

# Preguntas Angular-v-1

Gustavo García, 19 de octubre de 2022.

## Angular como plataforma

Explicar la diferencia<sup>1</sup> entre Angular<sup>2</sup> y Angularjs<sup>3</sup>.

Qué hay que instalar en la PC para comenzar a desarrollar una aplicación de frontend en Angular.

Qué significa SPA<sup>4, 5</sup>.

## Arquitectura

Cuáles son los patrones de diseño relevantes en Angular<sup>6</sup>.

Explicar la arquitectura de una aplicación en Angular<sup>7</sup>.

Qué es un template en Angular<sup>8</sup>.

Qué son template statements en Angular<sup>9</sup>.

Qué son las variables de template en Angular<sup>10</sup>.

Cómo se declara una variable de template en Angular<sup>11</sup>.

Cómo se referencia una variable de template en Angular<sup>11</sup>.

Qué es ligar un array de parámetros de enlace (link parameters array<sup>12</sup>) a una directiva RouterLink.

Qué es y qué contiene un array de parámetros de link (link parameters array)<sup>12</sup>.

Que hace el siguiente tag<sup>12</sup>: `<a [routerLink]="['/path']">Some comment</a>`

## Comandos

Qué hace el comando ng serve y cuáles son sus opciones<sup>13</sup>.

## Componentes

Qué archivos forman el componente generado en Angular.

Qué es un componente en Angular<sup>14</sup>.

Qué es el selector de un componente<sup>15, 16</sup>.

## Binding

Qué hace el data binding en Angular<sup>17</sup>.

Qué tipos de data binding que existen en Angular<sup>18</sup>.

Explicar la relación entre los tipos de data binding y sus targets en Angular<sup>19</sup>.

Qué es interpolación de expresiones en Angular, y por qué se llama así<sup>20, 21</sup>.

---

<sup>1</sup> <https://www.campusmvp.es/recursos/post/las-10-principales-diferencias-entre-angularjs-y-angular.aspx>

<sup>2</sup> <https://angular.io>

<sup>3</sup> <https://angularjs.org/>

<sup>4</sup> <https://developer.mozilla.org/en-US/docs/Glossary/SPA>

<sup>5</sup> [https://es.wikipedia.org/wiki/Single-page\\_application](https://es.wikipedia.org/wiki/Single-page_application)

<sup>6</sup> <https://angular-enterprise.com/en/ngpost/courses/design-patterns/>

<sup>7</sup> <https://angular.io/guide/architecture>

<sup>8</sup> <https://angular.io/guide/glossary#template>

<sup>9</sup> <https://angular.io/guide/template-statements#template-statements>

<sup>10</sup> <https://angular.io/guide/template-reference-variables#understanding-template-variables>

<sup>11</sup> <https://angular.io/guide/template-reference-variables#syntax>

<sup>12</sup> <https://angular.io/guide/router#link-parameters-array>

<sup>13</sup> <https://angular.io/cli/serve>

<sup>14</sup> <https://angular.io/guide/glossary#component>

<sup>15</sup> <https://angular.io/guide/architecture-components#component-metadata>

<sup>16</sup> <https://stackoverflow.com/q/37091718/2740402>

<sup>17</sup> <https://angular.io/guide/binding-syntax>

<sup>18</sup> <https://angular.io/guide/binding-syntax#types-of-data-binding>

<sup>19</sup> <https://angular.io/guide/binding-syntax#binding-types-and-targets>

<sup>20</sup> <https://angular.io/guide/interpolation>

<sup>21</sup> <https://www.telerik.com/blogs/all-about-angular-bindings#string-interpolation-vs-property-binding>

En el contexto de la comunicación de datos entre componentes, para qué se usan las dobles llaves {{ }} en Angular<sup>20, 21</sup>.

Qué es property binding en Angular<sup>22, 23</sup>.

En el contexto de la comunicación de datos entre componentes, para qué se usan los corchetes [] en Angular<sup>22, 23</sup>.

Qué hace el método ngOnInit()<sup>24</sup>.

Qué es binding de atributo, clase y estilo en Angular<sup>25</sup>.

Qué es binding de eventos en Angular<sup>26, 27</sup>.

En el contexto de la comunicación de datos entre componentes, para qué se usan los paréntesis () en Angular<sup>26, 27</sup>.

Qué es el binding bidireccional en Angular<sup>28, 29</sup>.

En el contexto de la comunicación de datos entre componentes, para qué se usan los paréntesis dentro de los corchetes [( )] en Angular<sup>28, 29</sup>.

Qué es el event binding<sup>30</sup>.

En el contexto de la comunicación de datos entre componentes, qué es el nombre de un evento encerrado entre paréntesis<sup>30</sup>.

## Decoradores

Qué son y para qué sirven los decoradores @Input() y @Output() en Angular<sup>31, 32</sup>.

## Directivas

Qué son las directivas en Angular<sup>33</sup>.

Qué es la directiva NgModel en Angular<sup>34, 35</sup>.

Diferencia entre [(ngModel)] y [ngModel] para ligar un estado a una propiedad<sup>36, 37</sup>.

Qué hace la directiva \*ngFor<sup>38, 39</sup>.

Qué es el routing en Angular<sup>40</sup>.

Qué es la directiva routerLink en Angular<sup>41</sup>.

## HTML - CSS

Qué es el tag <div><sup>42, 43</sup>.

Qué es un selector de clase en CSS<sup>44</sup>.

Qué significa la notación .nombre {...} en CSS<sup>44</sup>.

Cómo se aplica una clase definida en CSS a un tag de HTML<sup>44</sup>.

---

<sup>22</sup> <https://angular.io/guide/property-binding>

<sup>23</sup> <https://www.telerik.com/blogs/all-about-angular-bindings#property-binding>

<sup>24</sup> <https://angular.io/api/core/OnInit>

<sup>25</sup> <https://angular.io/guide/attribute-binding>

<sup>26</sup> <https://angular.io/guide/event-binding>

<sup>27</sup> <https://www.telerik.com/blogs/all-about-angular-bindings#event-binding>

<sup>28</sup> <https://angular.io/guide/two-way-binding>

<sup>29</sup> <https://www.telerik.com/blogs/all-about-angular-bindings#two-way-binding>

<sup>30</sup> <https://angular.io/guide/event-binding>

<sup>31</sup> <https://angular.io/guide/inputs-outputs>

<sup>32</sup> [https://youtu.be/57Z\\_U6vPII8?t=97](https://youtu.be/57Z_U6vPII8?t=97)

<sup>33</sup> <https://angular.io/guide/built-in-directives>

<sup>34</sup> <https://angular.io/api/forms/NgModel#description>

<sup>35</sup> <https://www.acontracorrientech.com/angular-entendiendo-la-ngmodel-directive/>

<sup>36</sup> <https://stackoverflow.com/questions/42504918/difference-between-ngmodel-and-ngmodel-for-binding-state-to-property>

<sup>37</sup> <https://blog.thoughttram.io/angular/2016/10/13/two-way-data-binding-in-angular-2.html>

<sup>38</sup> <https://angular.io/api/common/NgFor>

<sup>39</sup> <https://angular.io/guide/built-in-directives#ngFor>

<sup>40</sup> <https://angular.io/guide/routing-overview>

<sup>41</sup> <https://angular.io/api/router/RouterLink>

<sup>42</sup> [https://www.w3schools.com/tags/tag\\_div.ASP](https://www.w3schools.com/tags/tag_div.ASP)

<sup>43</sup> <https://developer.mozilla.org/es/docs/Web/HTML/Element/div>

<sup>44</sup> [https://www.w3schools.com/cssref/sel\\_class.asp](https://www.w3schools.com/cssref/sel_class.asp)

Qué es el CSS box model (modelo de cajas de CSS)<sup>45, 46</sup>.

Explicar la diferencia entre padding y margin en CSS<sup>45</sup>.

Qué significa que una aplicación es responsiva<sup>47</sup>.

## Procedimientos

Cómo se genera una nueva aplicación en Angular<sup>48</sup>.

Como se usan los componentes y templates en Angular<sup>49</sup>.

Cómo se genera un componente nuevo en Angular<sup>50</sup>.

Qué hace el comando `ng g c nombre`<sup>50</sup>.

Cómo se usa data binding en HTML en Angular<sup>51</sup>.

Qué son las anotaciones de tipo de parámetros de una función<sup>52</sup>.

Qué son las anotaciones de tipo de retorno de una función<sup>53</sup>.

Qué es la asignación contextual de tipos de parámetros en las funciones anónimas<sup>54</sup>.

Cómo y dónde se especifica el punto de entrada de la aplicación, es decir por dónde arranca<sup>55</sup>.

Qué es Angular Material<sup>56</sup>.

Qué es el operador de navegación segura, u operador de encadenamiento opcional en JavaScript<sup>57</sup>.

## Tecnologías

Qué es RxJS<sup>58, 59</sup>.

Qué hace la extensión vscode-icons de Visual Studio Code<sup>60</sup>.

Qué hace la extensión ESLint de Visual Studio Code<sup>61</sup>.

Qué significa que una aplicación sea responsiva (responsive<sup>62</sup> en inglés).

## TypeScript

Qué es TypeScript<sup>63, 64, 65</sup>.

Explicar los tipos primitivos de datos string, number y boolean<sup>66</sup>.

Explicar los tipos Object<sup>67</sup>.

Explicar cómo se especifica el tipo array<sup>68</sup>.

Explicar el tipo any<sup>69</sup>.

---

<sup>45</sup> [https://www.w3schools.com/css/css\\_boxmodel.asp](https://www.w3schools.com/css/css_boxmodel.asp)

<sup>46</sup> [https://developer.mozilla.org/es/docs/Learn/CSS/Building\\_blocks/The\\_box\\_model%C2%BFqu%C3%A9\\_es\\_el\\_modelo\\_de\\_cajas\\_css](https://developer.mozilla.org/es/docs/Learn/CSS/Building_blocks/The_box_model%C2%BFqu%C3%A9_es_el_modelo_de_cajas_css)

<sup>47</sup> [https://es.wikipedia.org/wiki/Dise%C3%B1o\\_web\\_adaptable](https://es.wikipedia.org/wiki/Dise%C3%B1o_web_adaptable)

<sup>48</sup> <https://youtu.be/R90EqA-O37U>

<sup>49</sup> <https://angular.io/guide/architecture-components>

<sup>50</sup> <https://angular.io/cli/generate#component-command>

<sup>51</sup> <https://angular.io/guide/binding-syntax#data-binding-and-html>

<sup>52</sup> <https://www.typescriptlang.org/docs/handbook/2/everyday-types.html#parameter-type-annotations>

<sup>53</sup> <https://www.typescriptlang.org/docs/handbook/2/everyday-types.html#return-type-annotations>

<sup>54</sup> <https://www.typescriptlang.org/docs/handbook/2/everyday-types.html#anonymous-functions>

<sup>55</sup> <https://angular.io/guide/bootstrapping>

<sup>56</sup> <https://material.angular.io/>

<sup>57</sup> [https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Operators/Optional\\_chaining](https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Operators/Optional_chaining)

<sup>58</sup> <https://rxjs.dev/>

<sup>59</sup> <https://angular.io/guide/rx-library>

<sup>60</sup> <https://youtu.be/R90EqA-O37U?t=202>

<sup>61</sup> <https://youtu.be/R90EqA-O37U?t=267>

<sup>62</sup> [https://en.wikipedia.org/wiki/Responsive\\_web\\_design](https://en.wikipedia.org/wiki/Responsive_web_design)

<sup>63</sup> <https://www.typescriptlang.org/>

<sup>64</sup> <https://en.wikipedia.org/wiki/TypeScript>

<sup>65</sup> <https://profile.es/blog/que-es-typescript-vs-javascript/>

<sup>66</sup> <https://www.typescriptlang.org/docs/handbook/2/everyday-types.html#the-primitives-string-number-and-boolean>

<sup>67</sup> <https://www.typescriptlang.org/docs/handbook/2/everyday-types.html#object-types>

<sup>68</sup> <https://www.typescriptlang.org/docs/handbook/2/everyday-types.html#arrays>

<sup>69</sup> <https://www.typescriptlang.org/docs/handbook/2/everyday-types.html#any>

Qué son los tipos de unión<sup>70</sup>.

Explicar el flag de compilador noImplicitAny<sup>71</sup>.

Explicar las anotaciones de tipo en variables<sup>72</sup>.

Qué son las funciones en JavaScript<sup>73, 74, 75, 76, 77</sup>.

Qué es una propiedad opcional de un objeto y cómo se especifica<sup>78</sup>.

Qué es el servicio Title de Angular<sup>79</sup>.

---

<sup>70</sup> <https://www.typescriptlang.org/docs/handbook/2/everyday-types.html#union-types>

<sup>71</sup> <https://www.typescriptlang.org/docs/handbook/2/everyday-types.html#noimplicitany>

<sup>72</sup> <https://www.typescriptlang.org/docs/handbook/2/everyday-types.html#type-annotations-on-variables>

<sup>73</sup> [https://www.w3schools.com/js/js\\_functions.asp](https://www.w3schools.com/js/js_functions.asp)

<sup>74</sup> <https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Functions>

<sup>75</sup> [https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global\\_Objects/Function](https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Function)

<sup>76</sup> <https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Functions>

<sup>77</sup> <https://www.typescriptlang.org/docs/handbook/2/everyday-types.html#functions>

<sup>78</sup> <https://www.typescriptlang.org/docs/handbook/2/everyday-types.html#optional-properties>

<sup>79</sup> <https://angular.io/api/platform-browser/Title>