Preguntas Angular-v-3

Gustavo García, 20 de mayo de 2023.

# Angular como plataforma

1. Quién desarrolla y mantiene Angular[[1]](#footnote-1).
2. Cuál es el principal lenguaje de programación que usa Angular1.
3. Qué es Angular CLI[[2]](#footnote-2).
4. Explicar la diferencia[[3]](#footnote-3) entre Angular[[4]](#footnote-4) y Angularjs[[5]](#footnote-5).
5. Qué hay que instalar en la PC para comenzar a desarrollar una aplicación de frontend en Angular.
6. Qué es Node.js[[6]](#footnote-6).
7. Qué es npm[[7]](#footnote-7).
8. Qué significa SPA[[8]](#footnote-8), [[9]](#footnote-9).

# Arquitectura

1. Cuáles son los patrones de diseño relevantes en Angular[[10]](#footnote-10).
2. Explicar la arquitectura de una aplicación en Angular[[11]](#footnote-11).
3. Qué es un template en Angular[[12]](#footnote-12).
4. Qué son template statements en Angular[[13]](#footnote-13).
5. Qué son las variables de template en Angular[[14]](#footnote-14).
6. Cómo se declara una variable de template en Angular[[15]](#footnote-15).
7. Cómo se referencia una variable de template en Angular11.
8. Qué es ligar un array de parámetros de enlace (link parameters array[[16]](#footnote-16)) a una directiva RouterLink.
9. Qué es y qué contiene un array de parámetros de link (link parameters array)12.
10. Que hace el siguiente tag12: <a [[routerLink](https://angular.io/api/router/RouterLink)]="['/path']">Some comment</a>

# Comandos

1. Qué hace el comando ng serve y cuáles son sus opciones[[17]](#footnote-17).

# Componentes

1. Qué archivos forman el componente generado en Angular.
2. Qué es un componente en Angular[[18]](#footnote-18).
3. Qué es el selector de un componente[[19]](#footnote-19), [[20]](#footnote-20).
4. Qué ventajas tiene esta arquitectura de Angular centrada en componentes formados por tres archivos dentro de una carpeta.

# Binding

1. Qué hace el data binding en Angular[[21]](#footnote-21).
2. Qué tipos de data binding existen en Angular[[22]](#footnote-22).
3. Explicar la relación entre los tipos de data binding y sus targets en Angular[[23]](#footnote-23).
4. Qué es interpolación de expresiones en Angular, y por qué se llama así[[24]](#footnote-24), [[25]](#footnote-25).
5. En el contexto de la comunicación de datos entre componentes, para qué se usan las dobles llaves {{   }} en Angular20, 21.
6. Qué es property binding en Angular[[26]](#footnote-26), [[27]](#footnote-27).
7. En el contexto de la comunicación de datos entre componentes, para qué se usan los corchetes [] en Angular22, 23.
8. Qué hace el método ngOnInit()[[28]](#footnote-28).
9. Diferencia entre el constructor y ngOnInit[[29]](#footnote-29).
10. Qué es binding de atributo, clase y estilo en Angular[[30]](#footnote-30).
11. Qué es binding de eventos en Angular[[31]](#footnote-31),[[32]](#footnote-32).
12. En el contexto de la comunicación de datos entre componentes, para qué se usan los paréntesis () en Angular26, 27.
13. Qué es el binding bidireccional en Angular[[33]](#footnote-33),[[34]](#footnote-34).
14. En el contexto de la comunicación de datos entre componentes, para qué se usan los paréntesis dentro de los corchetes [(   )] en Angular[[35]](#footnote-35), [[36]](#footnote-36).
15. Qué es el event binding[[37]](#footnote-37).
16. En el contexto de la comunicación de datos entre componentes, qué es el nombre de un evento encerrado entre paréntesis30.

# Decoradores

1. Qué son y para qué sirven los decoradores @[Input](https://angular.io/api/core/Input)() y @[Output](https://angular.io/api/core/Output)() en Angular[[38]](#footnote-38), [[39]](#footnote-39).

# Directivas

1. Qué son las directivas en Angular[[40]](#footnote-40).
2. Qué es la directiva NgModel en Angular[[41]](#footnote-41), [[42]](#footnote-42).
3. Diferencia entre [(ngModel)] y [ngModel] para ligar un estado a una propiedad[[43]](#footnote-43), [[44]](#footnote-44).
4. Qué hace la directiva \*ngFor[[45]](#footnote-45), [[46]](#footnote-46).
5. Qué es el routing en Angular[[47]](#footnote-47).
6. Qué es la directiva routerLink en Angular[[48]](#footnote-48).
7. Qué es la directiva <router-outlet></router-outlet> en Angular[[49]](#footnote-49).

# HTML - CSS

1. Qué es el tag <div>[[50]](#footnote-50), [[51]](#footnote-51).
2. Qué es un selector de clase en CSS[[52]](#footnote-52).
3. Qué significa la notación .nombre {…} en CSS44.
4. Cómo se aplica una clase definida en CSS a un tag de HTML44.
5. Qué es el CSS box model (modelo de cajas de CSS)[[53]](#footnote-53), [[54]](#footnote-54).
6. Explicar la diferencia entre padding y margin en CSS45.
7. Qué significa que una aplicación es responsiva[[55]](#footnote-55).

# Procedimientos

1. Cómo se genera una nueva aplicación en Angular[[56]](#footnote-56).
2. Como se usan los componentes y templates en Angular[[57]](#footnote-57).
3. Cómo se genera un componente nuevo en Angular[[58]](#footnote-58).
4. Qué hace el comando ng g c nombre50.
5. Cómo se usa data binding en HTML en Angular[[59]](#footnote-59).
6. Qué son las anotaciones de tipo de parámetros de una función[[60]](#footnote-60).
7. Qué son las anotaciones de tipo de retorno de una función [[61]](#footnote-61).
8. Qué es la asignación contextual de tipos de parámetros en las funciones anónimas[[62]](#footnote-62).
9. Cómo y dónde se especifica el punto de entrada de la aplicación, es decir por dónde arranca[[63]](#footnote-63).
10. Qué es Angular Material[[64]](#footnote-64).
11. Qué es el operador de navegación segura, u operador de encadenamiento opcional en JavaScript[[65]](#footnote-65).

# Tecnologías

1. Qué es RxJS[[66]](#footnote-66), [[67]](#footnote-67).
2. Qué hace la extensión vscode-icons de Visual Studio Code[[68]](#footnote-68).
3. Qué hace la extensión ESLint de Visual Studio Code[[69]](#footnote-69).
4. Qué significa que una aplicación sea responsiva (responsive[[70]](#footnote-70) en inglés).

# TypeScript

1. Qué es TypeScript[[71]](#footnote-71), [[72]](#footnote-72), [[73]](#footnote-73).
2. Explicar los tipos primitivos de datos string, number y boolean[[74]](#footnote-74).
3. Explicar los tipos Object[[75]](#footnote-75).
4. Explicar cómo se especifica el tipo array[[76]](#footnote-76).
5. Explicar el tipo any[[77]](#footnote-77).
6. Qué son los tipos de unión[[78]](#footnote-78).
7. Explicar el flag de compilador noImplicitAny[[79]](#footnote-79).
8. Explicar las anotaciones de tipo en variables[[80]](#footnote-80).
9. Qué son las funciones en JavaScript[[81]](#footnote-81), [[82]](#footnote-82), [[83]](#footnote-83), [[84]](#footnote-84), [[85]](#footnote-85).
10. Qué es una propiedad opcional de un objeto y cómo se especifica[[86]](#footnote-86).
11. Qué es el servicio Title de Angular[[87]](#footnote-87).

1. <https://es.wikipedia.org/wiki/Angular_(framework)> [↑](#footnote-ref-1)
2. <https://docs.angular.lat/cli> [↑](#footnote-ref-2)
3. <https://www.campusmvp.es/recursos/post/las-10-principales-diferencias-entre-angularjs-y-angular.aspx> [↑](#footnote-ref-3)
4. <https://angular.io> [↑](#footnote-ref-4)
5. <https://angularjs.org/> [↑](#footnote-ref-5)
6. <https://es.wikipedia.org/wiki/Node.js> [↑](#footnote-ref-6)
7. <https://es.wikipedia.org/wiki/Npm> [↑](#footnote-ref-7)
8. <https://developer.mozilla.org/en-US/docs/Glossary/SPA> [↑](#footnote-ref-8)
9. <https://es.wikipedia.org/wiki/Single-page_application> [↑](#footnote-ref-9)
10. <https://angular-enterprise.com/en/ngpost/courses/design-patterns/> [↑](#footnote-ref-10)
11. <https://angular.io/guide/architecture> [↑](#footnote-ref-11)
12. <https://angular.io/guide/glossary#template> [↑](#footnote-ref-12)
13. <https://angular.io/guide/template-statements#template-statements> [↑](#footnote-ref-13)
14. <https://angular.io/guide/template-reference-variables#understanding-template-variables> [↑](#footnote-ref-14)
15. <https://angular.io/guide/template-reference-variables#syntax> [↑](#footnote-ref-15)
16. <https://angular.io/guide/router#link-parameters-array> [↑](#footnote-ref-16)
17. <https://angular.io/cli/serve> [↑](#footnote-ref-17)
18. <https://angular.io/guide/glossary#component> [↑](#footnote-ref-18)
19. <https://angular.io/guide/architecture-components#component-metadata> [↑](#footnote-ref-19)
20. <https://stackoverflow.com/q/37091718/2740402> [↑](#footnote-ref-20)
21. <https://angular.io/guide/binding-syntax> [↑](#footnote-ref-21)
22. <https://angular.io/guide/binding-syntax#types-of-data-binding> [↑](#footnote-ref-22)
23. <https://angular.io/guide/binding-syntax#binding-types-and-targets> [↑](#footnote-ref-23)
24. <https://angular.io/guide/interpolation> [↑](#footnote-ref-24)
25. <https://www.telerik.com/blogs/all-about-angular-bindings#string-interpolation-vs-property-binding> [↑](#footnote-ref-25)
26. <https://angular.io/guide/property-binding> [↑](#footnote-ref-26)
27. <https://www.telerik.com/blogs/all-about-angular-bindings#property-binding> [↑](#footnote-ref-27)
28. <https://angular.io/api/core/OnInit> [↑](#footnote-ref-28)
29. <https://stackoverflow.com/questions/35763730/difference-between-constructor-and-ngoninit> [↑](#footnote-ref-29)
30. <https://angular.io/guide/attribute-binding> [↑](#footnote-ref-30)
31. <https://angular.io/guide/event-binding> [↑](#footnote-ref-31)
32. <https://www.telerik.com/blogs/all-about-angular-bindings#event-binding> [↑](#footnote-ref-32)
33. <https://angular.io/guide/two-way-binding> [↑](#footnote-ref-33)
34. <https://www.telerik.com/blogs/all-about-angular-bindings#two-way-binding> [↑](#footnote-ref-34)
35. <https://docs.angular.lat/guide/two-way-binding> [↑](#footnote-ref-35)
36. <https://angular.io/guide/two-way-binding> [↑](#footnote-ref-36)
37. <https://angular.io/guide/event-binding> [↑](#footnote-ref-37)
38. <https://angular.io/guide/inputs-outputs> [↑](#footnote-ref-38)
39. <https://youtu.be/57Z_U6vPIl8?t=97> [↑](#footnote-ref-39)
40. <https://angular.io/guide/built-in-directives> [↑](#footnote-ref-40)
41. <https://angular.io/api/forms/NgModel#description> [↑](#footnote-ref-41)
42. <https://www.acontracorrientech.com/angular-entendiendo-la-ngmodel-directive/> [↑](#footnote-ref-42)
43. <https://stackoverflow.com/questions/42504918/difference-between-ngmodel-and-ngmodel-for-binding-state-to-property> [↑](#footnote-ref-43)
44. <https://blog.thoughtram.io/angular/2016/10/13/two-way-data-binding-in-angular-2.html> [↑](#footnote-ref-44)
45. <https://angular.io/api/common/NgFor> [↑](#footnote-ref-45)
46. <https://angular.io/guide/built-in-directives#ngFor> [↑](#footnote-ref-46)
47. <https://angular.io/guide/routing-overview> [↑](#footnote-ref-47)
48. <https://angular.io/api/router/RouterLink> [↑](#footnote-ref-48)
49. <https://angular.io/api/router/RouterOutlet> [↑](#footnote-ref-49)
50. <https://www.w3schools.com/tags/tag_div.ASP> [↑](#footnote-ref-50)
51. <https://developer.mozilla.org/es/docs/Web/HTML/Element/div> [↑](#footnote-ref-51)
52. <https://www.w3schools.com/cssref/sel_class.asp> [↑](#footnote-ref-52)
53. <https://www.w3schools.com/css/css_boxmodel.asp> [↑](#footnote-ref-53)
54. <https://developer.mozilla.org/es/docs/Learn/CSS/Building_blocks/The_box_model#%C2%BFqu%C3%A9_es_el_modelo_de_cajas_css> [↑](#footnote-ref-54)
55. <https://es.wikipedia.org/wiki/Dise%C3%B1o_web_adaptable> [↑](#footnote-ref-55)
56. <https://youtu.be/R90EqA-O37U> [↑](#footnote-ref-56)
57. <https://angular.io/guide/architecture-components> [↑](#footnote-ref-57)
58. <https://angular.io/cli/generate#component-command> [↑](#footnote-ref-58)
59. <https://angular.io/guide/binding-syntax#data-binding-and-html> [↑](#footnote-ref-59)
60. <https://www.typescriptlang.org/docs/handbook/2/everyday-types.html#parameter-type-annotations> [↑](#footnote-ref-60)
61. <https://www.typescriptlang.org/docs/handbook/2/everyday-types.html#return-type-annotations> [↑](#footnote-ref-61)
62. <https://www.typescriptlang.org/docs/handbook/2/everyday-types.html#anonymous-functions> [↑](#footnote-ref-62)
63. <https://angular.io/guide/bootstrapping> [↑](#footnote-ref-63)
64. <https://material.angular.io/> [↑](#footnote-ref-64)
65. <https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Operators/Optional_chaining> [↑](#footnote-ref-65)
66. <https://rxjs.dev/> [↑](#footnote-ref-66)
67. <https://angular.io/guide/rx-library> [↑](#footnote-ref-67)
68. <https://youtu.be/R90EqA-O37U?t=202> [↑](#footnote-ref-68)
69. <https://youtu.be/R90EqA-O37U?t=267> [↑](#footnote-ref-69)
70. <https://en.wikipedia.org/wiki/Responsive_web_design> [↑](#footnote-ref-70)
71. <https://www.typescriptlang.org/> [↑](#footnote-ref-71)
72. <https://en.wikipedia.org/wiki/TypeScript> [↑](#footnote-ref-72)
73. <https://profile.es/blog/que-es-typescript-vs-javascript/> [↑](#footnote-ref-73)
74. <https://www.typescriptlang.org/docs/handbook/2/everyday-types.html#the-primitives-string-number-and-boolean> [↑](#footnote-ref-74)
75. <https://www.typescriptlang.org/docs/handbook/2/everyday-types.html#object-types> [↑](#footnote-ref-75)
76. <https://www.typescriptlang.org/docs/handbook/2/everyday-types.html#arrays> [↑](#footnote-ref-76)
77. <https://www.typescriptlang.org/docs/handbook/2/everyday-types.html#any> [↑](#footnote-ref-77)
78. <https://www.typescriptlang.org/docs/handbook/2/everyday-types.html#union-types> [↑](#footnote-ref-78)
79. <https://www.typescriptlang.org/docs/handbook/2/everyday-types.html#noimplicitany> [↑](#footnote-ref-79)
80. <https://www.typescriptlang.org/docs/handbook/2/everyday-types.html#type-annotations-on-variables> [↑](#footnote-ref-80)
81. <https://www.w3schools.com/js/js_functions.asp> [↑](#footnote-ref-81)
82. <https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Functions> [↑](#footnote-ref-82)
83. <https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Function> [↑](#footnote-ref-83)
84. <https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Functions> [↑](#footnote-ref-84)
85. <https://www.typescriptlang.org/docs/handbook/2/everyday-types.html#functions> [↑](#footnote-ref-85)
86. <https://www.typescriptlang.org/docs/handbook/2/everyday-types.html#optional-properties> [↑](#footnote-ref-86)
87. <https://angular.io/api/platform-browser/Title> [↑](#footnote-ref-87)