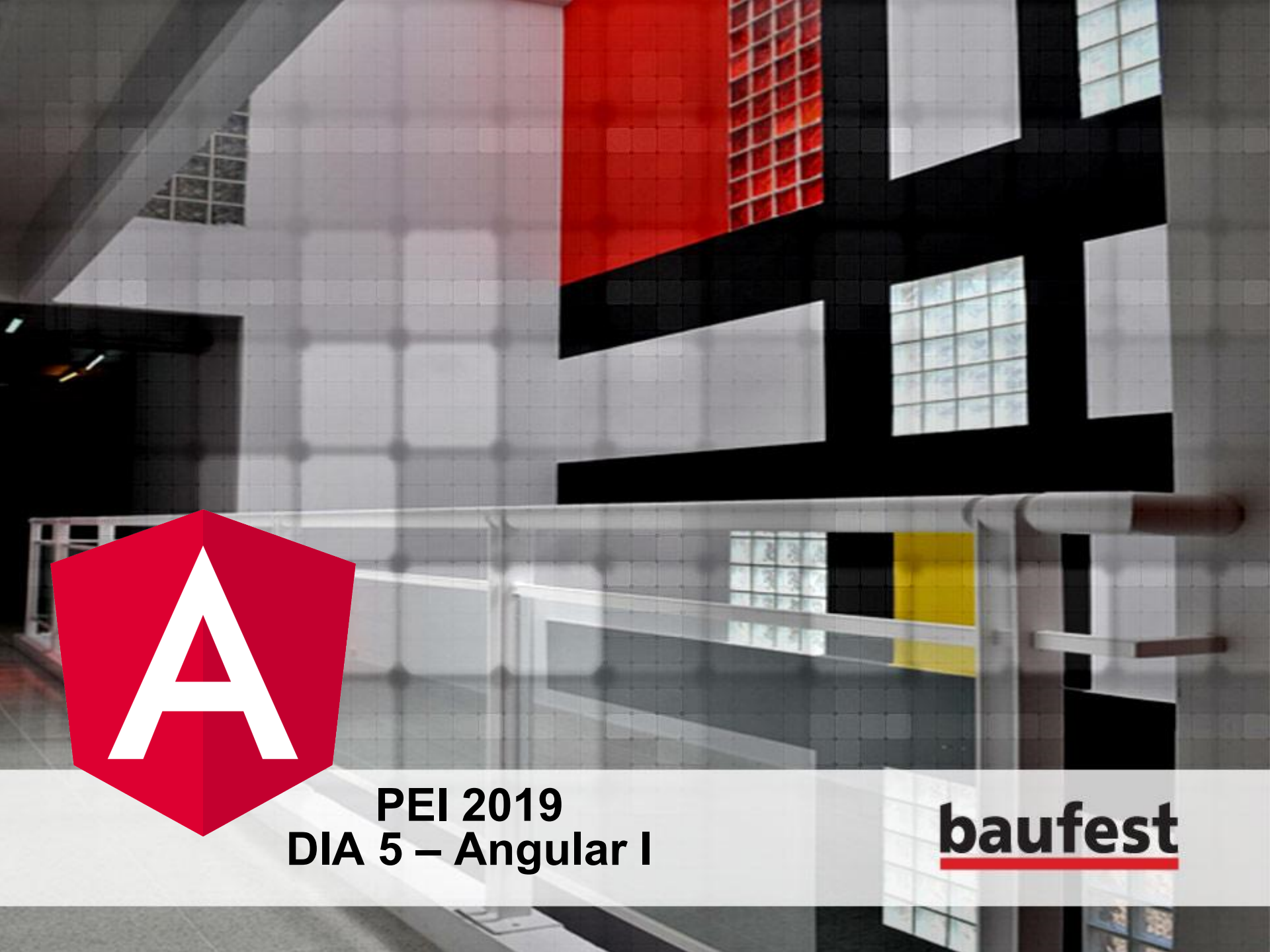


PEI 2019
DIA 5 – Angular I

baufest

Instructores

Diaz, Santiago Esteban

Panelo, Daniel Santiago

García, Maximiliano Daniel

baufest

Temario

Puntos Teóricos

- Introducción a SPA
- Frameworks JavaScript
- Introducción a Angular
- Entorno de trabajo
- Módulos
- Directivas
- Componentes
- UI data binding

Puntos Prácticos

- Primera aplicación Angular
- Agregamos el componente “jugadores”
- Agregamos un listado de jugadores

baufest

Frameworks JS

baufest

¿Qué es Angular?

baufest

¿Qué es Angular?

One framework.
Mobile & desktop.

Framework para aplicaciones web

Desarrollo de aplicaciones Single-Page

Código abierto. Mantenido por Google

PWA → Progressive Web Applications

baufest

¿Qué es Angular?

One framework.
Mobile & desktop.

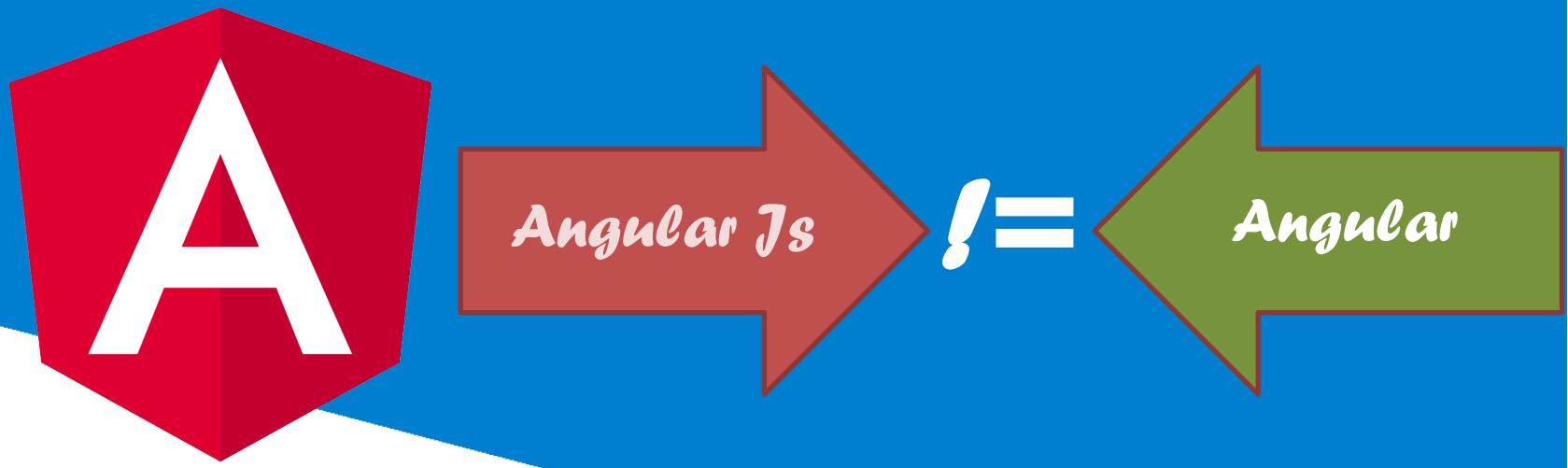
Basado en TypeScript

Angular CLI → Set de herramientas

- **-Transpilar TS → JS**
- **-Crear un servidor local**
- **-Generar recursos**

baufest

Angular != Angular Js



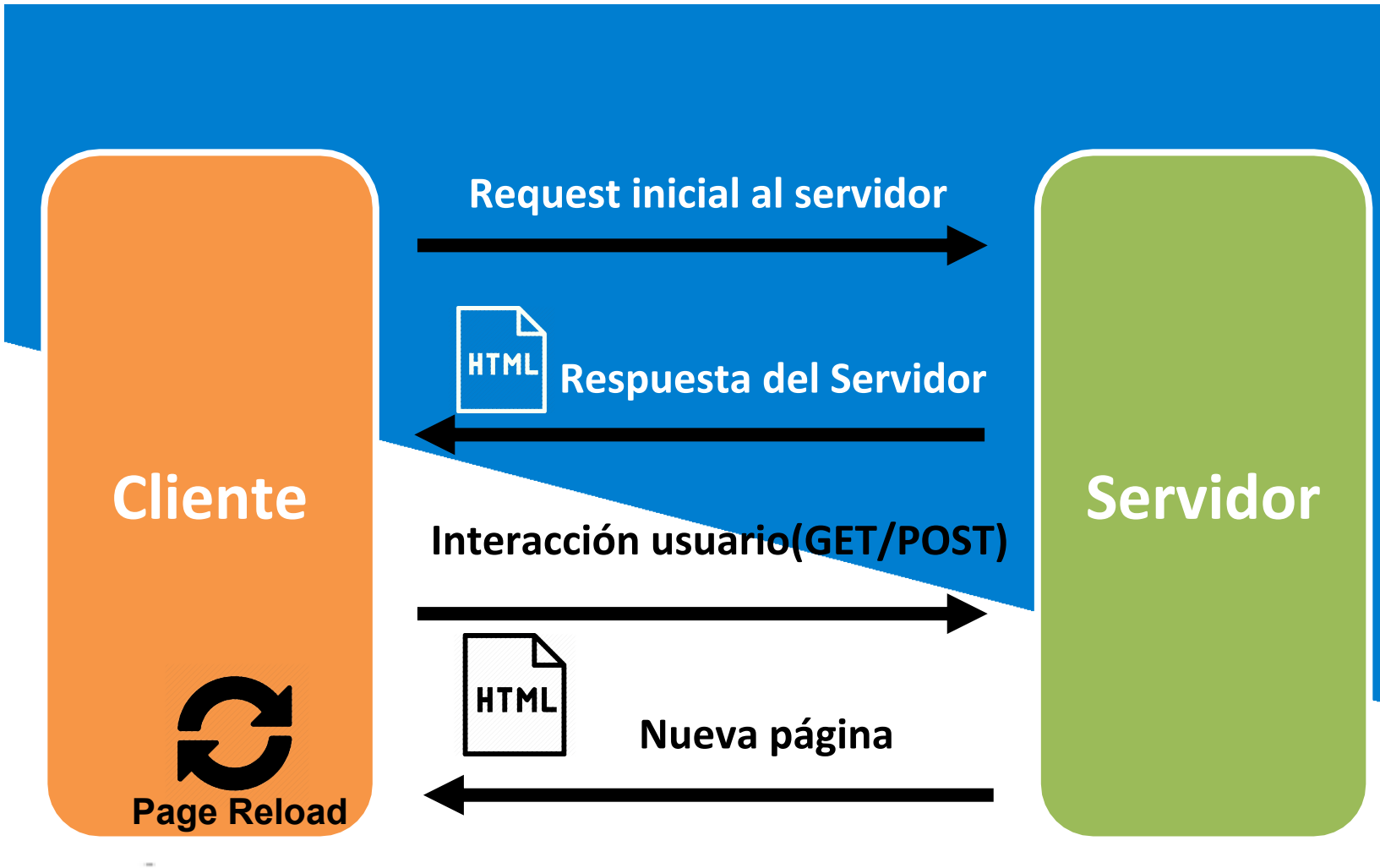
baufest

¿Qué es una SPA?

Single
Page
Application

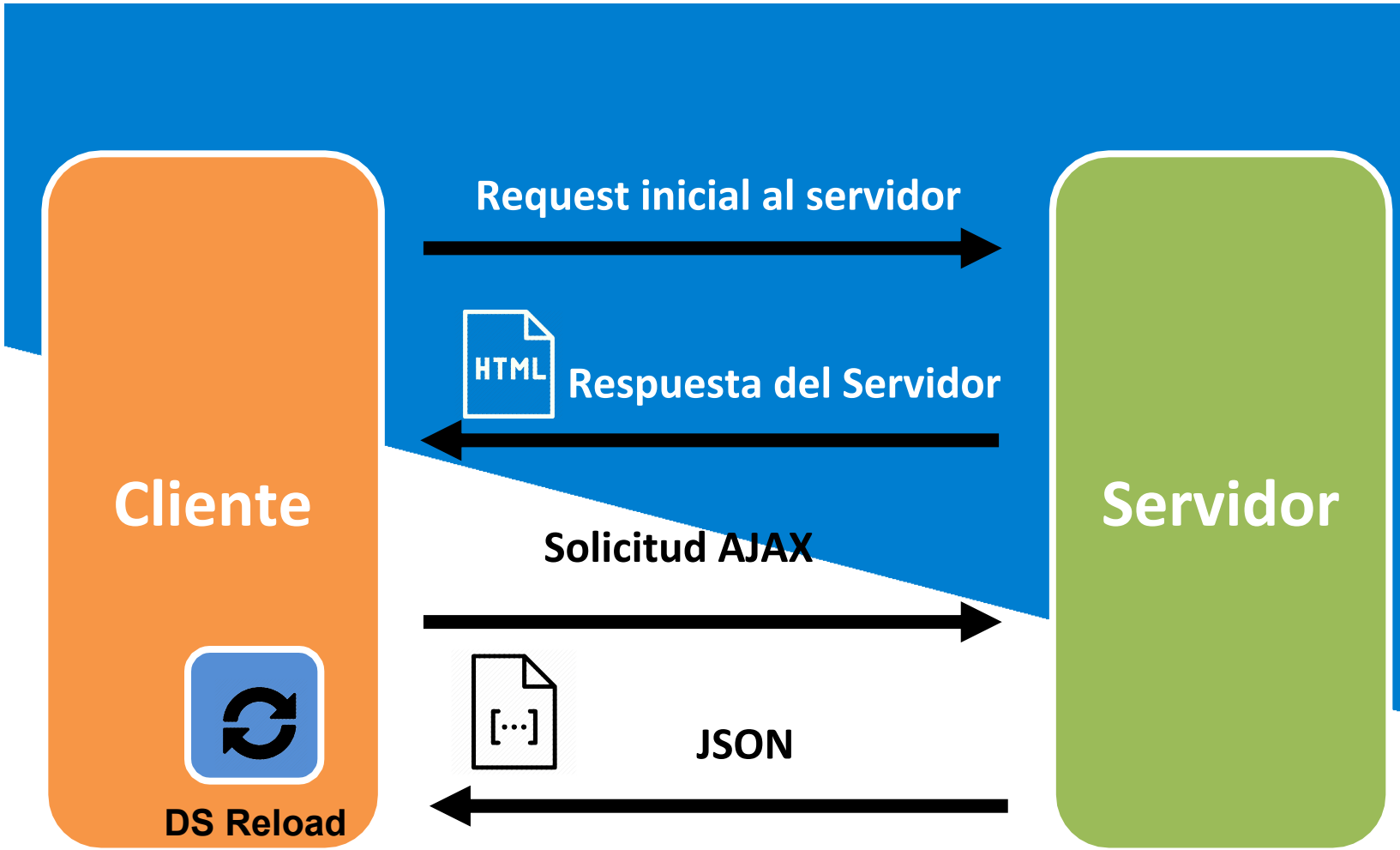
baufest

Ciclo de vida de una Web App tradicional



baufest

Ciclo de vida de una SPA

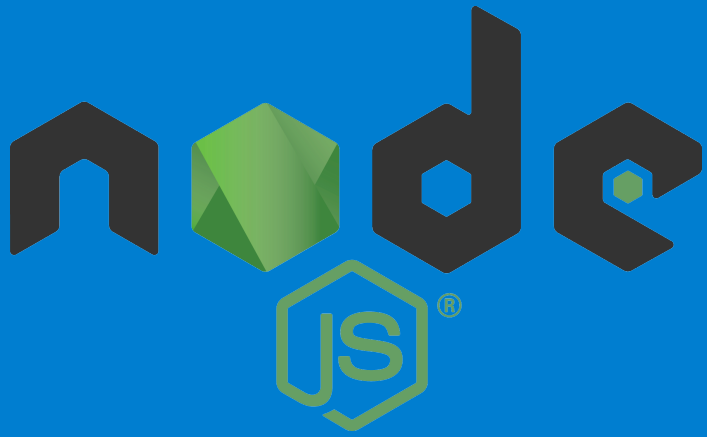


baufest

Entorno de Trabajo

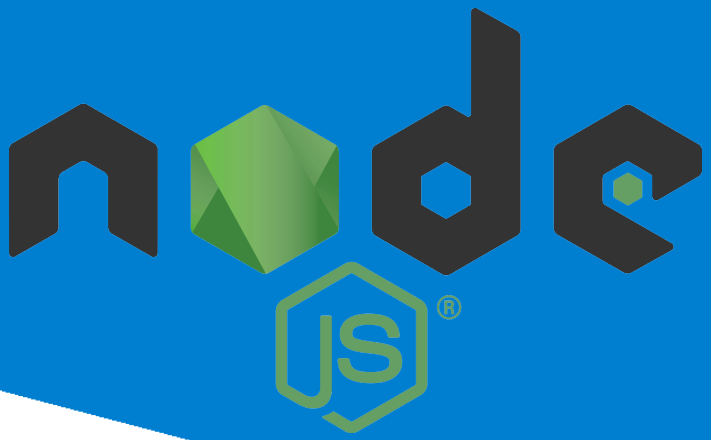
baufest

¿Qué vamos a utilizar?



baufest

Instalando Node JS



https://nodejs.org/es/

baufest

Instalando Angular CLI

npm install -g @angular/cli

baufest

baufest Tennis

baufest

Creando el Proyecto

ng new tennis-web


```
Windows PowerShell
PS C:\PEI 2019> ng new tennis-web
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? CSS
CREATE tennis-web/angular.json (3804 bytes)
CREATE tennis-web/package.json (1317 bytes)
CREATE tennis-web/README.md (1026 bytes)
CREATE tennis-web/tsconfig.json (408 bytes)
CREATE tennis-web/tslint.json (2837 bytes)
CREATE tennis-web/.editorconfig (245 bytes)
CREATE tennis-web/.gitignore (503 bytes)
CREATE tennis-web/src/favicon.ico (5430 bytes)
CREATE tennis-web/src/index.html (296 bytes)
CREATE tennis-web/src/main.ts (372 bytes)
CREATE tennis-web/src/polyfills.ts (3234 bytes)
CREATE tennis-web/src/test.ts (642 bytes)
CREATE tennis-web/src/styles.css (80 bytes)
CREATE tennis-web/src/browserslist (388 bytes)
CREATE tennis-web/src/karma.conf.js (964 bytes)
CREATE tennis-web/src/tsconfig.app.json (166 bytes)
```

baufest

Explorando el nuevo proyecto

cd tennis-web

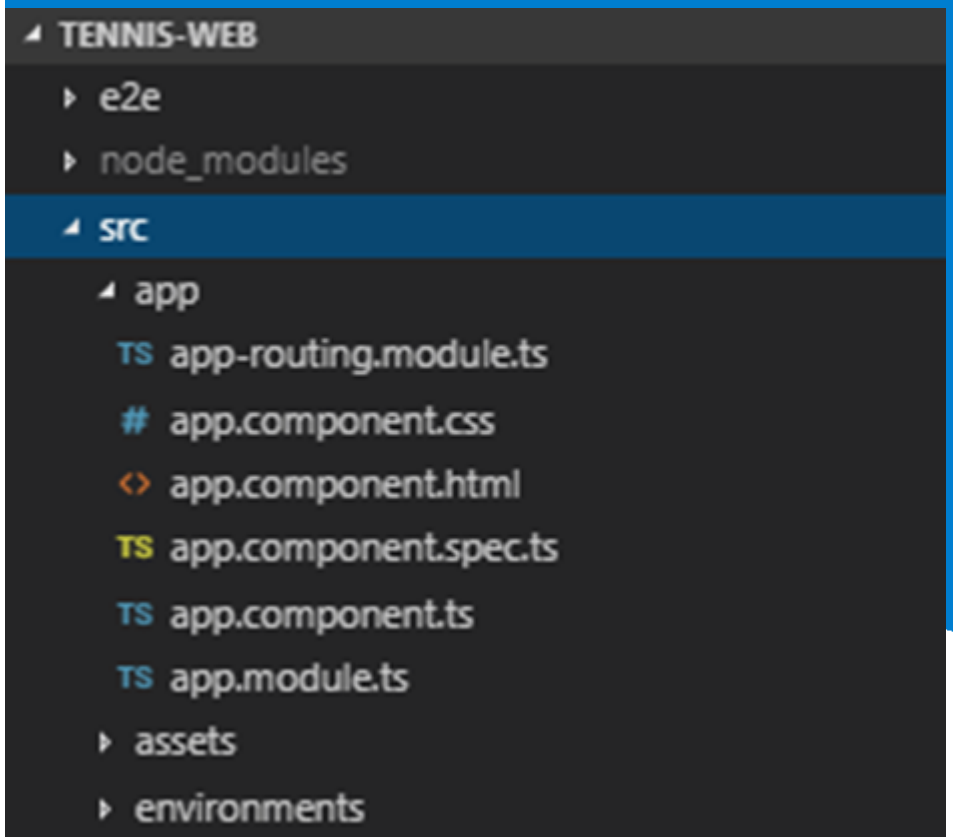
ng serve

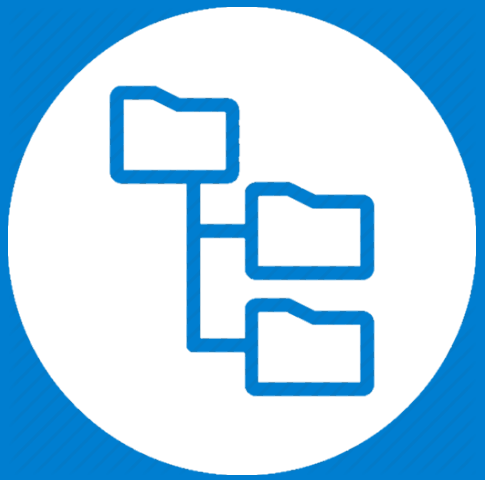

```
ng serve
PS C:\PEI 2019\tennis-web> ng serve
** Angular Live Development Server is listening on localhost:4200,
r on http://localhost:4200/ **

Date: 2019-02-19T11:49:57.596Z
Hash: ff085cb175981c5e124d
Time: 8616ms
chunk {main} main.js, main.js.map (main) 12.7 kB [initial] [rendered]
chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 236 kB [initial] [rend
ered]
chunk {runtime} runtime.js, runtime.js.map (runtime) 6.08 kB [entry] [rendered]
chunk {styles} styles.js, styles.js.map (styles) 16.2 kB [initial] [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 3.55 MB [initial] [rendered]
i [wdm]: Compiled successfully.
```

baufest

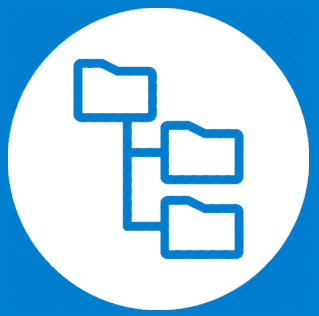
¿Cómo se estructura la aplicación?





baufest

src\app



src\app

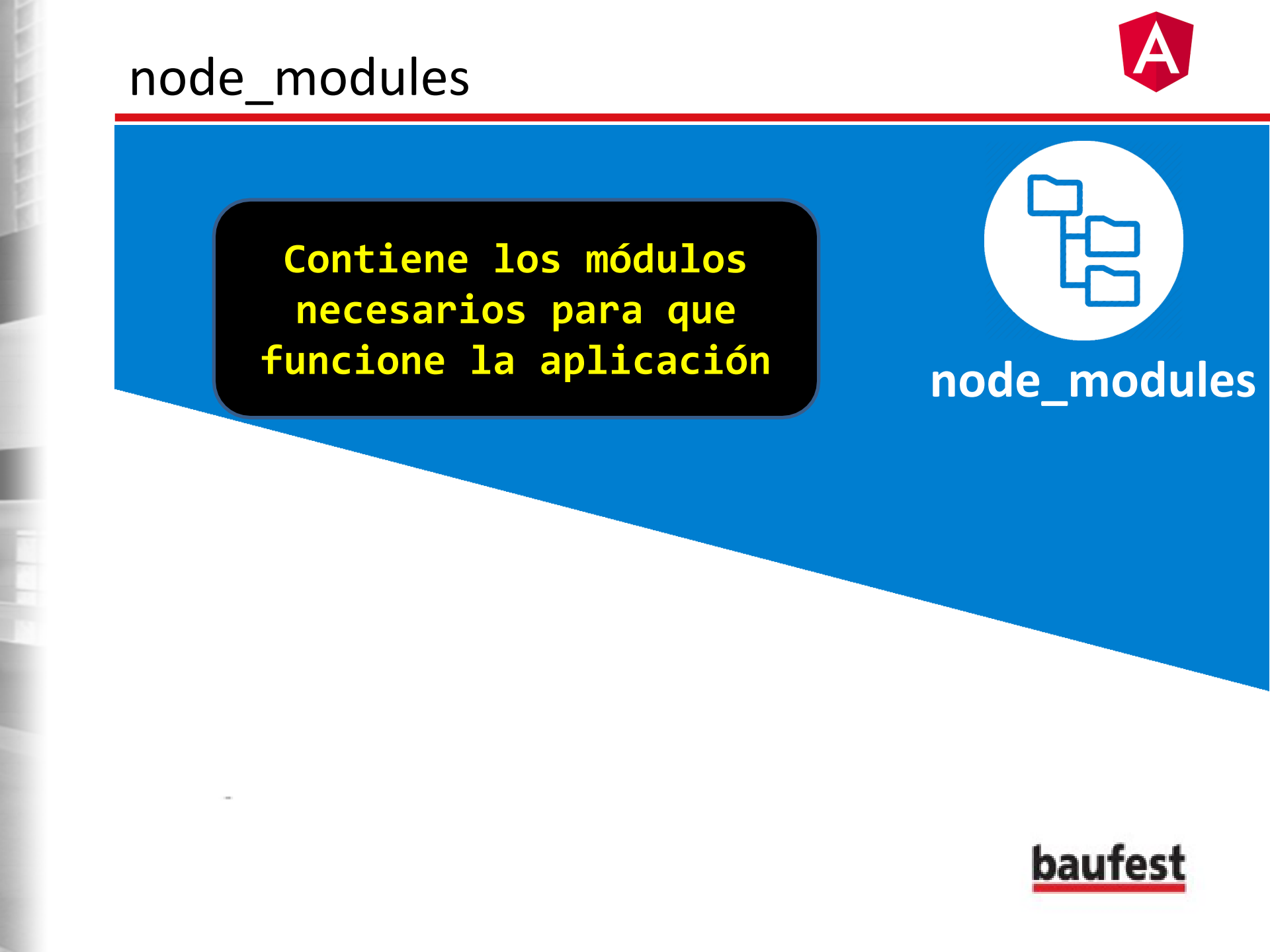
Contiene todo nuestro código

Contiene los módulos, servicios, componentes, etc.

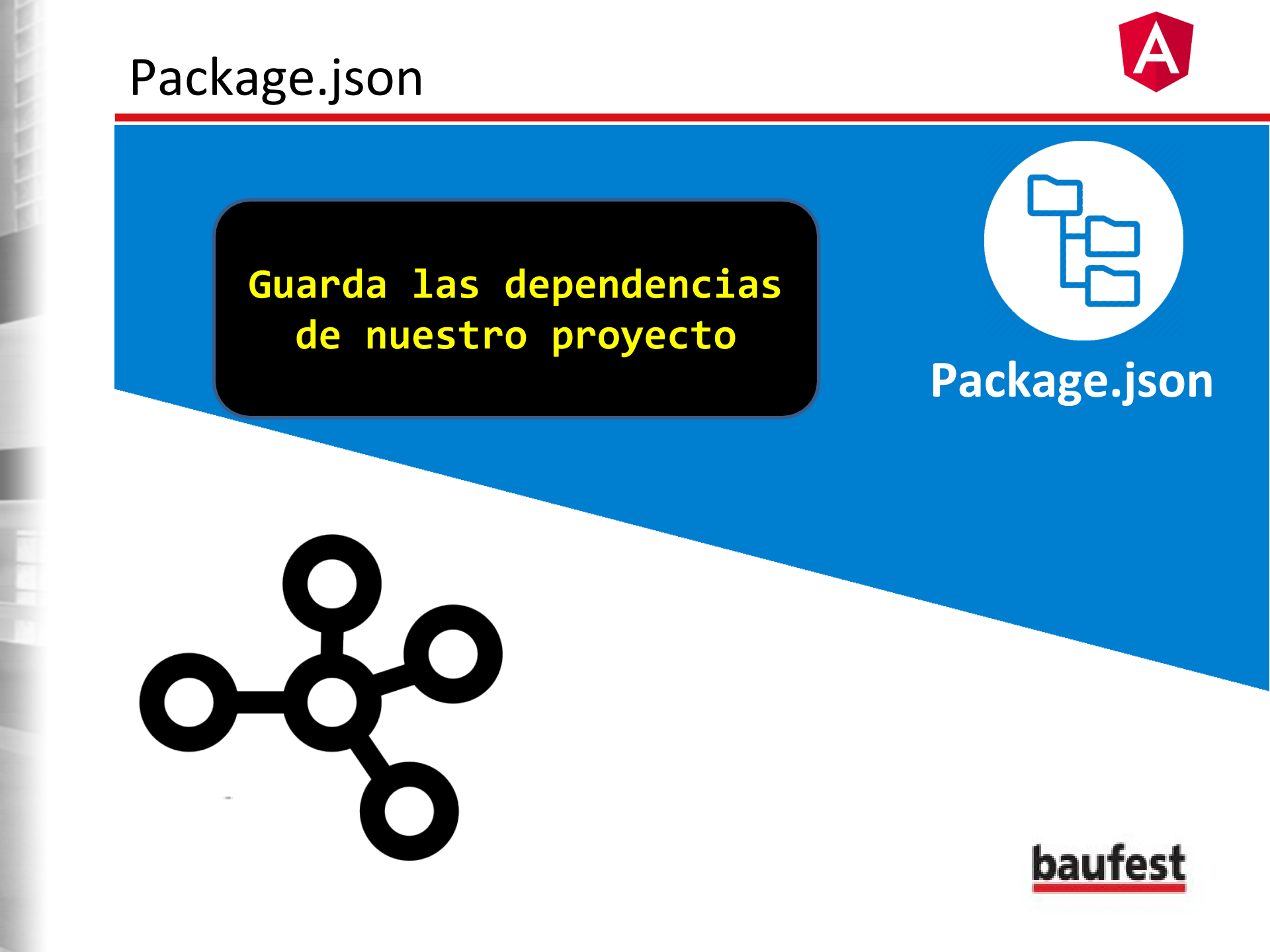
La mayoría de nuestro desarrollo lo haremos aquí

baufest

node_modules

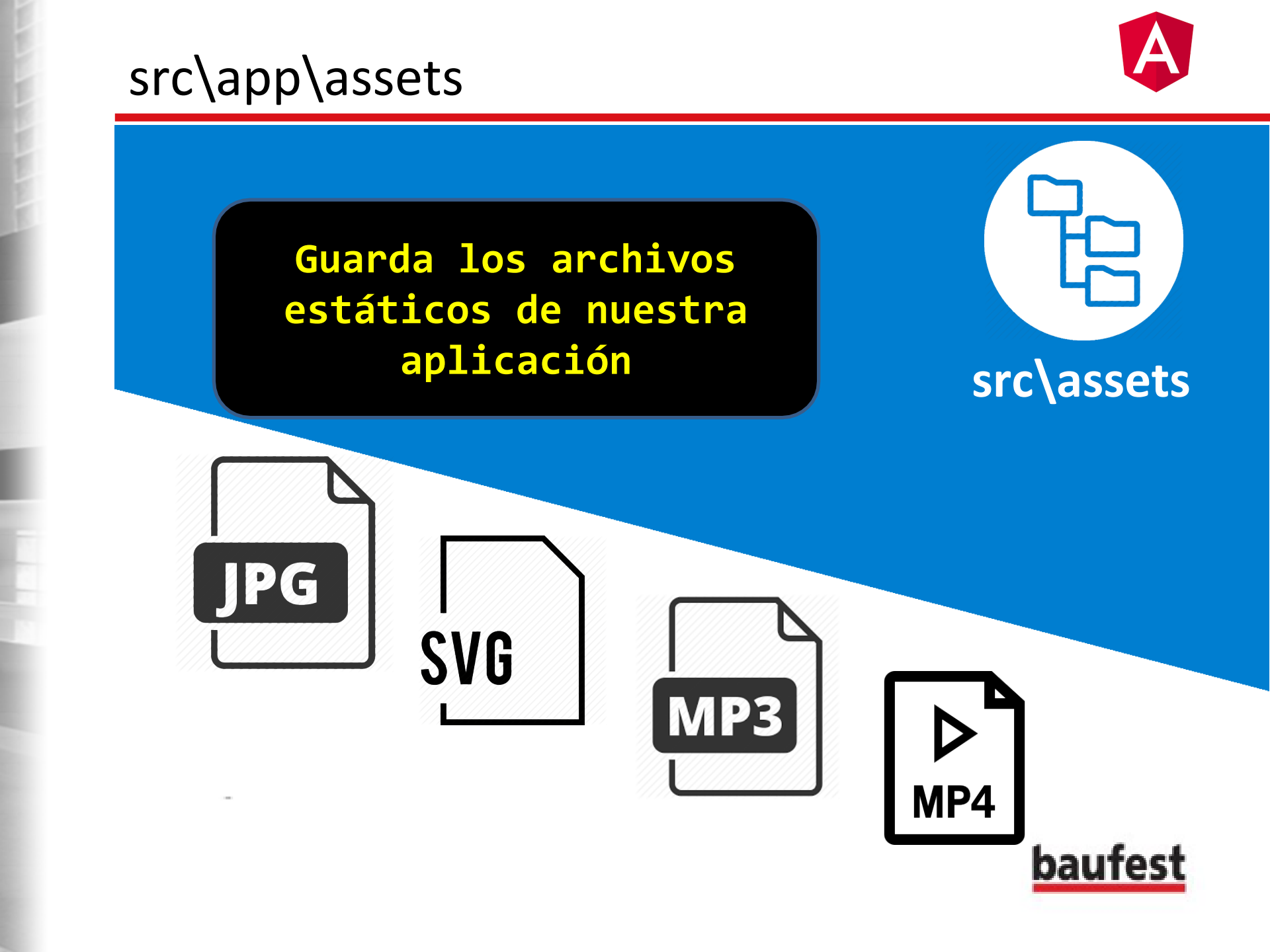


Package.json



baufest

src\app\assets



Src\styles.css

Módulos

¿Qué es un módulo?

Clase con un decorador @NgModule

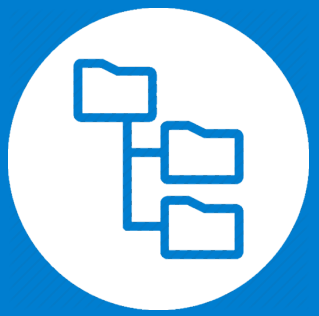
Organiza funcionalidades y las ordena en bloques

Permite extender funcionalidad con módulos externos

Provee de un entorno donde podemos utilizar templating

baufest

Src\app\app.module.ts



app.modules.ts

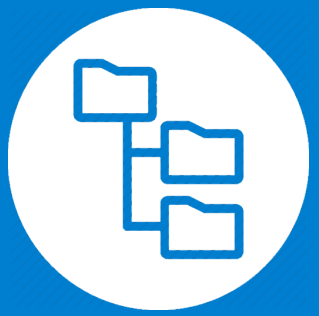
Módulo principal de la aplicación

Es el primer módulo que carga

Contiene definiciones y requerimientos

baufest

Src\app\app.module.ts



app.modules.ts

declarations:[]	Declaramos nuestros propios componentes
imports:[]	Importamos módulos de terceros o del core de angular.
providers:[]	Declaramos servicios: nuestros o de terceros
bootstrap:[]	Establecemos el componente inicial

baufest

¿Qué es un módulo? – Otros Módulos nativos

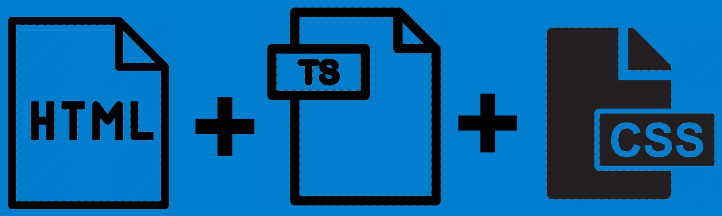
- Http
- Browser
- Forms
- Reactive Forms
- App-Routing

baufest

Componentes

Componente- ¿Qué es?

Compuesto por:



Controla una zona de espacio en la pantalla denominada “Vista”

Clase con un decorador @Component

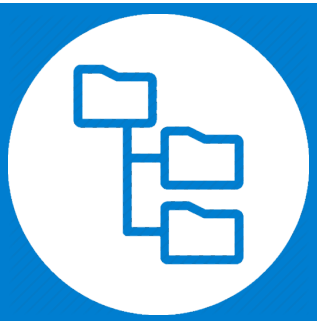
baufest

Componente- Estructura

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})

export class AppComponent {
  title = 'tennis-web'  ;
}
```



app.component.ts

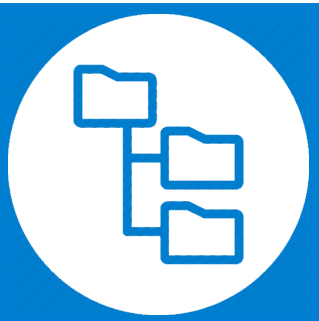
Nombramos la clase y la exportamos

selector	Tag que Angular busca en el template para renderizar el componente
templateUrl	Html se que utiliza como template
styleUrls	Hojas de Estilos

baufest

Componente- Estructura

```
<div style="text-align:center">
  <h1>
    Welcome to {{ title }}!
  </h1>
  <img width="300" alt="Angular Logo" src="data:im
</div>
<h2>Here are some links to help you start: </h2>
<ul>
  <li>
    <h2><a target="_blank" rel="noopener" href="ht
```



app.component.html

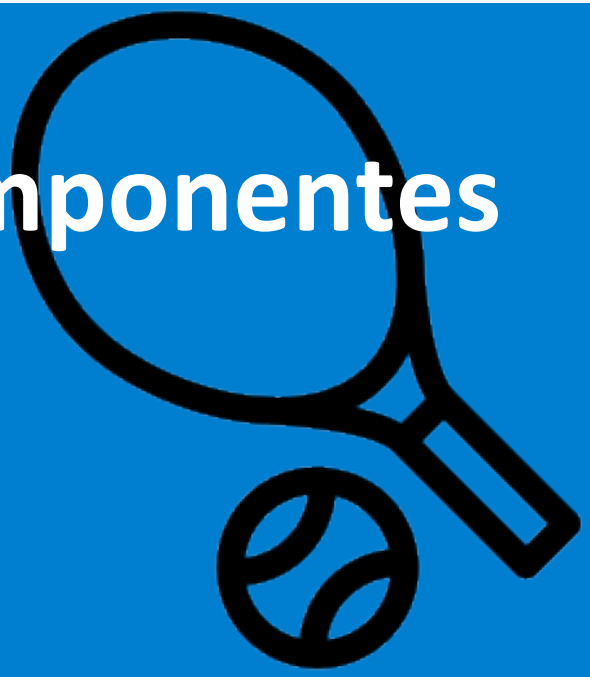
*

{{title}}

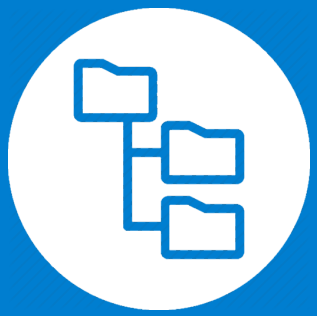
Las dobles llaves son un tipo de bindeo, es decir que en el html final se va a mostrar el valor de la propiedad 'title' de app.component.ts

baufest

Creando nuestros componentes



Componentes - Jugador



.app\jugadores

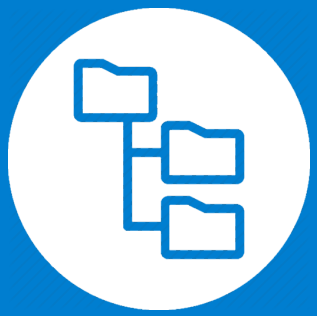
jugadores.component.ts

jugadores.component.html

jugadores.component.css

baufest

Componentes - Jugador



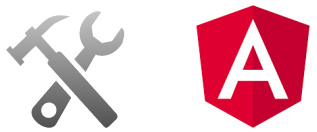
.app\jugadores

¿Cómo lo vemos?

¿Desde dónde se accede?

baufest

app-routing.module.ts



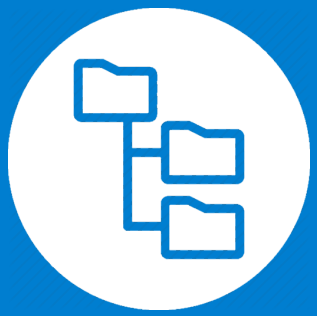
Routing

Las rutas le determinan qué vista mostrar cuando se hace click en un enlace o se pega una URL en la barra de direcciones del navegador.

path: una cadena que coincide con la URL en la barra de direcciones del navegador.
component: el componente que el enrutador debe crear al navegar a esta ruta.

baufest

app-routing.module.ts



.app-routing.module.ts

```
import { JugadoresComponent } from './jugadores/jugadores.component';

const routes: Routes = [
  { path: 'jugadores', component: JugadoresComponent },
];
```

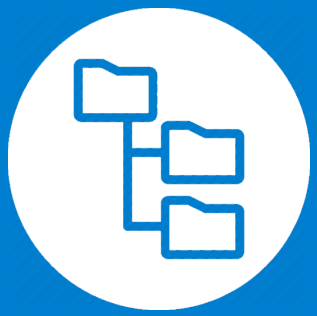
baufest

app.module

Describe cómo encajan las partes de la aplicación. Cada aplicación tiene al menos un módulo Angular, el módulo raíz que arranca para iniciar la aplicación.

Se declaran los módulos de la aplicación.
Se declaran los servicios de la aplicación.
Se importan los recursos externos que se utilizarán en la aplicación.

app.module.ts



app.module.ts

```
import { JugadoresComponent } from './jugadores/jugadores.component';

@NgModule({
  declarations: [
    AppComponent,
    JugadoresComponent
  ],
```

baufest

Selector

Embeber el módulo

.app\app.component.html

```
<app-jugadores></app-jugadores>
<router-outlet></router-outlet>
```

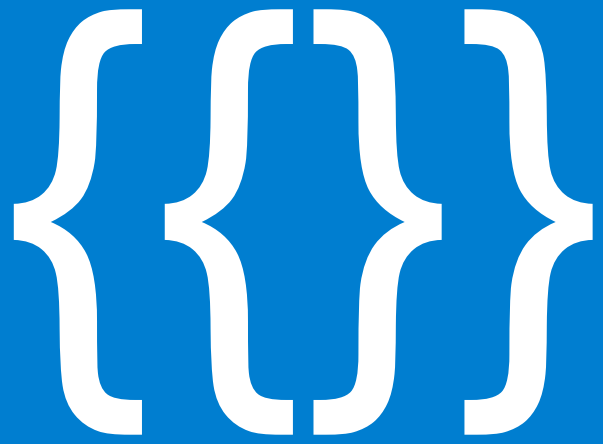
.app\jugadores\jugadores.component.ts

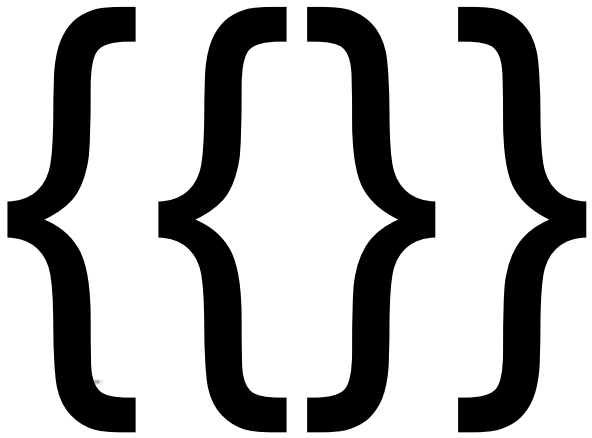
```
import { Component, OnInit } from '@angular/core';
import { Jugador } from '../models/jugador';
import { JugadorService } from '../jugador.service';

@Component({
  selector: 'app-jugadores',
  templateUrl: './jugadores.component.html',
  styleUrls: ['./jugadores.component.css']
})
```

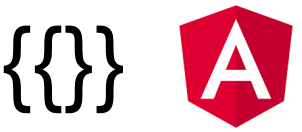
baufest

Bindings





Bindgs – ¿Qué son?



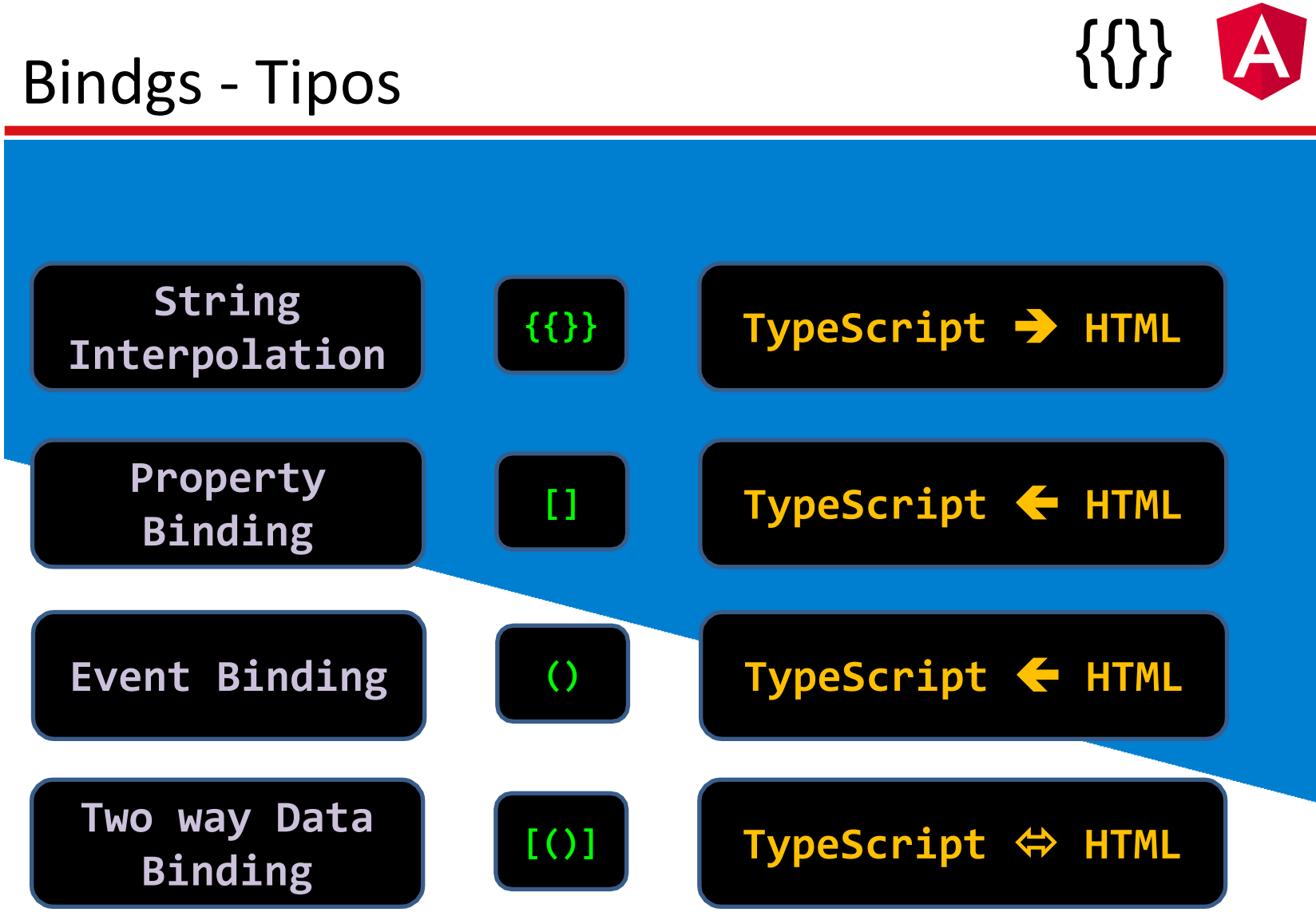
Son la sincronización automática de los datos entre los componentes del modelo y la vista.

La vista es una proyección del modelo en todo momento.

Cuando el modelo cambia, la vista refleja el cambio, y viceversa.

baufest

Bindgs - Tipos



baufest

Bindgs – String Interpolation

String Interpolation

{{}}

TypeScript => HTML

Bindea desde el TypeScript al HTML mediante el uso de {{}}

app.component.ts

```ts
//..
export class AppComponent {
  title = 'tennis-web';
  a = 5
  b = 3
}
```

app.component.html

```html
<h1>
  Welcome to {{ title }}!
</h1>
<h1>
  Resultado: {{ a + b }}!
</h1>
```

Welcome to tennis-web!

Resultado: 8!

baufest

Bindgs – Property Binding

Property Binding

[]

TypeScript ← HTML

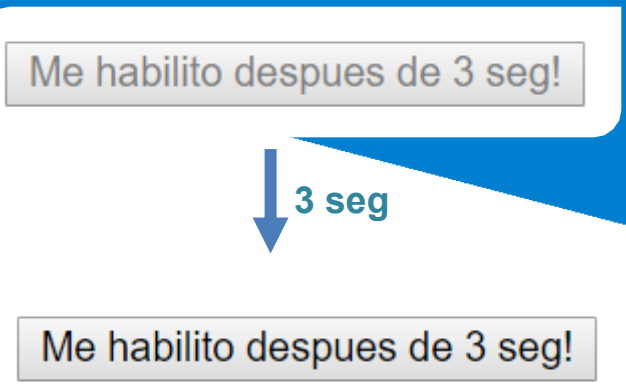
Bindea desde el HTML a TypeScript mediante el uso de []

app.component.ts

```ts
//..
export class AppComponent {
  title = 'tennis-web';
  listo = false;
  constructor(){
    setTimeout(() => {
      this.listo = true
    }, 3000)
  }
}
```

app.component.html

```html
<button [disabled]="!listo">
  Me habilito despues de 3 seg!
</button>
```



baufest

Bindgs – Event Binding

Event Binding

()

TypeScript ← HTML

Bindea eventos del .html con funciones definidas en .ts

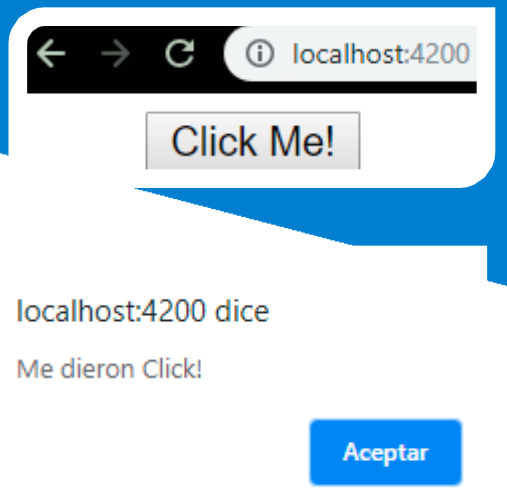
app.component.ts

```ts
//..
export class AppComponent {
  title = 'tennis-web';

  onClickEventHandler(){
    alert("Me dieron Click!")
  }
}
```

app.component.html

```html
<button (click)="onClickEventHandler()">
  Click Me!
</button>
```



baufest

Bindgs – Event Binding

Two way Data Binding

[()]

TypeScript ⇔ HTML

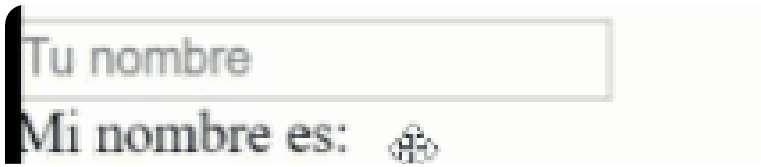
Bindea propiedades del .ts con valores del .html Las modificaciones del .html modifica el .ts y viceversa

app.component.html

```
<input type="text"
 placeholder="Tu nombre"
 [(ngModel)]="nombre" />
Mi nombre es: {{nombre}}
```

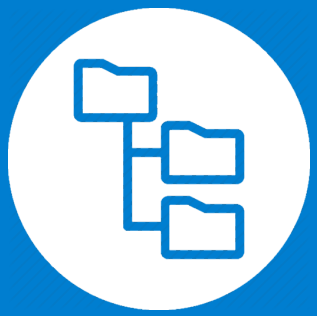
app.component.ts

```
//..
export class AppComponent {
  title = 'tennis-web';
  nombre = '';
}
```



baufest

Model - Jugador



.app\models

jugador.ts

```
export class Jugador {
    public id: number;
    public nombre: string;
    public puntos: number;
}
```

baufest

Directivas

Directivas – ¿Qué son?

Las directivas son marcas en los elementos del árbol DOM, en los nodos del HTML, que indican al compilador de Angular que debe asignar cierto comportamiento a dichos elementos o transformarlos según corresponda.

Directivas – ¿Qué son?

Existen dos tipos principales de directivas:

.De Componente

. Administra una región del HTML de manera nativa como un elemento del HTML

. Comienzan con * y camelCase

.De Atributos

. Cambia la apariencia o comportamiento de un elemento, componente u otra directiva

. Encerrada entre []

baufest

Directivas Nativas

ngFor	*ngFor	Ejecuta bucles sobre elementos del DOM
ngIf	*ngIf	Muestra/oculta elementos del DOM
ngStyle	[ngStyle]	Setea estilos de elementos del DOM
ngClass	[ngClass]	Setea clases de elementos del DOM
ngSwitch	[ngSwitch]	Ejecuta casos condicionales sobre elementos del DOM
	*ngSwitchCase	

baufest

Directivas – ngFor

ngFor

*ngFor

Ejecuta bucles sobre elementos del DOM

Itera sobre alguna propiedad declarada en el .ts y genera elementos en el .html

app.component.ts

```
//..
export class AppComponent {
  title = 'tennis-web';
  instructores:any =[
    {nombre:'Daniel', apellido:'Panelo'},
    {nombre:'Maximiliano', apellido:'Garcia'},
    {nombre:'Santiago', apellido:'Diaz'}
  ]
}
```

app.component.html

```
<ul>
    <li *ngFor="let i of instructores">
      {{i.apellido}}, {{i.nombre}}
    </li>
</ul>
```

- Panelo, Daniel
- Garcia, Maximiliano
- Diaz, Santiago

baufest

Directivas – ngIf

ngIf

***ngIf**

Muestra/oculta elementos del DOM

Evalúa una propiedad o una expresión booleana y oculta/muestra elementos en el .html

app.component.ts

```
//..
export class AppComponent {
  title = 'tennis-web';
  instructores:any =[
    {nombre:'Daniel', apellido:'Panelo', activo: true},
    {nombre:'Maximiliano', apellido:'Garcia', activo: true},
    {nombre:'Santiago', apellido:'Diaz', activo: false}
  ]
}
```

app.component.html

```
<ul>
    <ng-container *ngFor="let i of instructores">
      <li *ngIf="i.activo">
        {{i.apellido}}, {{i.nombre}}
      </li>
    </ng-container>
</ul>
```

- Panelo, Daniel
- Garcia. Maximiliano

baufest

Directivas – [ngStyle]

ngStyle

[ngStyle]

Setea estilos de elementos del DOM

Evalúa una propiedad del modelo y setea un estilo en el elemento del DOM

app.component.ts

```
//..
export class AppComponent {
  title = 'tennis-web';
  instructores:any =[
    {nombre:'Daniel', apellido:'Panelo', activo: true},
    {nombre:'Maximiliano', apellido:'Garcia', activo: true},
    {nombre:'Santiago', apellido:'Diaz', activo: false}
  ]
}
```

app.component.html

```
<ul>
  <ng-container *ngFor="let i of instructores">
    <li [ngStyle]="{background: (i.activo) ? 'green': 'red'}">
      {{i.apellido}}, {{i.nombre}}
    </li>
  </ng-container>
</ul>
```

- Panelo, Daniel
- Garcia, Maximiliano
- Diaz, Santiago

baufest

Directivas – [ngClass]

ngClass

[ngClass]

Setea clases de elementos del DOM

Evalúa una propiedad del modelo y setea una clase en el elemento del DOM

app.component.ts

```
//..
export class AppComponent {
  title = 'tennis-web';
  instructores:any =[
    {nombre:'Daniel', apellido:'Panelo', activo: true},
    {nombre:'Maximiliano', apellido:'Garcia', activo: true},
    {nombre:'Santiago', apellido:'Diaz', activo: false}
  ]
}
```

app.component.html

```
<ul>
  <ng-container *ngFor="let i of instructores; let inx = index">
    <li [ngStyle]="{background: (i.activo) ? 'green': 'red'}"
    [ngClass]="{'filaPar': (inx+1) % 2 !== 0}">
      {{i.apellido}}, {{i.nombre}}
    </li>
  </ng-container>
</ul>
```

- Panelo, Daniel
- Garcia, Maximiliano
- Diaz, Santiago

baufest

Directivas – [ngSwitch] *ngSwitchCase

ngSwitch

[ngSwitch]

*ngSwitchCase

Ejecuta casos condicionales sobre elementos del DOM

Evalúa una propiedad del modelo y modifica el dom según valores determinados de la misma

app.component.ts

```
//..
export class AppComponent {
  title = 'tennis-web';
  instructores:any =[
    {nombre:'Daniel', apellido:'Panelo', especialidad: 1},
    {nombre:'Maximiliano', apellido:'Garcia', especialidad: 2},
    {nombre:'Santiago', apellido:'Diaz', especialidad: 3}
  ]
}
```

app.component.html

```
<ul>
    <ng-container *ngFor="let i of instructores"
                  [ngSwitch]="i.especialidad">
        <li> {{i.apellido}}, {{i.nombre}} <b>(
            <span *ngSwitchCase="1">.Net</span>
            <span *ngSwitchCase="2">Java</span>
            <span *ngSwitchCase="3">Go</span>
            <span *ngSwitchDefault>N/C</span> )</b>
        </li>
    </ng-container>
</ul>
```

- Panelo, Daniel (**.Net**)
- Garcia, Maximiliano (**Java**)
- Diaz, Santiago (**Go**)

baufest