[](https://angular.io/)

# Comenzando con Angular: su primer  aplicación

Angular.io

# Comenzando con Angular: su primer  aplicación

¡Bienvenido a Angular!

Este tutorial le presenta lo esencial de Angular al guiarlo a través de la construcción de un sitio de comercio electrónico simple con un catálogo, un carrito de compras y un formulario de pago. Utiliza el [entorno](https://stackblitz.com/) de desarrollo en línea [StackBlitz](https://stackblitz.com/) para que pueda comenzar de inmediato.

Esta guía utiliza el generador StackBlitz para mostrarle una aplicación simple y lista para usar que puede examinar y jugar de forma interactiva. En el desarrollo real, normalmente utilizará la [CLI angular](https://angular.io/guide/glossary#command-line-interface-cli) , una potente herramienta de línea de comandos que le permite generar y modificar aplicaciones. Para obtener más información, consulte la [Descripción general de](https://angular.io/cli) la [CLI](https://angular.io/cli) .

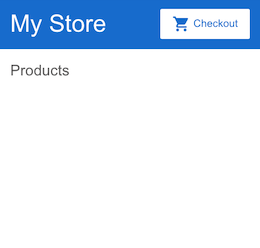
¿NUEVO EN EL DESARROLLO WEB?

Hay muchos recursos para complementar los documentos angulares. Los documentos MDN de Mozilla incluyen introducciones tanto [HTML](https://developer.mozilla.org/en-US/docs/Learn/HTML) como [JavaScript](https://developer.mozilla.org/en-US/docs/Web/JavaScript) . [Los documentos de TypeScript](https://www.typescriptlang.org/docs/home.html) incluyen un tutorial de 5 minutos. Varias plataformas de cursos en línea, como [Udemy](http://www.udemy.com/) y [Codecademy](https://www.codecademy.com/) , también cubren conceptos básicos de desarrollo web.

## Crear un nuevo  proyecto

#### [Haga clic aquí para crear un nuevo proyecto en StackBlitz.](https://angular.io/generated/live-examples/getting-started-v0/stackblitz.html)

StackBlitz crea una aplicación Angular de inicio con una barra superior (que contiene el nombre de la tienda y el ícono de pago) y el título de una lista de productos.



CONSEJOS STACKBLITZ

* Inicie sesión en StackBlitz para que pueda guardar y reanudar su trabajo. Si tiene una cuenta de GitHub, puede iniciar sesión en StackBlitz con esa cuenta. Para guardar su progreso, primero bifurque el proyecto con el botón Bifurcación en la esquina superior izquierda, luego podrá guardar su trabajo en su propia cuenta StackBlitz haciendo clic en el botón Guardar.
* Para copiar un ejemplo de código de este tutorial, haga clic en el icono en la parte superior derecha del cuadro de ejemplo de código y luego pegue el fragmento de código del portapapeles en StackBlitz.
* Si el panel de vista previa de StackBlitz no muestra lo que espera, guarde y luego haga clic en el botón Actualizar.
* StackBlitz mejora continuamente, por lo que puede haber ligeras diferencias en el código generado, pero el comportamiento de la aplicación será el mismo.

## sintaxis de plantilla

La sintaxis de plantilla de Angular extiende HTML y JavaScript. Esta sección presenta la sintaxis de la plantilla al mejorar el área "Productos".

Para ayudarlo a comenzar, los siguientes pasos utilizan datos y métodos de productos predefinidos del archivo:

product-list.component.ts

1. En la product-list carpeta, abra el archivo de plantilla product-list.component.html.
2. Modifique la plantilla de la lista de productos para mostrar una lista de nombres de productos.
   1. Cada producto en la lista se muestra de la misma manera, uno tras otro en la página. Para iterar sobre la lista predefinida de productos, coloque la directiva en un <div> como sigue:\*[ngFor](https://angular.io/api/common/NgForOf)

src / app / product-list / product-list.component.html

<h2>Products</h2>

<div \*[ngFor](https://angular.io/api/common/NgForOf)="let product of products">

</div>

Con \*[ngFor](https://angular.io/api/common/NgForOf), <div>las repeticiones para cada producto en la lista

\*[ngFor](https://angular.io/api/common/NgForOf)es una "directiva estructural". Las directivas estructurales dan forma o remodelan la estructura del DOM, típicamente agregando, eliminando y manipulando los elementos a los que están unidos. Cualquier directiva con un asterisco \*, es una directiva estructural.

b.Para mostrar los nombres de los productos, use la sintaxis de interpolación {{ }}. La interpolación representa el valor de una propiedad como texto. Dentro de <div>, agregue un <h3>para mostrar la interpolación de la propiedad del nombre del producto:

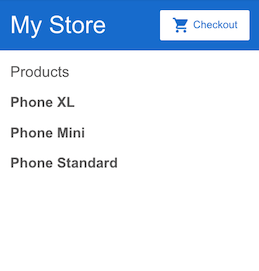
src / app / product-list / product-list.component.html

content\_copy<h2>Products</h2>

<div \*[ngFor](https://angular.io/api/common/NgForOf)="let product of products"><h3>{{ product.name }}

</h3></div>

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



1. Para hacer que cada nombre de producto sea un enlace a los detalles del producto, agregue el elemento y establezca su título para que sea el nombre del producto utilizando la sintaxis de enlace de propiedad , de la siguiente manera:<[a](https://angular.io/api/router/RouterLinkWithHref)>[ ]

src / app / product-list / product-list.component.html

content\_copy<h2>Products</h2>

<div \*[ngFor](https://angular.io/api/common/NgForOf)="let product of products">

<h3>

<[a](https://angular.io/api/router/RouterLinkWithHref) [title]="product.name + ' details'">

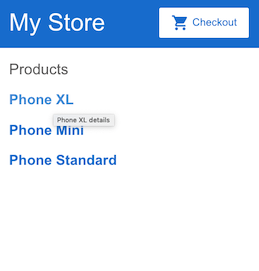
{{ product.name }}

</[a](https://angular.io/api/router/RouterLinkWithHref)>

</h3>

</div>

En el panel de vista previa, mantenga el puntero sobre el nombre del producto para ver el valor de la propiedad del nombre enlazado, que es el nombre del producto más la palabra "detalles". La interpolación le {{ }}permite representar el valor de la propiedad como texto; El enlace de propiedad le [ ]permite utilizar el valor de la propiedad en una expresión de plantilla.



1. Agregue las descripciones del producto. En el <p>elemento, use una directiva para que Angular solo cree el elemento si el producto actual tiene una descripción.\*[ngIf](https://angular.io/api/common/NgIf)<p>

src / app / product-list / product-list.component.html

content\_copy<h2>Products</h2>

<div \*[ngFor](https://angular.io/api/common/NgForOf)="let product of products">

<h3>

<[a](https://angular.io/api/router/RouterLinkWithHref) [title]="product.name + ' details'">

{{ product.name }}

</[a](https://angular.io/api/router/RouterLinkWithHref)>

</h3>

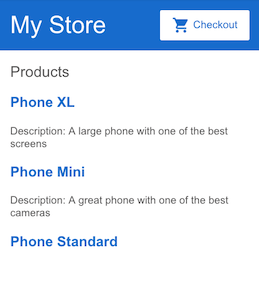
<p \*[ngIf](https://angular.io/api/common/NgIf)="product.description">

Description: {{ product.description }}

</p>

</div>

La aplicación ahora muestra el nombre y la descripción de cada producto en la lista. Tenga en cuenta que el producto final no tiene un párrafo de descripción. Debido a que la propiedad de descripción del producto está vacía, Angular no crea el <p>elemento, incluida la palabra "Descripción".



1. Agregue un botón para que los usuarios puedan compartir un producto con amigos. Vincula el clickevento del botón al share()método (in product-list.component.ts). El enlace de eventos utiliza un conjunto de paréntesis, ( )alrededor del evento, como en el siguiente <button>elemento:

src / app / product-list / product-list.component.html

content\_copy<h2>Products</h2>

<div \*[ngFor](https://angular.io/api/common/NgForOf)="let product of products">

<h3>

<[a](https://angular.io/api/router/RouterLinkWithHref) [title]="product.name + ' details'">

{{ product.name }}

</[a](https://angular.io/api/router/RouterLinkWithHref)>

</h3>

<p \*[ngIf](https://angular.io/api/common/NgIf)="product.description">

Description: {{ product.description }}

</p>

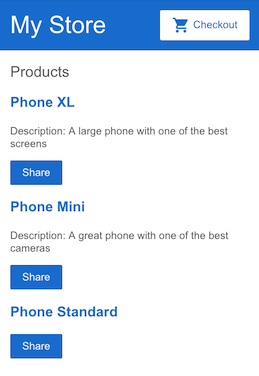
<button (click)="share()">

Share

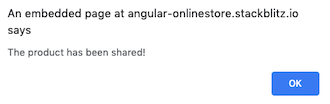
</button>

</div>

Cada producto ahora tiene un botón "Compartir":



Prueba el botón "Compartir":



La aplicación ahora tiene una lista de productos y una función para compartir. En el proceso, aprendió a usar cinco características comunes de la sintaxis de plantilla de Angular:

* \*[ngFor](https://angular.io/api/common/NgForOf)
* \*[ngIf](https://angular.io/api/common/NgIf)
* Interpolación {{ }}
* Enlace de propiedad [ ]
* Enlace de eventos ( )

Para obtener más información sobre las capacidades completas de la sintaxis de plantilla de Angular, consulte [Sintaxis de plantilla](https://angular.io/guide/template-syntax) .

## componentes

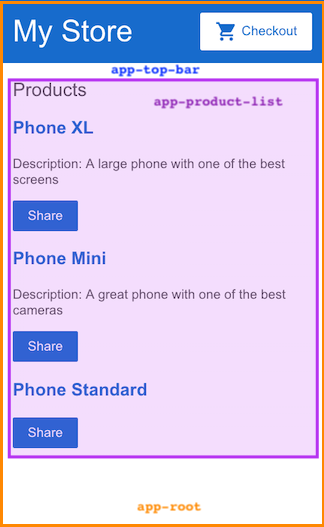
Los componentes definen áreas de responsabilidad en la interfaz de usuario, o UI, que le permiten reutilizar conjuntos de funcionalidades de UI. Ya ha creado uno con el componente de lista de productos.

Un componente consta de tres cosas:

* ****Una clase de componente**** que maneja datos y funcionalidad. En la sección anterior, los datos del producto y el share()método en la clase de componente manejan los datos y la funcionalidad, respectivamente.
* ****Una plantilla HTML**** que determina la interfaz de usuario. En la sección anterior, la plantilla HTML de la lista de productos muestra el nombre, la descripción y un botón "Compartir" para cada producto.
* ****Estilos específicos de componentes**** que definen la apariencia. Aunque la lista de productos no define ningún estilo, aquí es donde reside el componente CSS.

Una aplicación angular comprende un árbol de componentes, en el que cada componente angular tiene un propósito y una responsabilidad específicos.

Actualmente, la aplicación de ejemplo tiene tres componentes:



* app-root(cuadro naranja) es el shell de la aplicación. Este es el primer componente para cargar y el padre de todos los demás componentes. Puedes pensarlo como la página base.
* app-top-bar (fondo azul) es el nombre de la tienda y el botón de pago.
* app-product-list (cuadro morado) es la lista de productos que modificó en la sección anterior.

La siguiente sección amplía las capacidades de la aplicación al agregar un nuevo componente, una alerta de producto, como elemento secundario del componente de la lista de productos.

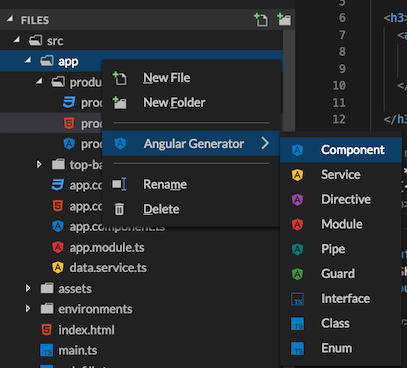
Para obtener más información sobre los componentes y cómo interactúan con las plantillas, consulte [Introducción a los componentes](https://angular.io/guide/architecture-components) .

## entrada

Actualmente, la lista de productos muestra el nombre y la descripción de cada producto. El componente de la lista de productos también define una productspropiedad que contiene datos importados para cada producto de la productsmatriz en products.ts.

El siguiente paso es crear una nueva función de alerta que tome un producto como entrada. La alerta verifica el precio del producto y, si el precio es superior a $ 700, muestra un botón "Notificarme" que permite a los usuarios registrarse para recibir notificaciones cuando el producto sale a la venta.

1. Cree un nuevo componente de alertas de productos.
   1. Haga clic derecho en la appcarpeta y use Angular Generatorpara generar un nuevo componente llamado product-alerts.



El generador crea archivos de inicio para las tres partes del componente:

* + - product-alerts.component.ts
    - product-alerts.component.html
    - product-alerts.component.css

1. Abierto product-alerts.component.ts.

src / app / product-alertas / producto-alertas.component.ts

content\_copyimport { [Component](https://angular.io/api/core/Component), [OnInit](https://angular.io/api/core/OnInit) } from '@angular/core';

@[Component](https://angular.io/api/core/Component)({

selector: 'app-product-alerts',

templateUrl: './product-alerts.component.html',

[styleUrls](https://angular.io/api/core/Component#styleUrls): ['./product-alerts.component.css']

})

export class ProductAlertsComponent implements [OnInit](https://angular.io/api/core/OnInit) {

constructor() { }

ngOnInit() {

}

}

* 1. Fíjate en el decorador. Esto indica que la siguiente clase es un componente. Proporciona metadatos sobre el componente, incluidos su selector, plantillas y estilos.@[Component](https://angular.io/api/core/Component)()
     + El selectoridentifica el componente. El selector es el nombre que le da al componente angular cuando se representa como un elemento HTML en la página. Por convención, los selectores de componentes angulares comienzan con el prefijo app-, seguido del nombre del componente.
     + Los nombres de plantilla y estilo hacen referencia a los archivos HTML y CSS que genera StackBlitz.
  2. La definición del componente también exporta la clase ProductAlertsComponent, que maneja la funcionalidad del componente.

1. Configure el nuevo componente de alertas de productos para recibir un producto como entrada:
   1. Importar [Input](https://angular.io/api/core/Input)desde @angular/core.

src / app / product-list / product-alertas.component.ts

content\_copyimport { [Component](https://angular.io/api/core/Component), [OnInit](https://angular.io/api/core/OnInit) } from '@angular/core';

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

* 1. En la ProductAlertsComponentdefinición de clase, defina una propiedad nombrada productcon un decorador. El decorador indica que el valor de la propiedad pasa del componente principal del componente, el componente de la lista de productos.@[Input](https://angular.io/api/core/Input)()@[Input](https://angular.io/api/core/Input)()

src / app / product-list / product-alertas.component.ts

content\_copyexport class ProductAlertsComponent implements [OnInit](https://angular.io/api/core/OnInit) {

@[Input](https://angular.io/api/core/Input)() product;

constructor() { }

ngOnInit() {

}

}

1. Defina la vista para el nuevo componente de alerta de producto.
   1. Abra la product-alerts.component.htmlplantilla y reemplace el párrafo del marcador de posición con un botón "Notificarme" que aparece si el precio del producto es superior a $ 700.

src / app / product-alertas / producto-alertas.component.html

content\_copy<p \*[ngIf](https://angular.io/api/common/NgIf)="product.price > 700">

<button>Notify Me</button>

</p>

1. Display the new product alert component as a child of the product list.
   1. Open product-list.component.html.
   2. To include the new component, use its selector, app-product-alert, as you would an HTML element.
   3. Pass the current product as input to the component using property binding.

src/app/product-list/product-list.component.html

content\_copy<button (click)="share()">

Share

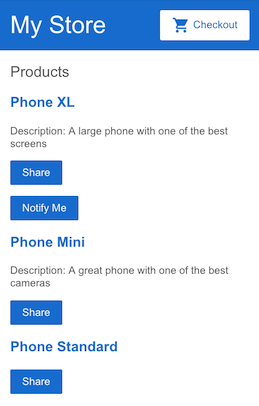
</button>

<app-product-alerts

[product]="product">

</app-product-alerts>

The new product alert component takes a product as input from the product list. With that input, it shows or hides the "Notify Me" button, based on the price of the product. The Phone XL price is over $700, so the "Notify Me" button appears on that product.



See [Component Interaction](https://angular.io/guide/component-interaction) for more information about passing data from a parent to child component, intercepting and acting upon a value from the parent, and detecting and acting on changes to input property values.

## Output

To make the "Notify Me" button work, you need to configure two things:

* the product alert component to emit an event when the user clicks "Notify Me"
* the product list component to act on that event

1. Open product-alerts.component.ts.
2. Import [Output](https://angular.io/api/core/Output) and [EventEmitter](https://angular.io/api/core/EventEmitter) from @angular/core:

src/app/product-alerts/product-alerts.component.ts

content\_copyimport { [Component](https://angular.io/api/core/Component) } from '@angular/core';

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

import { [Output](https://angular.io/api/core/Output), [EventEmitter](https://angular.io/api/core/EventEmitter) } from '@angular/core';

1. In the component class, define a property named notify with an @[Output](https://angular.io/api/core/Output)() decorator and an instance of [EventEmitter](https://angular.io/api/core/EventEmitter)(). This allows the product alert component to emit an event when the value of the notify property changes.

src/app/product-alerts/product-alerts.component.ts

content\_copyexport class ProductAlertsComponent {

@[Input](https://angular.io/api/core/Input)() product;

@[Output](https://angular.io/api/core/Output)() notify = new [EventEmitter](https://angular.io/api/core/EventEmitter)();

}

1. In the product alert template, product-alerts.component.html, update the "Notify Me" button with an event binding to call the notify.emit() method.

src/app/product-alerts/product-alerts.component.html

content\_copy<p \*[ngIf](https://angular.io/api/common/NgIf)="product.price > 700">

<button (click)="notify.emit()">Notify Me</button>

</p>

1. Next, define the behavior that should happen when the user clicks the button. Recall that it's the parent, product list component—not the product alerts component—that's acts when the child raises the event. In product-list.component.ts, define an onNotify() method, similar to the share() method:

src/app/product-list/product-list.component.ts

content\_copyexport 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');

}

}

1. Finally, update the product list component to receive output from the product alerts component.

In product-list.component.html, bind the app-product-alerts component (which is what displays the "Notify Me" button) to the onNotify() method of the product list component.

src/app/product-list/product-list.component.html

content\_copy<button (click)="share()">

Share

</button>

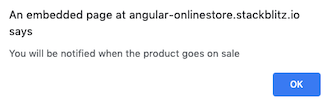
<app-product-alerts

[product]="product"

(notify)="onNotify()">

</app-product-alerts>

1. Try the "Notify Me" button:



See [Component Interaction](https://angular.io/guide/component-interaction) for more information about listening for events from child components, reading child properties or invoking child methods, and using a service for bi-directional communication between components.

## Next steps

Congratulations! You've completed your first Angular app!

You have a basic online store catalog with a product list, "Share" button, and "Notify Me" button. You've learned about the foundation of Angular: components and template syntax. You've also learned how the component class and template interact, and how components communicate with each other.

To continue exploring Angular, choose either of the following options:

* [Continúe con la sección "Enrutamiento"](https://angular.io/start/routing) para crear una página de detalles del producto a la que se pueda acceder haciendo clic en el nombre de un producto y que tenga su propio patrón de URL.
* [Vaya a la sección "Implementación"](https://angular.io/start/deployment) para pasar al desarrollo local o implemente su aplicación en Firebase o en su propio servidor.
* [Comenzando con Angular: su primera aplicación](https://angular.io/start#getting-started-with-angular-your-first-app)
* [Crea un nuevo proyecto](https://angular.io/start#create-a-new-project)
* [Sintaxis de plantilla](https://angular.io/start#template-syntax)
* [Componentes](https://angular.io/start#components)
* [Entrada](https://angular.io/start#input)
* [Salida](https://angular.io/start#output)
* [Próximos pasos](https://angular.io/start#next-steps)

### RECURSOS

* [Acerca de](https://angular.io/about)
* [Listado de recursos](https://angular.io/resources)
* [Carpeta de prensa](https://angular.io/presskit)
* [Blog](https://blog.angular.io/)
* [Analítica de uso](https://angular.io/analytics)

### AYUDA

* [Desbordamiento de pila](https://stackoverflow.com/questions/tagged/angular)
* [Gitter](https://gitter.im/angular/angular)
* [Informar problemas](https://github.com/angular/angular/issues)
* [Código de Conducta](https://github.com/angular/code-of-conduct/blob/master/CODE_OF_CONDUCT.md)

### COMUNIDAD

* [Eventos](https://angular.io/events)
* [Meetups](http://www.meetup.com/topics/angularjs/)
* [Gorjeo](https://twitter.com/angular)
* [GitHub](https://github.com/angular/angular)
* [Contribuir](https://angular.io/contribute)

### IDIOMAS

* [简体 中文 版](https://angular.cn/)
* [正 體 中文 版](https://angular.tw/)
* [日本語 版](https://angular.jp/)
* [한국어](https://angular.kr/)

Super-alimentado por Google © 2010-2019. Código con licencia bajo una [licencia de estilo MIT](https://angular.io/license) . Documentación bajo licencia [CC BY 4.0](http://creativecommons.org/licenses/by/4.0/) .

Versión 8.2.10-local + sha.2952ea57a.

 📁 📂 📄 @ 

@