Angular

# Angular y React

React es como Angular, pero "more minimal", en palabras de la documentación oficial. Hay una cantidad de reviews que los comparan. Todos son sorprendentemente largos y minuciosos. La idea que tengo, por ahora, es que son realmente distintos. Angular es un framework, React es una biblioteca. Parece que Angular es más OOP que React. Fuera de eso, desde cada uno de los numerosos puntos de vista en los que se puede enfocar el análisis, ambos tienen pros y contras. En gran medida, es una cuestión de gustos, o inclinaciones personales.

# Using Angular in Visual Studio Code

<https://code.visualstudio.com/docs/nodejs/angular-tutorial>

Sale todo perfecto.

node -v o node --version

v16.14.2

npm -v o npm --version

8.7.0

code -v o code –version

1.65.2

c722ca6c7eed3d7987c0d5c3df5c45f6b15e77d1

x64

# Instalar Angular Cli

Open PowerShell or Console in Admin Mode.

npm uninstall -g angular-cli

npm uninstall -g @angular-devkit/architect

npm uninstall -g @angular-devkit/core

npm uninstall -g @angular-devkit/schematics

npm uninstall -g @schematics/angular

npm uninstall angular-cli

npm uninstall @angular-devkit/architect

npm uninstall @angular-devkit/core

npm uninstall @angular-devkit/schematics

npm uninstall @schematics/angular

npm cache verify

npm cache clean --force

npm install -g @angular/cli@latest

ng --version

\_ \_ \_\_\_\_ \_ \_\_\_

/ \ \_ \_\_ \_\_ \_ \_ \_| | \_\_ \_ \_ \_\_ / \_\_\_| | |\_ \_|

/ △ \ | '\_ \ / \_` | | | | |/ \_` | '\_\_| | | | | | |

/ \_\_\_ \| | | | (\_| | |\_| | | (\_| | | | |\_\_\_| |\_\_\_ | |

/\_/ \\_\\_| |\_|\\_\_, |\\_\_,\_|\_|\\_\_,\_|\_| \\_\_\_\_|\_\_\_\_\_|\_\_\_|

|\_\_\_/

Angular CLI: 13.3.4

Node: 16.14.2

Package Manager: npm 8.7.0

OS: win32 x64

Angular:

...

Package Version

------------------------------------------------------

@angular-devkit/architect 0.1303.4 (cli-only)

@angular-devkit/core 13.3.4 (cli-only)

@angular-devkit/schematics 13.3.4 (cli-only)

@schematics/angular 13.3.4 (cli-only)

# [Global Angular CLI version greater than local version](https://stackoverflow.com/questions/44525746/global-angular-cli-version-greater-than-local-version)

Si aparece un mensaje como:

[Your global Angular CLI version is greater than your local version. To disable this warning use "ng config -g cli.warnings.versionMismatch false"](https://stackoverflow.com/questions/57611447/your-global-angular-cli-version-is-greater-than-your-local-version-to-disable-t)

usar la solución de esta respuesta de StackOverflow[[1]](#footnote-1):

npm install --save-dev @angular/cli@latest -g

Esa respuesta, además, explica por qué aparece el mensaje en cuestión.

# Curso de Angular desde cero - Instalaciones necesarias 2022 - #1

<https://youtu.be/csTZcCAE8aw>

11 minutos.

Spoiler alert: los tutoriales son excelentes, pero este muchacho pronuncia mal muchas palabras del inglés. You've been warned.

Desde cargar Visual Studio Code. [https://code.visualstudio.com/](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqbkl6dnAxcU9qY3UzNEZTVmZPTlZiSVI5N21sQXxBQ3Jtc0traXVYMTJNMWVDcVYtNFZfQVpkbl84MkV0UjJPaU1TMGV6RmJwR3VwRTd1NllKd0Jyb1JZSUZ3TkttUmZMTkYyalJZWmxTOGZQUkNPRy03ZDNpZjlVZmt3cDUzLXpNRHVZdmJZODlranZpSXVsckEycw&q=https%3A%2F%2Fcode.visualstudio.com%2F)

1º Instalar NodeJs: [https://nodejs.org/es/](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqbFp4X3hmajBGTHpsMC16emlJVFdMSlR3TGt1QXxBQ3Jtc0tsWVQ0dTM1LXhrc2dQY2Z1MDRCQ3pJTEVLcE1qZzJWZGduWUdqOExvSmdiYkZsLTV4cDl6eExFRmxLc0p5dnBQUVBQNG5GQl9xZkZPTW9LemhCT3V1MFE1NFRjZDMzZjVjVEZHMEs2YV9jR3lxaE8tTQ&q=https%3A%2F%2Fnodejs.org%2Fes%2F)

2º Instalar Angular Cli: [https://angular.io/cli](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqbUJVSzBVWW9HTDBLOGEtSkQ2aHFvaGVWNkFmZ3xBQ3Jtc0tuTWdDUFpxUmExVjJySnR4ZFNUMXd3bUtjcGxRdGtZY0hxaGRDN3luZVRFUWxhcDRIRk5QV2NfUU5DTW83Y2R1aDBnRDBMTkt1dHd4bXBuazVOUmRXQmVBdlF1YmtlNTZFTmpNSEpXR0pCMk53ZHM1Zw&q=https%3A%2F%2Fangular.io%2Fcli)

npm install -g @angular/cli

## Crear nueva aplicación proyecto my-app

El proyecto para este tutorial está en:

20-que-es-angular\my-app\

ng new my-app

cd my-app

ng serve -–o

El flag -o es sinónimo de --o y de --open. Es para que la aplicación se abra automáticamente en el browser. Para cortar el servicio desde Git Bash, en la misma ventana en la que arrancamos la aplicación con ng serve, ingresamos Ctrl + c. Vemos el prompt del sistema, lo que nos confirma que el servicio fue terminado.

Si abrimos en el browser la URL:

<http://localhost:4200/>

veremos la aplicación.

Para limpiar el puerto 4200, ejecutar en la consola de Windows el comando:

for /f "tokens=5" %a in ('netstat -aon ^| find /i "LISTENING" ^| find ":4200"') do taskkill /f /pid %a

# Curso de Angular - Primer Proyecto - Plugin - 2022 - #2

<https://www.youtube.com/watch?v=R90EqA-O37U&ab_channel=YouTics>

En esta segunda mini clase creamos nuestro primer proyecto, lo ejecutamos en un servidor local, generamos algunos cambios y además instalamos algunos plugin: vscode-icons y ESLint

OJO: al crear el proyecto ponerle que SÍ queremos el router. Él dice que no, pero después sí lo usa.

## Crear proyecto MiPrimeraApp

El proyecto para este tutorial está en:

C:\GIT\20-que-es-angular\MiPrimeraApp\

Si al crear el proyecto nos sale un mensaje como:

Directory is already under version control. Skipping initialization of git.

eso se debe a que la carpeta que contiene a la carpeta del proyecto ya está inicializada como repositorio de Git. Ir a la carpeta raíz del repositorio y pegar el comando:

git add --all

para que agregue todo el proyecto que acabamos de crear, carpetas y subcarpetas.

# Cambiar el título del documento HTML

No se trata de inyectar una variable en un tag <h1>...</h1>, sino de cambiar en run time el tag <title> dentro de la sección <head> de index.html, programáticamente. Mr voy a basar en el siguiente artículo:

<https://www.tektutorialshub.com/angular/set-page-title-using-title-service-angular-example/>

Comienzo por crear una nueva aplicación, sobre la cual haré luego los cambios.

En Git Bash navego hasta la carpeta GIT\20-que-es-angular. Pego el comando:

ng new title

Agrego routing. Elijo SCSS. Tarda un poco.

cd title

code .

En la carpeta GIT\20-que-es-angular\title está el proyecto completo. Los comentarios explican todo detalladamente.

# Curso de Angular Crear Componentes + CSS - 2022 - #3

<https://www.youtube.com/watch?v=lW_zfBm5t4A&ab_channel=YouTics>

Seguimos avanzando desde cero en la programación web utilizando el framework de Angular. En este capítulo aprenderemos a crear un componente, llamar ese componente dentro de otro utilizando un selector. También aprenderemos a crear clases de css y ver como aplicarlas dentro de un html.

Lo más importante que tenemos que aprender, al terminar este capítulo, es cómo se arma el frontend, una aplicación completa de Angular, a partir de los componentes.

Creamos el proyecto para este tutorial. Necesitamos una ventana de Git Bash en la carpeta GIT\20-que-es-angular. Pegamos el comando:

git new tutorial-3

y seguimos el procedimiento que ya sabemos.

Cuando termina de generar el proyecto, pegamos el comando:

git all --all

Si ahora hacemos

git status

nos va a decir que estamos en la rama main, y tenemos un montón de cambios staged, esperando un commit.

# CSS box model

La explicación de border y padding (palabra que nuestro superhéroe mispronounces[[2]](#footnote-2)) es confusa. La siguiente página da una explicación simple, clara y correcta:

<https://www.w3schools.com/css/css_boxmodel.asp>

Las unidades de medida que se usan en CSS están explicadas en la siguiente página:

<https://www.w3schools.com/cssref/css_units.asp>

El siguiente artículo de la Wikipedia resume todo el tema:

<https://en.wikipedia.org/wiki/CSS_box_model>

The Cascading Style Sheets (CSS) specification describes how elements of [web pages](https://en.wikipedia.org/wiki/Web_page) are displayed by graphical browsers. Section 4 of the CSS1 specification defines a "formatting model" that gives block-level elements—such as p and blockquote—a width and height, and three levels of boxes surrounding it: padding, borders, and margins.[[4]](https://en.wikipedia.org/wiki/CSS_box_model#cite_note-CSS-19990111-4) While the specification never uses the term "box model" explicitly, the term has become widely used by web developers and web browser vendors.

All HTML elements can be considered "boxes", this includes div tag, p tag, or a tag. Each of those boxes has five modifiable dimensions:

* the height and width describe dimensions of the actual content of the box (text, images, ...)
* the padding describes the space between this content and the border of the box
* the border is any kind of line (solid, dotted, dashed...) surrounding the box, if present
* the margin is the space around the border

According to the CSS1 specification, released by W3C in 1996 and revised in 1999, when a width or height is explicitly specified for any block-level element, it should determine only the width or height of the visible element, with the padding, borders, and margins applied afterward. Before CSS3, this box model was known as **W3C box model**, in CSS3, it is known as the content-box.

The total width of a box is therefore left-margin + left-border + left-padding + width + right-padding + right-border + right-margin. Similarly, the total height of a box equals top-margin + top-border + top-padding + height + bottom-padding + bottom-border + bottom-margin.

For example, the following CSS code

.**myClass** {

**width**: 200px;

**height**: 100px;

**padding**: 10px;

**border**: **solid** 10px **black**;

**margin**: 10px;

}

would specify the box dimensions of each block belonging to 'myClass'. Moreover, each such box will have total height 160[px](https://en.wikipedia.org/wiki/Pixel) and width 260px.

CSS3 introduced the **Internet Explorer box model** to the standard, known referred to as border-box.

# Curso Angular - Data binding interface - array - \*ngFor - 2022 - #4

<https://youtu.be/rOsd3OI9dOA>

**Seguimos avanzando desde cero en la programación web utilizando el framework de Angular. En este capítulo** seguiremos comunicando entre componentes y aprenderemos a crear interface, arrays, Property data binding y la directiva \*ngFor. También seguiremos analizando las propiedades de css e implementadolas en nuestro proyecto.

## Listado de elementos con la directiva NgFor

Utilice la directiva NgFor para presentar una lista de elementos:

1. Defina un bloque de HTML que determine cómo Angular representa un solo elemento.
2. Para enumerar sus elementos, asigne la letra abreviada let elemento de elementos a \*ngFor.

Por ejemplo:

<div \*ngFor="let item of items">{{item.name}}</div>

La cadena "let item of items" le indica a Angular que:

* Almacene cada elemento del array items en la variable de bucle local item.
* Haga que cada item esté disponible para el HTML con plantilla para cada iteración
* Traduzca "let item of items" en una <ng-template> alrededor del elemento host. Esto se hace detrás de escena.
* Repita la <ng-template> para cada elemento de la lista

Para obtener más información, consulte la [sección abreviada de directivas estructurales](https://angular.io/guide/structural-directives#shorthand) de [Directivas estructurales](https://angular.io/guide/structural-directives).

## Repetir una vista de componente

Para repetir un elemento componente, aplique \*ngFor al selector. En el siguiente ejemplo, el selector es:

<app-item-detail>.

Recordar que el selector lo encontramos en la clase que define el componente, el archivo:

item-detail.component.ts.

<app-item-detail \*ngFor="let item of items" [item]="item"></app-item-detail>

Haga referencia a una variable de entrada de plantilla, como item, en las siguientes ubicaciones:

* Dentro del elemento host ngFor.
* Dentro de los descendientes del elemento host, para acceder a las propiedades de item.

El siguiente ejemplo hace referencia a item en una interpolación:

<div \*ngFor="let item of items">{{item.name}}</div>

El siguiente ejemplo pasa item como binding a la propiedad item del componente <app-item-detail>:

<app-item-detail \*ngFor="let item of items" [item]="item"></app-item-detail>

# Curso de Angular - Clonar e Instalar un Proyecto de Angular - Git + GitHub - 2022 - #5

<https://youtu.be/x3nFRQtauEM>

En este capitulo aprenderemos lo básico para guardar un proyecto en la nube y como recuperar ese proyecto con todos los archivos que necesita angular para trabajar.

# Curso de Angular Habilitar Terminal de Windows en Visual Studio Code - 2022 - #6

<https://www.youtube.com/watch?v=4aqA8Ny090o&t=1s&ab_channel=YouTics>

En este capitulo contestaremos a una duda importante acerca de cómo habilitar una terminal en Windows.

## PowerShell's execution policy[[3]](#footnote-3)

La política de ejecución de PowerShell es una característica de seguridad que controla las condiciones bajo las cuales PowerShell carga archivos de configuración y ejecuta scripts. Esta característica ayuda a prevenir la ejecución de scripts maliciosos. En una computadora con Windows, puede establecer una política de ejecución para la computadora local, para el usuario actual o para una sesión en particular. También puede usar una configuración de Política de grupo para establecer políticas de ejecución para computadoras y usuarios. Las políticas de ejecución para la computadora local y el usuario actual se almacenan en el registro. No necesita establecer políticas de ejecución en su perfil de PowerShell. La política de ejecución para una sesión en particular se almacena solo en la memoria y se pierde cuando se cierra la sesión. La política de ejecución no es un sistema de seguridad que restrinja las acciones del usuario. Por ejemplo, los usuarios pueden omitir fácilmente una política escribiendo el contenido del script en la línea de comando cuando no pueden ejecutar un script. En cambio, la política de ejecución ayuda a los usuarios a establecer reglas básicas y evita que las violen sin querer. En equipos que no son Windows, la política de ejecución predeterminada es Unrestricted y no se puede cambiar. El cmdlet Set-ExecutionPolicy está disponible, pero PowerShell muestra un mensaje de consola que indica que no es compatible. Si bien Get-ExecutionPolicy devuelve Unrestricted en plataformas que no son de Windows, el comportamiento realmente coincide con Bypass porque esas plataformas no implementan las Zonas de seguridad de Windows.

## Para configurar

Un modo es cerrar todas las ventanas de VS Code que tengamos abiertas; ejecutar VS Code como Administrador, en VS Code abrir una nueva terminal.

Otro modo es el que recomienda el tutorial.

En PowerShell pegar el comando:

PS C:\Users\Gustavo> Get-ExecutionPolicy

Restricted

Esto es lo que hay que cambiar. Para eso, pegar el comando:

Set-ExecutionPolicy Unrestricted

# Curso de Angular - @Input - 2022 - #7

<https://youtu.be/57Z_U6vPIl8>

En este capitulo aprenderemos a comunicar información entre componentes utilizando el decorador @Input.

# Curso de Angular | Databinding - Interpolación - #8

<https://www.youtube.com/watch?v=13jsC5nLntE&ab_channel=YouTics>

En este capitulo del curso de angular abordaremos con mas profundidad el concepto de Databinding con ejemplos de interpolación.

La interpolación es one-way binding, parte del ts y se dirige al html.

Property binding es one-way binding, parte del ts y se dirige al html.

Event binding es one-way binding, parte del html y se dirige al ts.

## BootStrap

Vamos a ver que en este tutorial usamos al framework BootStrap, que es muy popular. El sitio oficial es:

<https://getbootstrap.com/>

En esa página, vamos a Include via CDN, elegimos la opción CSS only, copiamos y pegamos en la sección head de index.html.

# Angular 12 - Interacción de componentes. @Input() y @Output(). Tutorial en español #9JS

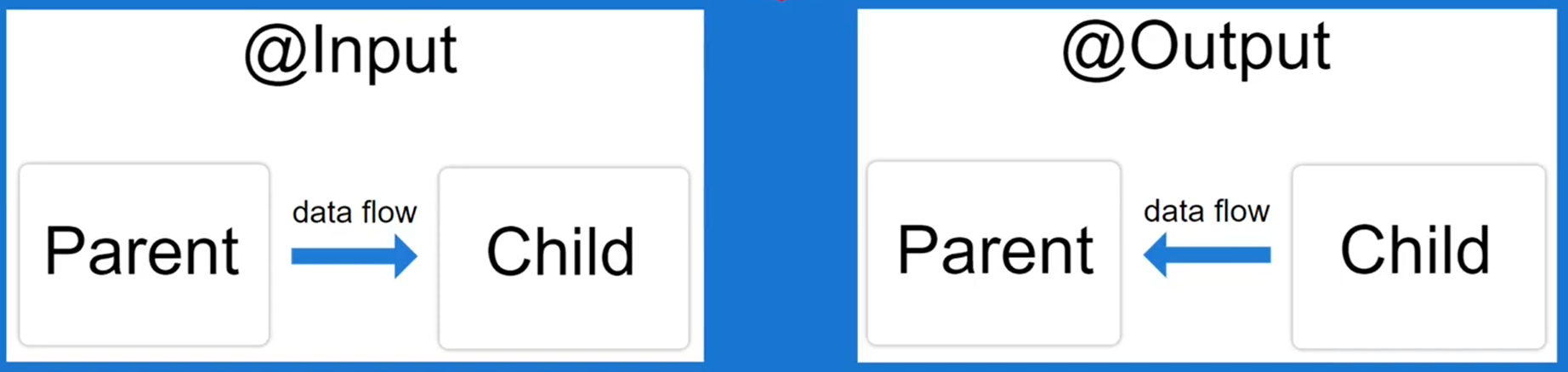
Excelente.

<https://www.youtube.com/watch?v=acxmiISz5Ag&ab_channel=JamesSchneider-Programaci%C3%B3nInform%C3%A1tica>

El proyecto se llama sharing-data-between-components.

El código del proyecto tiene muchos comentarios, que tratan de explicar cómo funciona cada uno de estos tres mecanismos.

Angular 12 - Interacción de componentes. @Input() y @Output(). Tutorial en español. Angular 12 - Sharing data between child and parent components.



Padre al hijo con @Input y property binding.

Hijo al padre con @Output y event binding.

00:00 - Intro y configuración del proyecto

04:38 - Enviar datos del padre al hijo

07:52 - Enviar datos del hijo al padre

15:08 - Borrar input

Se puede poner en index.html otros componentes además de app-root, pero hay que agregarlos a la sección bootstrap, como se explica en la siguiente respuesta de StackOverflow:

<https://stackoverflow.com/a/57916874/2740402>

# Comunicación entre componentes de Angular

<https://www.youtube.com/watch?v=df0eH9mM9nU&ab_channel=FernandoHerrera>

# Agregar routing a una SPA de Angular[[4]](#footnote-4)

Editar app.module.ts.

Agregar al principio:

import { [RouterModule](https://angular.io/api/router/RouterModule) } from '@angular/router';

Bajar hasta la sección @[NgModule](https://angular.io/api/core/NgModule)(). Agregar, dentro de la propiedad imports lo siguiente:

RouterModule.forRoot([

{path: 'crisis-list', component: CrisisListComponent},

{path: 'heroes-list', component: HeroesListComponent},

]),

de manera que quede por ejemplo así:

  imports: [

    BrowserModule,

    RouterModule.forRoot([

      { path: 'crisis-list', component: CrisisListComponent },

      { path: 'heroes-list', component: HeroesListComponent },

    ]),

  ],

Grabar.

Editar app.component.html. Agregar la directiva router-outlet:

<[router-outlet](https://angular.io/api/router/RouterOutlet)></[router-outlet](https://angular.io/api/router/RouterOutlet)>

Agregar los siguiente controles de navegación:

<nav>

<a class="button" [routerLink](https://angular.io/api/router/RouterLink)="/crisis-list">Crisis Center</a> |

<a class="button" [routerLink](https://angular.io/api/router/RouterLink)="/heroes-list">Heroes</a>

</nav>

1. <https://stackoverflow.com/a/44526528/2740402> [↑](#footnote-ref-1)
2. <https://dictionary.cambridge.org/dictionary/english/padding> [↑](#footnote-ref-2)
3. <https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_execution_policies?view=powershell-7.2#long-description> [↑](#footnote-ref-3)
4. <https://angular.io/guide/router-tutorial> [↑](#footnote-ref-4)