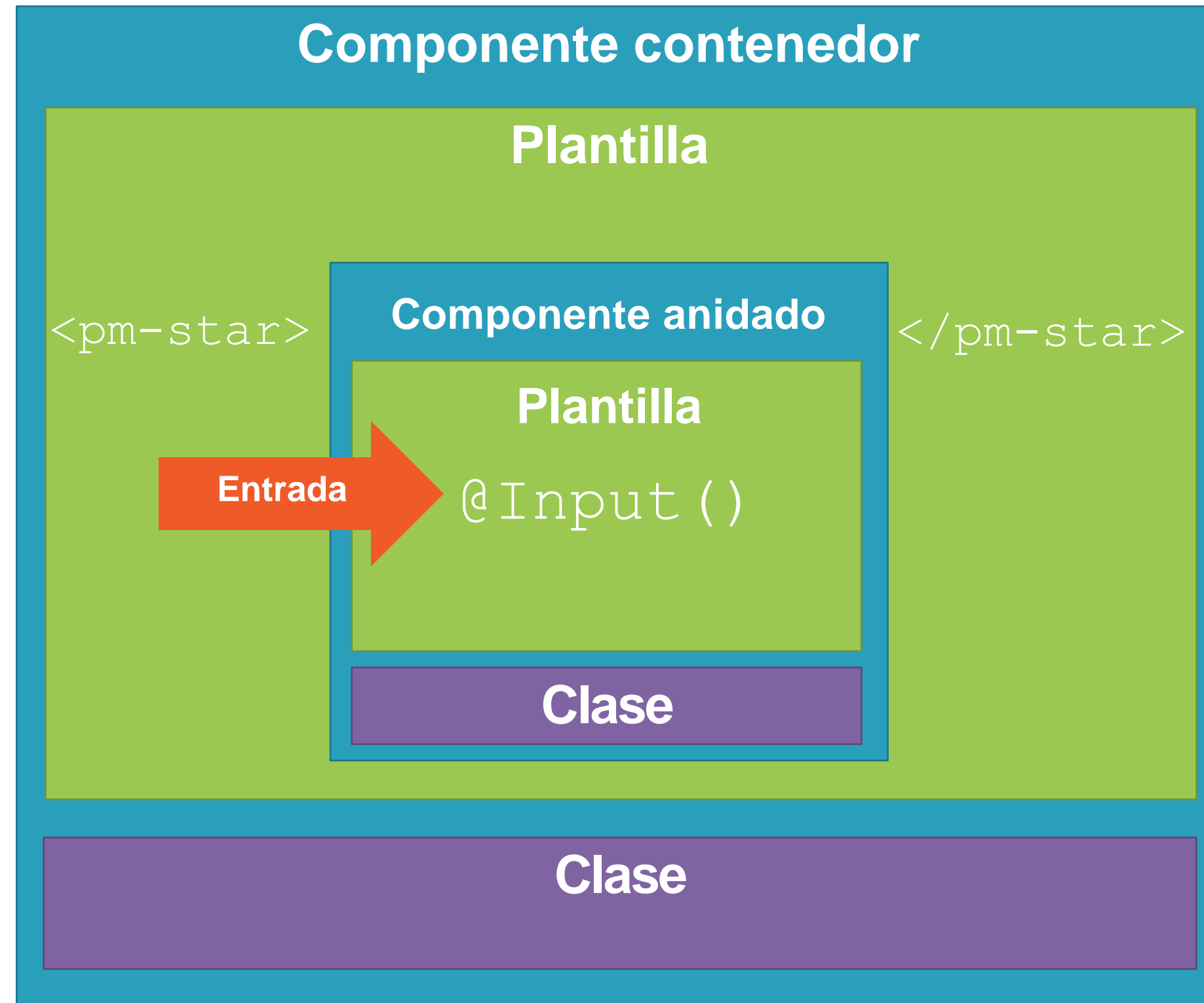
```
...
import { StarComponent } from '../shared/star.component';

@NgModule({
  imports: [
    BrowserModule,
    FormsModule ],
  declarations: [
    AppComponent,
    ProductListComponent,
    ConvertToSpacesPipe,
    StarComponent ],
  bootstrap: [ AppComponent ]
})
export class AppModule { }
```


Crear un componente anidado



Paso de datos a un componente anidado (@Input)



Pasar datos a un componente anidado (@Input)

product-list.component.ts

```
@Component ({  
  selector: 'pm-products',  
  templateUrl: './product-list.component.html'  
})  
export class ProductListComponent { }
```

product-list.component.html

```
<td>  
  <pm-star></pm-star>  
</td>
```

star.component.ts

```
@Component ({  
  selector: 'pm-star',  
  templateUrl: './star.component.html'  
})  
export class StarComponent {  
  @Input() rating: number;  
  cropWidth: number;  
}
```

Pasar datos a un componente anidado (@Input)

product-list.component.ts

```
@Component ({
  selector: 'pm-products',
  templateUrl: './product-list.component.html'
})
export class ProductListComponent { }
```

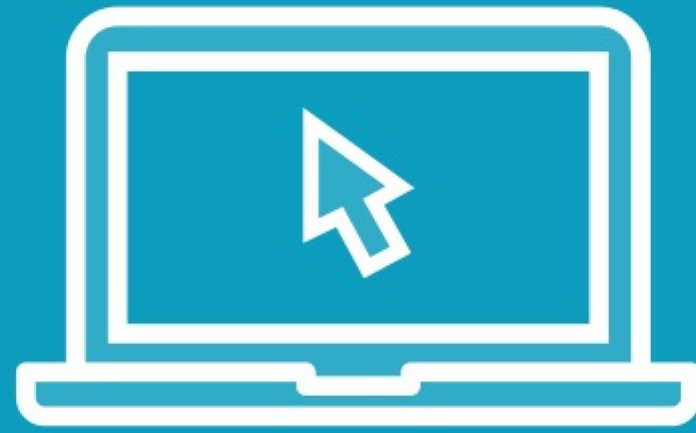
product-list.component.html

```
<td>
  <pm-star [rating]='product.starRating'>
  </pm-star>
</td>
```

star.component.ts

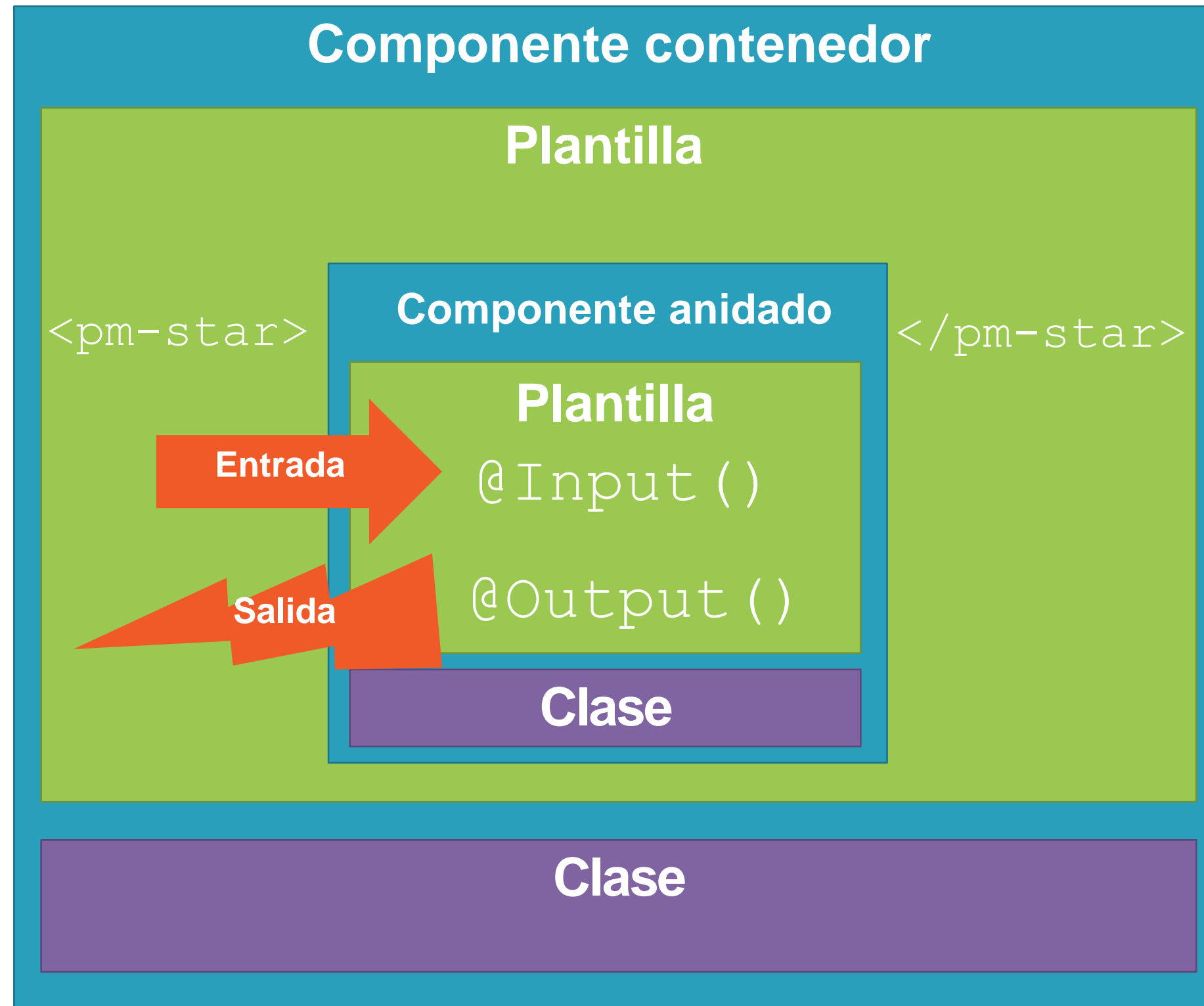
```
@Component ({
  selector: 'pm-star',
  templateUrl: './star.component.html'
})
export class StarComponent {
  @Input() rating: number;
  cropWidth: number;
}
```

Demo



Gestión de eventos

Emitir un evento (@Output)



Emitir un evento (@Output)

product-list.component.ts

```
@Component ({
  selector: 'pm-products',
  templateUrl: './product-list.component.html'
})
export class ProductListComponent { }
```

star.component.ts

```
@Component ({
  selector: 'pm-star',
  templateUrl: './star.component.html'
})
export class StarComponent {
  @Input() rating: number;
  cropWidth: number;
  @Output() notify: EventEmitter<string> =
    new EventEmitter<string>();
}
```

product-list.component.html

```
<td>
  <pm-star [rating]='product.starRating'>
  </pm-star>
</td>
```

Emitir un evento (@Output)

product-list.component.ts

```
@Component ({
  selector: 'pm-products',
  templateUrl: './product-list.component.html'
})
export class ProductListComponent { }
```

product-list.component.html

```
<td>
  <pm-star [rating]='product.starRating'>
  </pm-star>
</td>
```

star.component.ts

```
@Component ({
  selector: 'pm-star',
  templateUrl: './star.component.html'
})
export class StarComponent {
  @Input() rating: number;
  cropWidth: number;
  @Output() notify: EventEmitter<string> =
    new EventEmitter<string>();

  onClick() {
    this.notify.emit('clicked!');
  }
}
```

star.component.html

```
<div (click)='onClick()'>
  ... stars ...
</div>
```

Emitir un evento (@Output)

product-list.component.ts

```
@Component ({
  selector: 'pm-products',
  templateUrl: './product-list.component.html'
})
export class ProductListComponent { }
```

product-list.component.html

```
<td>
  <pm-star [rating]='product.starRating'
           (notify)='onNotify($event)'>
  </pm-star>
</td>
```

star.component.ts

```
@Component ({
  selector: 'pm-star',
  templateUrl: './star.component.html'
})
export class StarComponent {
  @Input() rating: number;
  cropWidth: number;
  @Output() notify: EventEmitter<string> =
    new EventEmitter<string>();

  onClick() {
    this.notify.emit('clicked!');
  }
}
```

star.component.html

```
<div (click)='onClick()'>
  ... stars ...
</div>
```

Emitir un evento (@Output)

product-list.component.ts

```
@Component ({
  selector: 'pm-products',
  templateUrl: './product-list.component.html'
})
export class ProductListComponent {
  onNotify(message: string): void { }
}
```

product-list.component.html

```
<td>
  <pm-star [rating]='product.starRating'
           (notify)='onNotify($event)'>
  </pm-star>
</td>
```

star.component.ts

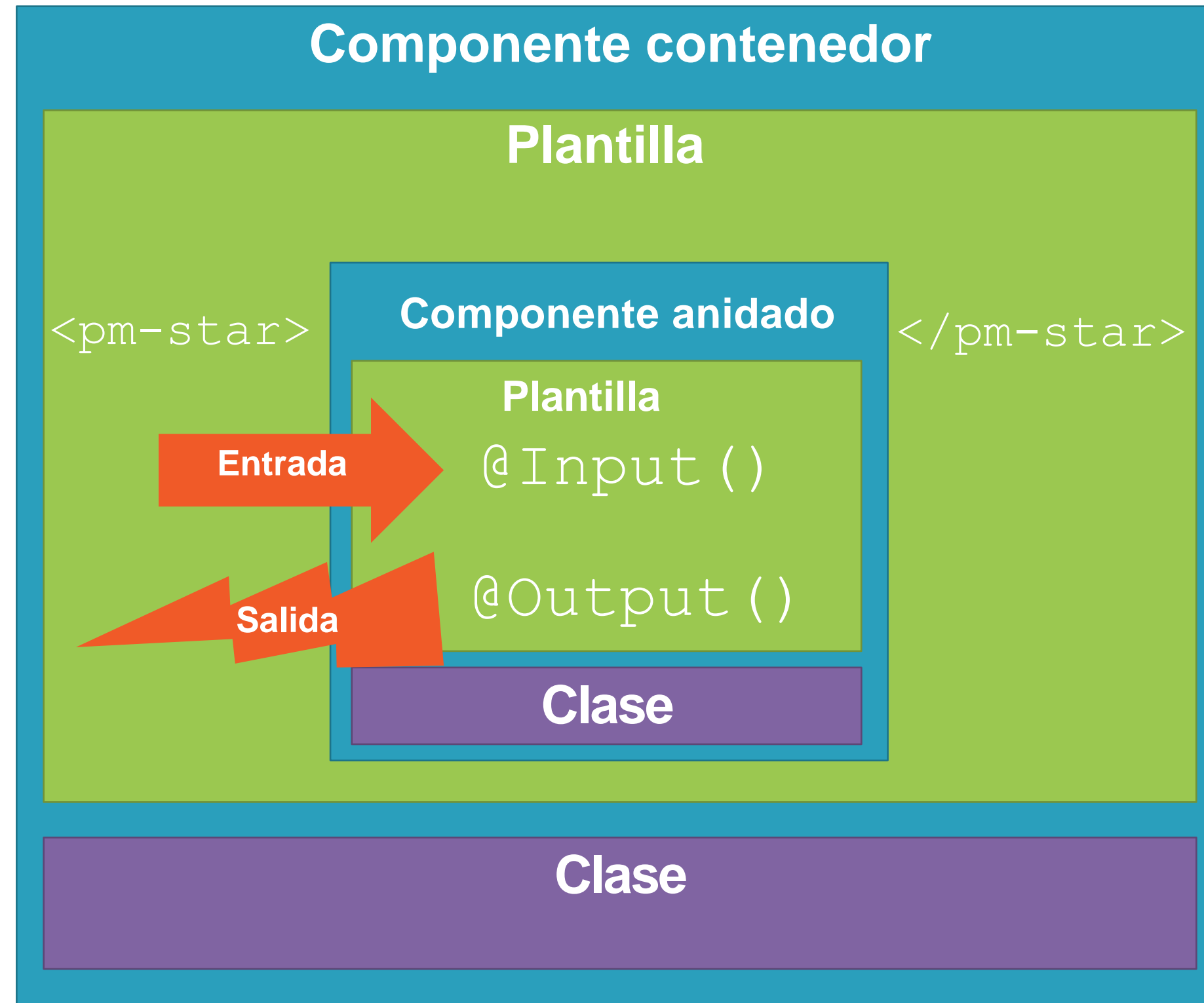
```
@Component ({
  selector: 'pm-star',
  templateUrl: './star.component.html'
})
export class StarComponent {
  @Input() rating: number;
  cropWidth: number;
  @Output() notify: EventEmitter<string> =
    new EventEmitter<string>();

  onClick() {
    this.notify.emit('clicked!');
  }
}
```

star.component.html

```
<div (click)='onClick()'>
  ... stars ...
</div>
```


API pública de componentes anidables



Lista de comprobación de anidamiento: Propiedades de entrada



- Decorador `Input`
- Adjuntar a una propiedad de cualquier tipo
- Prefijo con `@`; Sufijo con `()`

```
export class StarComponent {  
  @Input() rating: number;  
}
```

Lista de comprobación de anidamiento: Propiedades de salida



- **Decorador** `Output`
- Se adjunta a una propiedad declarada como `EventEmitter`
- Utiliza el argumento genérico para definir el tipo de datos del evento
- Utiliza la palabra clave `new` para crear una instancia del `EventEmitter`
- Prefijo con `@`; Sufijo con `()`

```
export class StarComponent {  
    @Output() notify: EventEmitter<string> =  
        new EventEmitter<string>();  
}
```

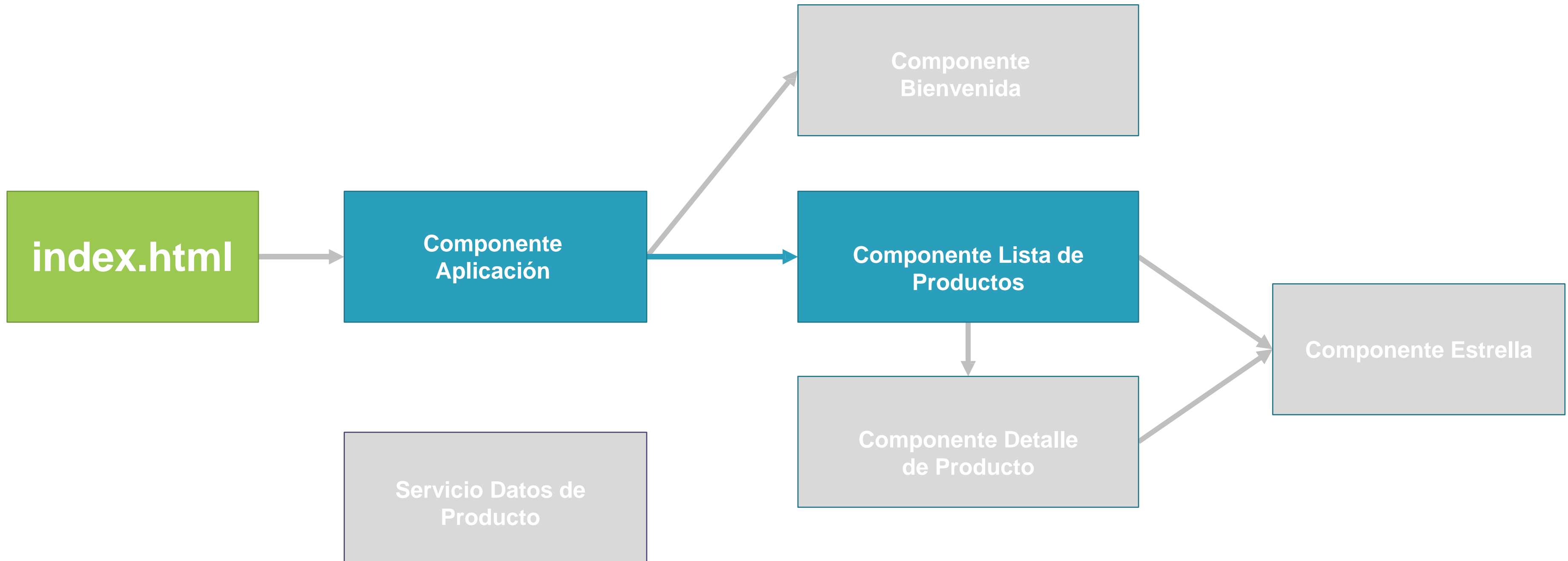
Lista de comprobación de anidamiento: Componente contenedor



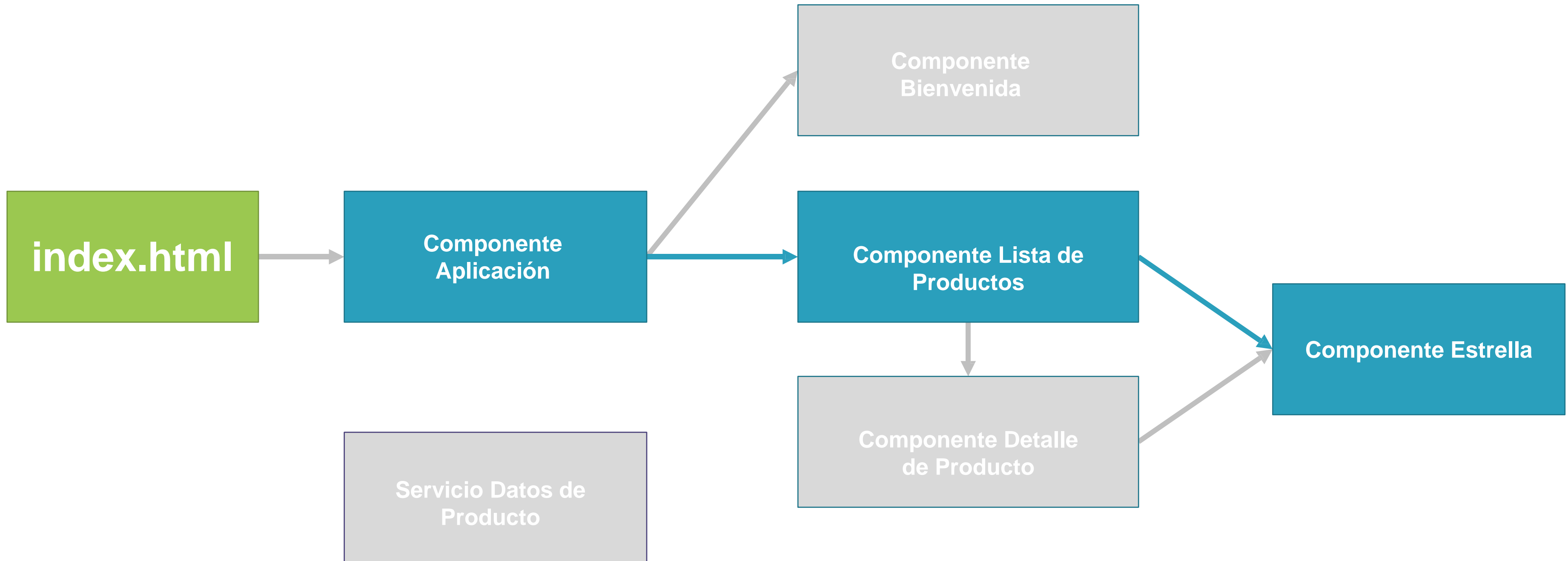
- **Utiliza la directiva**
 - Nombre de la directiva -> selector del componente anidado
- **Utiliza el enlace de propiedades para pasar datos al componente anidado**
- **Utiliza el enlace de eventos para responder a los eventos del componente anidado**
- **Utiliza \$event para acceder a los datos de eventos pasados desde el componente anidado**

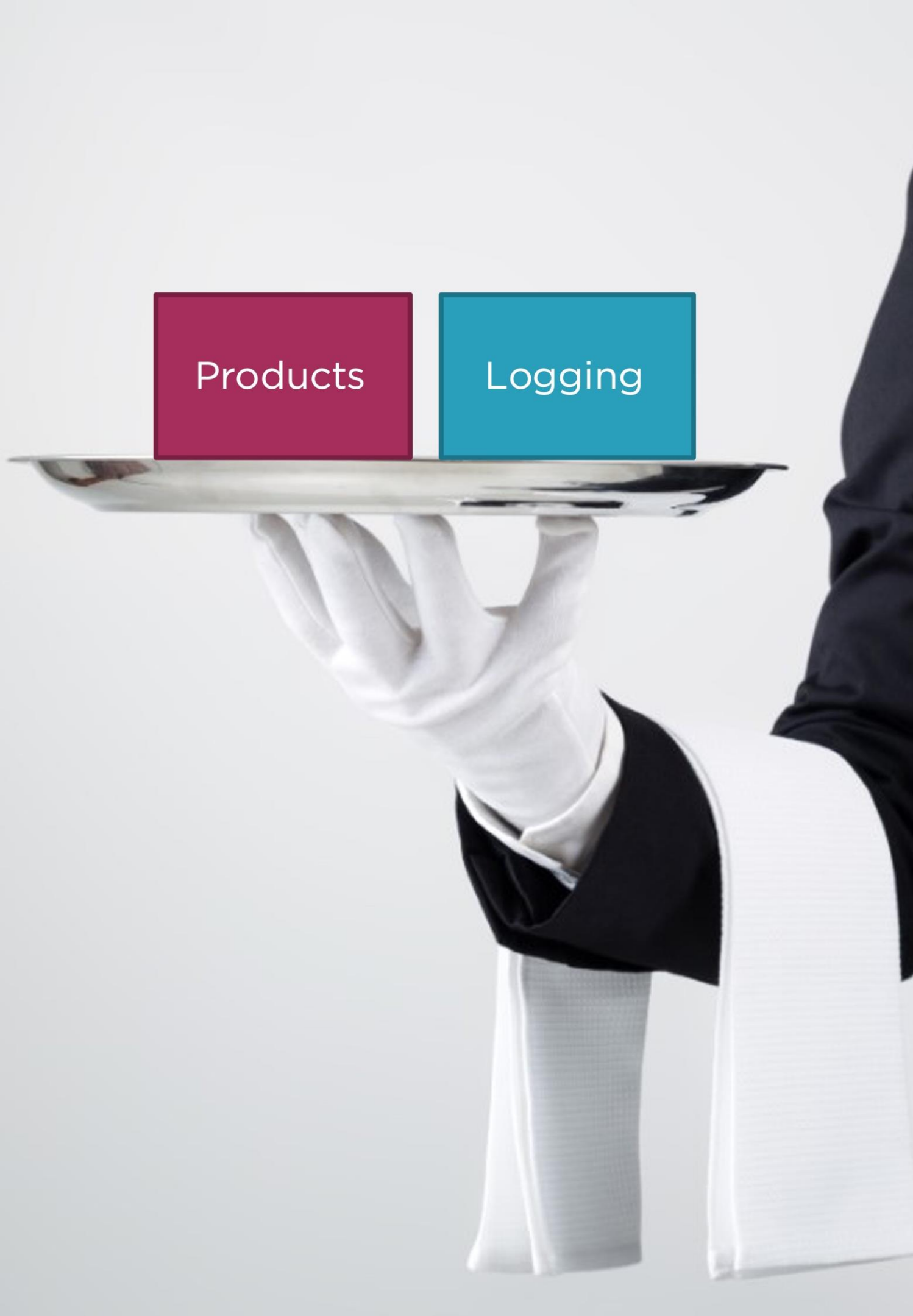
```
<pm-star [rating]='product.starRating'  
         (notify)='onNotify($event)'>  
</pm-star>
```

Arquitectura de la aplicación



Arquitectura de la aplicación





A continuación ...

Servicios e inyección de dependencias