



TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE MILPA ALTA II

ASIGNATURA:

MEAN STACK FOR FRONT – END

TEMA:

Practica angular tryit

ELABORA:

POZOS RIVERA ALEJANDRO

NÚMERO DE CONTROL:

171190025

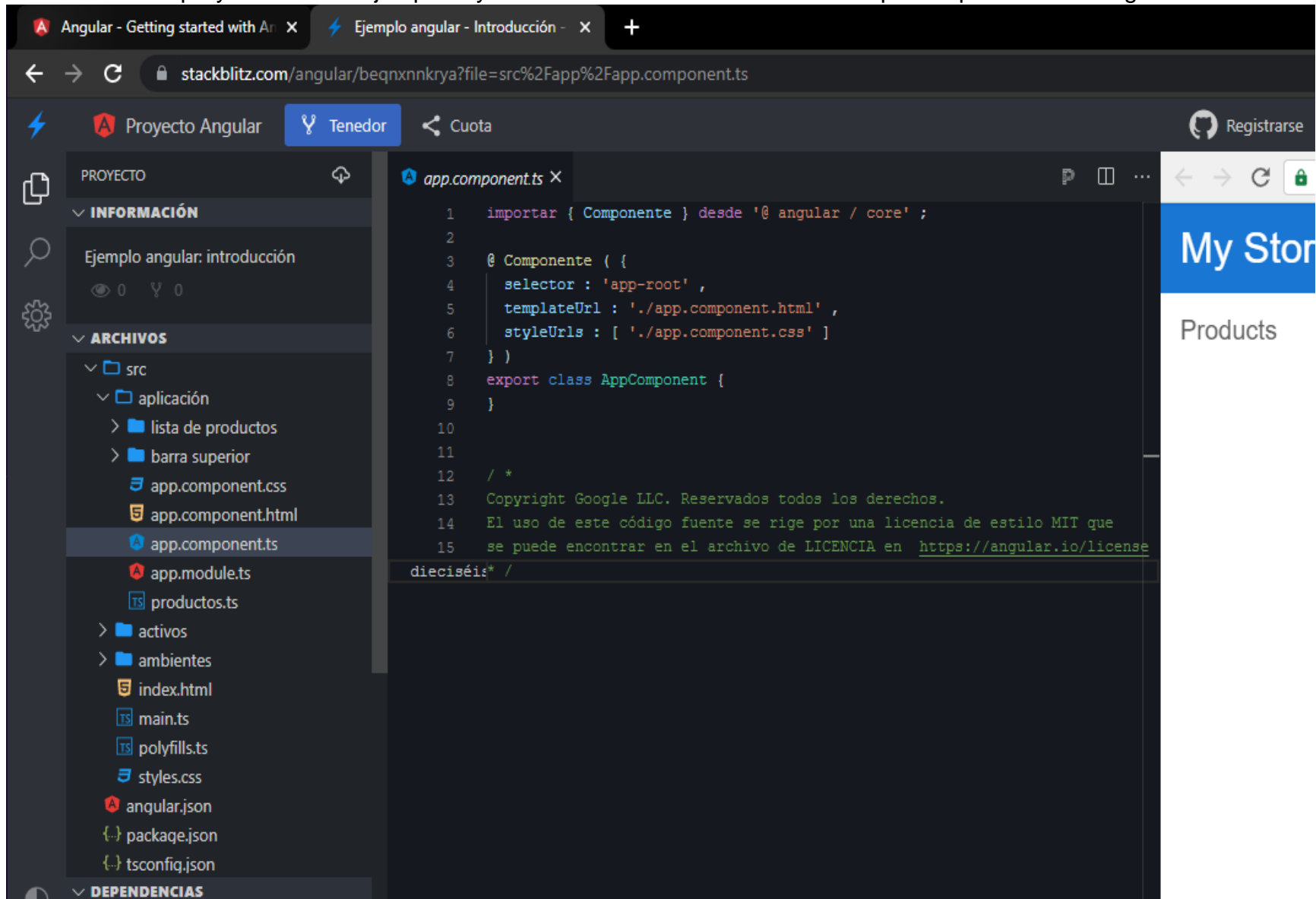
PROFESOR:

ROLDAN AQUINO SEGURA

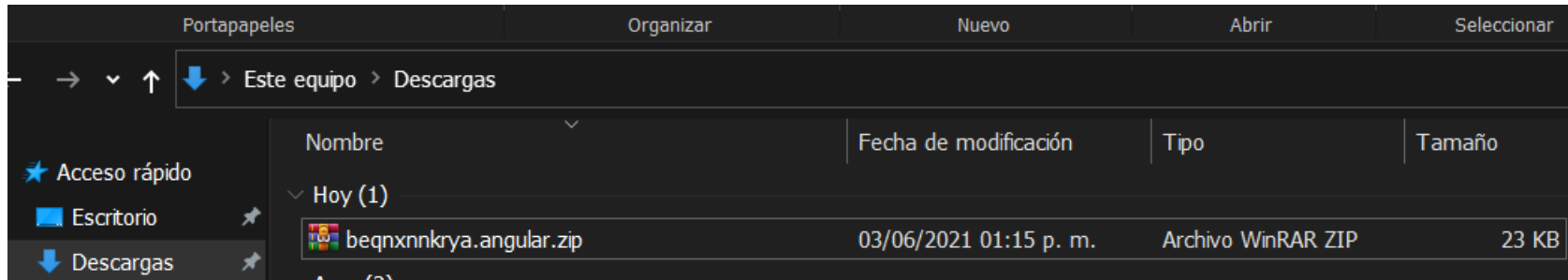
04 DE MAYO DE 2021

Levantar proyecto de angular

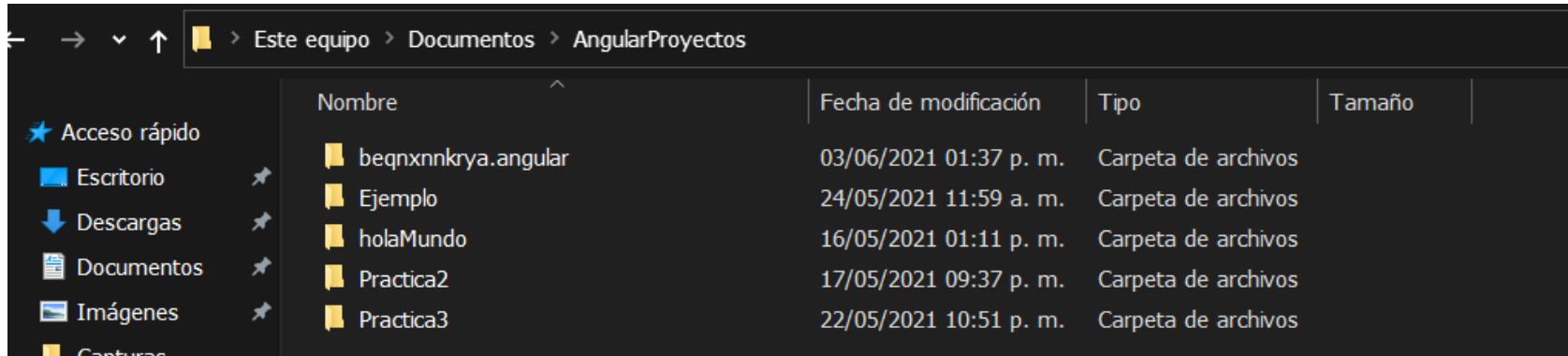
1.- primero vamos a dirigir a la página oficial de angular o al siguiente link <https://angular.io/start>, una vez estando ahí nos vamos a dirigir al apartado que dice ready-made sample project in StackBlitz. Y nos abrirá una nueva interfaz en donde nos muestra un proyecto de ejemplo y le daremos en la nube para poder descargar el archivo.



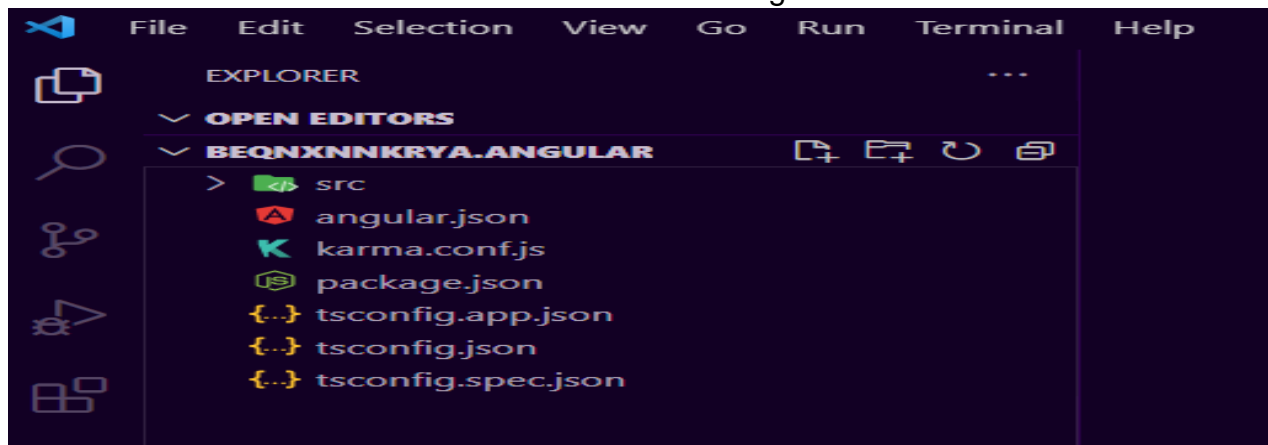
2.- Una vez que hayamos descargado el archivo nos vamos a dirigir a nuestro explorador de archivos → descargas → beqnxnnkrya.angular.rar y vamos a descomprimir el archivo.



3.- Ahora vamos a pasar la carpeta a nuestra carpeta de proyectos de angular



4.- Abriremos nuestro editor de código visual studio code y abriremos el archivo



5.- Abriremos la terminal y con el comando “ng serve” solo ejecutaremos la aplicación. Y como se puede ver a la hora de ejecutar la aplicación nos manda diferentes tipos de errores.

```
PS C:\Users\Admin\Documents\AngularProyectos\beqnxnnkrya.angular> ng serve
An unhandled exception occurred: Cannot find module '@angular-devkit/build-angular/package.json'
Require stack:
- C:\Users\Admin\AppData\Roaming\npm\node_modules\@angular\cli\node_modules\@angular-devkit\architect\node\node-modules-architect-host.js
- C:\Users\Admin\AppData\Roaming\npm\node_modules\@angular\cli\node_modules\@angular-devkit\architect\node\index.js
- C:\Users\Admin\AppData\Roaming\npm\node_modules\@angular\cli\models\architect-command.js
- C:\Users\Admin\AppData\Roaming\npm\node_modules\@angular\cli\commands\serve-impl.js
- C:\Users\Admin\AppData\Roaming\npm\node_modules\@angular\cli\node_modules\@angular-devkit\schematics\tools\export-ref.js
- C:\Users\Admin\AppData\Roaming\npm\node_modules\@angular\cli\node_modules\@angular-devkit\schematics\tools\index.js
- C:\Users\Admin\AppData\Roaming\npm\node_modules\@angular\cli\utilities\json-schema.js
- C:\Users\Admin\AppData\Roaming\npm\node_modules\@angular\cli\models\command-runner.js
- C:\Users\Admin\AppData\Roaming\npm\node_modules\@angular\cli\lib\cli\index.js
- C:\Users\Admin\AppData\Roaming\npm\node_modules\@angular\cli\lib\init.js
- C:\Users\Admin\AppData\Roaming\npm\node_modules\@angular\cli\bin\ng
See "C:\Users\Admin\AppData\Local\Temp\ng-bSQZ8l\angular-errors.log" for further details.
PS C:\Users\Admin\Documents\AngularProyectos\beqnxnnkrya.angular> █
```

6.- Para poder solucionar esto debemos desinstalar el CLI de angular con el siguiente comando “npm uninstall -g @angular/cli”, esto lo hacemos para que borre los datos en cache que se habían obtenido anteriormente.

```
PS C:\Users\Admin\Documents\AngularProyectos\beqnxnnkrya.angular> npm uninstall -g @angular/cli
removed 234 packages, and audited 1 package in 3s

found 0 vulnerabilities
PS C:\Users\Admin\Documents\AngularProyectos\beqnxnnkrya.angular> █
```

7.- Ahora lo volveremos a instalar con el siguiente comando “npm install -g @angular/cli”.

```
PS C:\Users\Admin\Documents\AngularProyectos\beqnxnnkrya.angular> npm install -g @angular/cli
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN deprecated uid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142

added 234 packages, and audited 235 packages in 20s

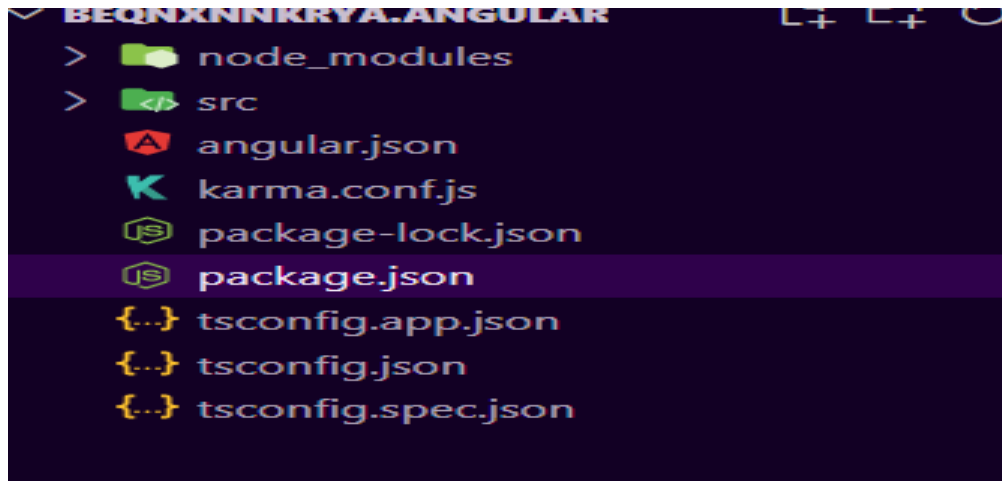
23 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\Admin\Documents\AngularProyectos\beqnxnnkrya.angular> █
```

8.- Instalaremos todos los node_modules con el comando “npm install --save --legacy-peer-deps” para que nos cargue completamente todo.

```
found 0 vulnerabilities
PS C:\Users\Admin\Documents\AngularProyectos\beqnxnnkrya.angular> npm install --save --legacy-peer-deps
[.....] / idealTree:worker-plugin: sill fetch manifest webpack-sources@1.3.0
```

Y como podemos ver ya nos crea la carpeta de node_modules.



7.- Ahora con el comando con el comando “ng serve” vamos a volver a ejecutar nuestra aplicación. Y primero nos manda una alerta que la versión de angular CLI que tenemos es la 12.0.3 y la versión de la aplicación es la 11.0.0, después nos muestra un link de una sugerencia para poder actualizar la aplicación a una versión más reciente.

```
PS C:\Users\Admin\Documents\AngularProyectos\beqnxnnkrya.angular> ng serve
Your global Angular CLI version (12.0.3) is greater than your local version (11.0.7). The local Angular CLI version is used.

To disable this warning use "ng config -g cli.warnings.versionMismatch false".
This version of CLI is only compatible with Angular versions 0.0.0 || ^11.0.0-beta || >=11.0.0 <12.0.0,
but Angular version 12.0.3 was found instead.

Please visit the link below to find instructions on how to update Angular.
https://update.angular.io/
```

9.- Le daremos clic en el link y nos redirigirá a una página donde podemos copiar el comando para poder actualizar la aplicación. El siguiente comando es “ng update @angular/core@12 @angular/cli@12”.

```
PS C:\Users\Admin\Documents\AngularProyectos\beqnxnnkrya.angular> ng update @angular/core@12 @angular/cli@12
Using package manager: 'npm'
Collecting installed dependencies...
Found 29 dependencies.
Fetching dependency metadata from registry...
Package '@angular/core' is already up to date.
  Updating package.json with dependency @angular-devkit/build-angular @ "12.0.3" (was "0.1100.7")...
  Updating package.json with dependency @angular/cli @ "12.0.3" (was "11.0.7")...
  Updating package.json with dependency @angular/compiler-cli @ "12.0.3" (was "11.0.9")...
  Updating package.json with dependency karma @ "6.3.3" (was "5.1.1")...
  Updating package.json with dependency typescript @ "4.2.4" (was "4.0.7")...
UPDATE package.json (1238 bytes)
```

10.- Ejecutaremos el mismo comando “ng serve” para poder ver si se actualizo o no la aplicación.

```
PS C:\Users\Admin\Documents\AngularProyectos\beqnxnnkrya.angular> ng serve
Generating browser application bundles (phase: setup)...Compiling @angular/core : es2015 as esm2015
Compiling @angular/common : es2015 as esm2015
Compiling @angular/platform-browser : es2015 as esm2015
Compiling @angular/router : es2015 as esm2015
Compiling @angular/platform-browser-dynamic : es2015 as esm2015
Compiling @angular/forms : es2015 as esm2015
✓Browser application bundle generation complete.

Initial Chunk Files | Names          | Size
vendor.js           | vendor         | 2.65 MB
polyfills.js        | polyfills      | 473.30 kB
styles.css, styles.js | styles        | 347.37 kB
main.js             | main           | 15.63 kB
runtime.js          | runtime        | 6.85 kB

| Initial Total | 3.48 MB

Build at: 2021-06-05T20:36:38.037Z - Hash: 628cea24f2f5fbadcd14 - Time: 37669ms

** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **

✓ Compiled successfully.
```

11.- Ya nos ejecuta la aplicación en el navegador.



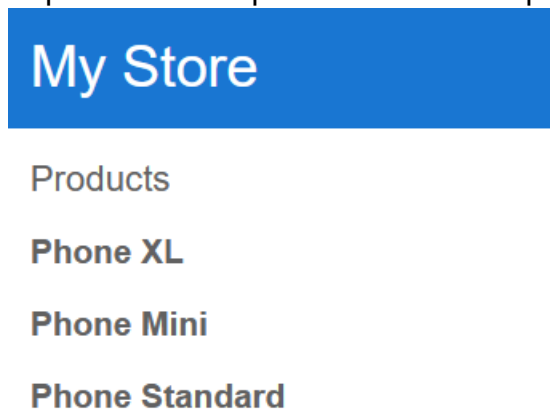
12.- Primera parte de la práctica

Agregue una directiva estructural en a, de la siguiente manera. `*ngFor<div>`

Dentro de `<div>`, agregue una `<h3>` y `{{ product.name }}`. La `{{ product.name }}` declaración es un ejemplo de la sintaxis de interpolación de Angular.

```
src > app > product-list > product-list.component.html > div > h3
1  <h2>Products</h2>
2
3  <div *ngFor="let product of products">
4    <h3>
5      {{product.name}}
6    </h3>
7  </div>
```

El panel de vista previa se actualiza para mostrar el nombre de cada producto en la lista.

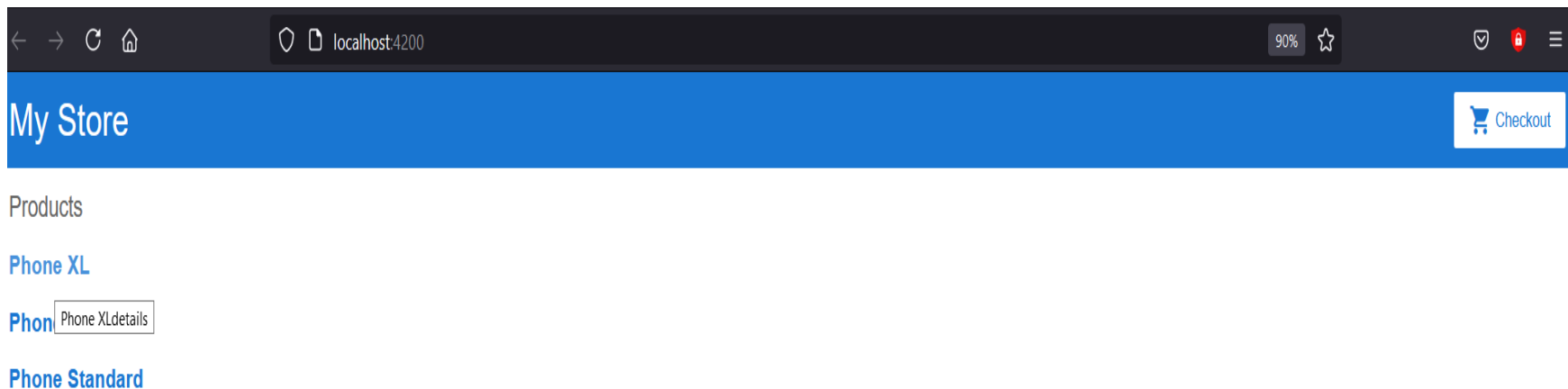


Para hacer que el nombre de cada producto sea un enlace a los detalles del producto, agregue el `<a>` elemento alrededor `{{ product.name }}`.

Establezca el título para que sea el nombre del producto utilizando la `[]` sintaxis de enlace de propiedad , de la siguiente manera:

```
src > app > product-list > product-list.component.html > div > h3
1  <h2>Products</h2>
2
3  <div *ngFor="let product of products">
4    <h3>
5      <a [title]= "product.name + ' details'"></a>
6      {{ product.name }}
7    </h3>
8  </div>
```

En el panel de vista previa, coloque el cursor sobre el nombre de un producto para ver el valor de la propiedad del nombre enlazado, que es el nombre del producto más la palabra "detalles". El enlace de propiedad `[]` permite usar el valor de la propiedad en una expresión de plantilla.



Agregue las descripciones de los productos. En un `<p>` elemento, use una directiva para que Angular solo cree el elemento si el producto actual tiene una descripción. `*ngIf<p>`

```
product-list.component.html X product-list.component.ts TS products.ts
src > app > product-list > product-list.component.html > div > p
1 <h2>Products</h2>
2
3 <div *ngFor="let product of products">
4 <h3>
5 <a [title]="product.name + 'details'">
6 {{ product.name}}
7 </a>
8 </h3>
9 <p *ngIf="product.description">
10 Description: {{ product.description}}
11 </p>
12 </div>
```

La aplicación ahora muestra el nombre y la descripción de cada producto en la lista.



Products

[Phone XL](#)

Description: A large phone with one of the best screens

[Phone Mini](#)

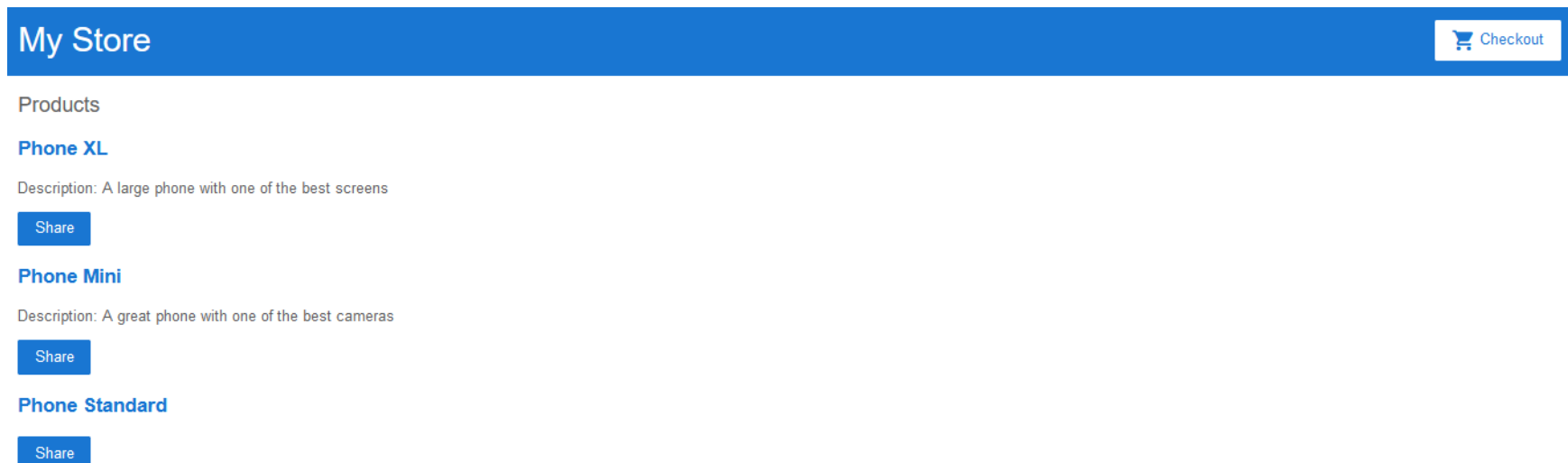
Description: A great phone with one of the best cameras

[Phone Standard](#)

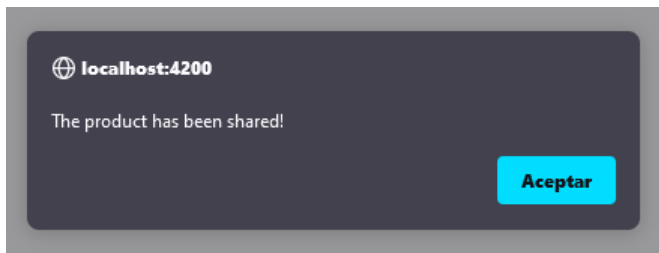
Agrega un botón para que los usuarios puedan compartir un producto. Vincula el clickevento del botón al share()método en product-list.component.ts. El enlace de eventos usa un conjunto de paréntesis, ()alrededor del evento, como en el (click)evento del <button>elemento.

```
product-list.component.html X product-list.component.ts TS products.ts
src > app > product-list > product-list.component.html > div > button
1 <h2>Products</h2>
2
3 <div *ngFor="let product of products">
4   <h3>
5     <a [title]="product.name + 'details'">
6       {{ product.name}}
7     </a>
8   </h3>
9   <p *ngIf="product.description">
10     Description: {{ product.description}}
11   </p>
12   <button (click)="share()">
13     Share
14   </button>
15 </div>
```

Cada producto ahora tiene un botón Compartir.



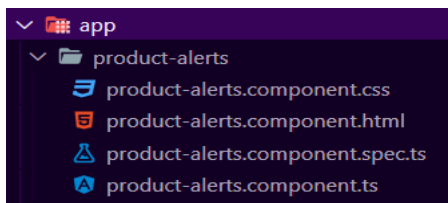
Al hacer clic en el botón Compartir, se activa una alerta que dice: "¡El producto ha sido compartido!".



13.- Segunda parte de la practica

Generar un nuevo componente llamado product-alerts. Para ello lo aremos con el comando "ng generate component product-alerts".

```
✓ Compiled successfully.  
PS C:\Users\Admin\Documents\AngularProyectos\beqnxnnkrya.angular> ng generate component product-alerts
```



Abrir product-alerts.component.ts.

```
product-alerts.component.ts X  
src > app > product-alerts > product-alerts.component.ts > ...  
1  import { Component, OnInit } from '@angular/core';  
2  
3  @Component({  
4    selector: 'app-product-alerts',  
5    templateUrl: './product-alerts.component.html',  
6    styleUrls: ['./product-alerts.component.css']  
7  })  
8  export class ProductAlertsComponent implements OnInit {  
9  
10   constructor() { }  
11  
12   ngOnInit(): void {  
13   }  
14  
15 }
```

Para configurar la ProductAlertsComponent recepción de datos de productos, primero importe Input de @angular/core.

```
product-alerts.component.ts X
src > app > product-alerts > product-alerts.component.ts > ProductAlertsComponent > product
1 import { Component, OnInit } from '@angular/core';
2 import { Input } from '@angular/core';
3 import { Product } from '../products';
```

En la ProductAlertsComponent definición de clase, defina una propiedad nombrada product con un decorador. El decorador indica que el valor de la propiedad pasa del padre del componente, [@Input\(\)@Input\(\)ProductListComponent](#)

```
export class ProductAlertsComponent implements OnInit {
  @Input() product!: Product;

  constructor() { }

  ngOnInit(): void {
  }
}
```

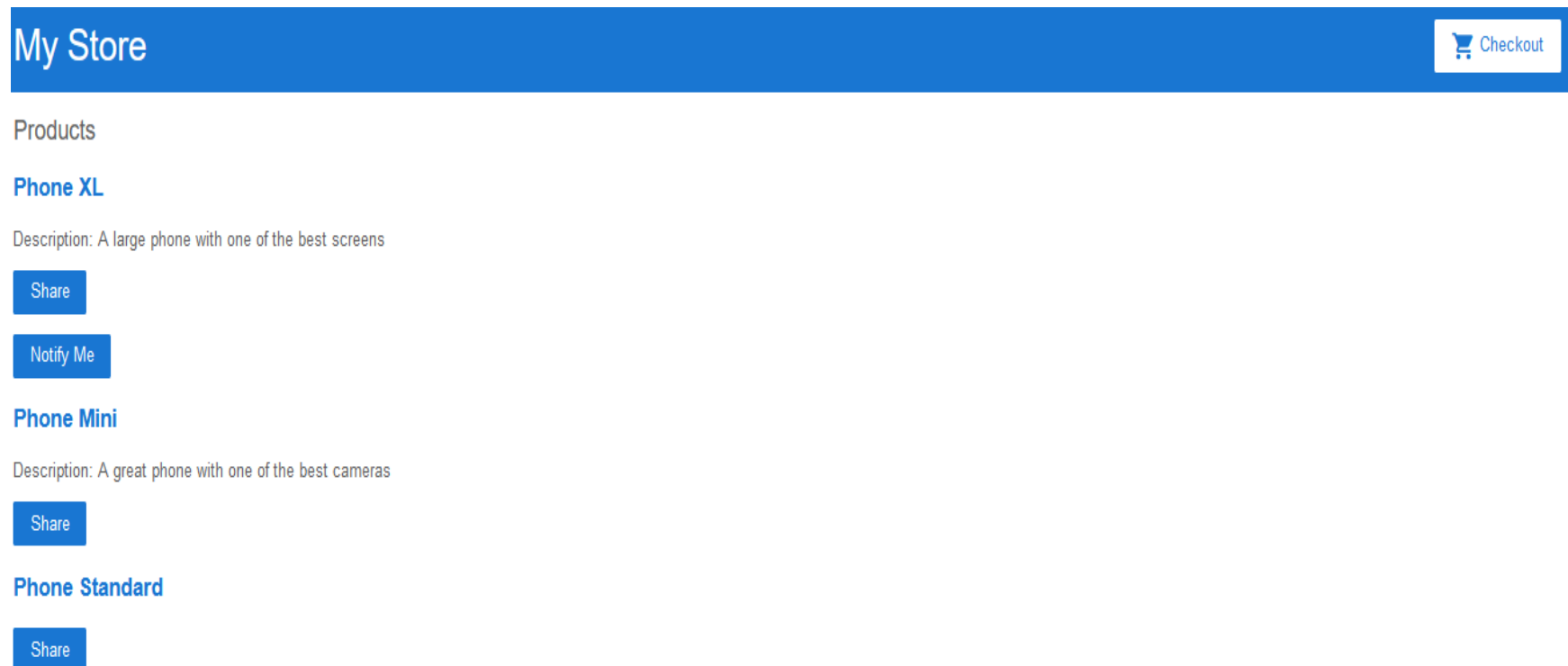
Abra product-alerts.component.html y reemplace el párrafo de marcador de posición con un botón Notificarme que aparece si el precio del producto es superior a \$ 700.

```
product-alerts.component.ts product-alerts.component.html X
src > app > product-alerts > product-alerts.component.html > p > button
1 <!-- <p>product-alerts works!</p> -->
2 <p *ngIf="product && product.price > 700">
3   <button>Notify Me</button>
4 </p>
```

Para mostrar ProductAlertsComponent como hijo de ProductListComponent, agregue el selector <app-product-alerts> a product-list.component.html. Pase el producto actual como entrada al componente mediante el enlace de propiedad.

```
<button (click)="share()">
  Share
</button>
<app-product-alerts
  [product]="product">
</app-product-alerts>
```

El componente de alerta de nuevo producto toma un producto como entrada de la lista de productos. Con esa entrada, muestra u oculta el botón Notificarme , según el precio del producto. El precio del Phone XL es de más de \$ 700, por lo que el botón Notificarme aparece en ese producto.



14.- Tercera parte de la practica

En product-alerts.component.ts, importar Outputy EventEmitterdesde @angular/core.

```
app > product-alerts > product-alerts.component.ts > ProductAlertsComponent > notify
import { Component } from '@angular/core';
import { Input } from '@angular/core';
import { Output, EventEmitter } from '@angular/core';
import { Product } from '../products';
```

En la clase de componente, defina una propiedad nombrada notifycon un decorador y una instancia de . La configuración con un le permite emitir un evento cuando cambia el valor de la propiedad.@Output()EventEmitter()ProductAlertsComponent@Output()ProductAlertsComponentnotify.

```
export class ProductAlertsComponent{
  @Input() product!: Product;
  @Output() notify = new EventEmitter();
}
```

En product-alerts.component.html, actualice el botón Notificarme con un enlace de evento para llamar al notify.emit()método.

```
c > app > product-alerts > product-alerts.component.html > p > button
1 <!-- <p>product-alerts works!</p> -->
2 <p *ngIf="product && product.price > 700">
3   <button (click)="notify.emit()">Notify Me</button>
4 </p>
```

En product-list.component.ts, defina un onNotify() método, similar al share() método.

```
✓ export class ProductListComponent {  
  products = products;  
  
  share() {  
    window.alert('The product has been shared!');  
  }  
  
  onNotify() {  
    window.alert('You will be notified when the product goes on sale');  
  }  
}
```

Actualice ProductListComponent para recibir datos de ProductAlertsComponent.

En product-list.component.html, enlace <app-product-alerts> al onNotify() método del componente de lista de productos. <app-product-alerts> es lo que muestra el botón Notificarme.

```
<app-product-alerts  
  [product]="product"  
  
  (notify)="onNotify()"  
>  
</app-product-alerts>
```

Haga clic en el botón Notificarme para activar una alerta que dice: "Se le notificará cuando el producto salga a la venta".

Cuadro de diálogo de confirmación de notificación de alerta de producto.

