

Coches segunda mano

Angular es un popular marco front-end que permite a los desarrolladores crear aplicaciones web dinámicas utilizando una arquitectura basada en componentes. Con sus potentes características y herramientas, Angular es ampliamente utilizado para crear aplicaciones responsivas y fáciles de usar. En este ejercicio, aprenderás a utilizar Angular para crear una aplicación web desde cero. Explorarás las diferentes características de Angular, incluyendo componentes, plantillas, interpolación de texto, bindings, routing y formularios. Esta página se realizará por equipos.

Cree una nueva aplicación Angular con la CLI.

- Abre un símbolo del sistema o terminal y ejecuta el comando: `ng new car-app`
- Elige las opciones por defecto para la configuración del proyecto, incluyendo el formato de la hoja de estilos (CSS)
- Navegue hasta el directorio del proyecto: `cd car-app`
- Inicie la aplicación con el comando: `ng serve`

Crea un componente navbar

- Implementa un componente navbar con enlaces a varias páginas, como "Inicio", "Coches en venta", "Vende tu coche", "Sobre nosotros" y "Contacto".
- Cree un nuevo componente llamado navbar utilizando el comando: `ng generate component navbar`
- En el archivo `navbar.component.html`, añada el HTML de la barra de navegación, incluidos los enlaces a las distintas páginas.
- Utiliza la directiva `routerLink` para enlazar a las diferentes páginas de la aplicación.
- Importa el `RouterModule` y `Routes` de `@angular/router` en el archivo `app.module.ts` y define las rutas para las diferentes páginas.

Página inicio

- Cree un componente de página de inicio que muestre un mensaje de bienvenida y alguna información sobre el sitio web.

- Crea un nuevo componente llamado home usando el comando: `ng generate component home`
- En el archivo `home.component.html`, añade el HTML para la página de inicio, incluyendo un mensaje de bienvenida y algo de información sobre el sitio web.

Coches en venta

- Cree un componente de coches que muestre una lista de coches en venta, incluyendo su marca, modelo, año, precio y una imagen. Los datos deben obtenerse de un servicio.
- Cree un nuevo componente llamado cars utilizando el comando: `ng generate component cars`
- En el archivo `cars.component.html`, añade el HTML para mostrar la lista de coches.
- Crea un nuevo servicio llamado `car.service` utilizando el comando: `ng generate service car`
- En el archivo `car.service.ts`, define una función que devuelva una matriz de coches con propiedades como marca, modelo, año, precio e imagen.
- En el archivo `cars.component.ts`, importa el servicio `car.service` y utilízalo para obtener la lista de coches a mostrar en el HTML.
- Implementa una funcionalidad de búsqueda que permita a los usuarios buscar coches basándose en su marca, modelo, año o precio.
- En el archivo `cars.component.html`, añade un campo de búsqueda y un botón de búsqueda.
- En el archivo `cars.component.ts`, cree una función que filtre la lista de coches en función de los criterios de búsqueda introducidos por el usuario.

- Vincula el campo de entrada de búsqueda a una propiedad utilizando la directiva `ngModel`, y utiliza el evento (`click`) para llamar a la función de búsqueda.

Detalle coche

- Cree un componente de detalles del coche que muestre información detallada sobre un coche específico, incluyendo su marca, modelo, año, precio, descripción e imagen.
- Cree un nuevo componente llamado `car-details` utilizando el comando: `ng generate component car-details`
- En el archivo `car-details.component.html`, añada el HTML para mostrar los detalles de un coche específico.
- Define una ruta en el archivo `app.module.ts` que incluya un parámetro para el ID del coche, y utilízalo para navegar al componente `car-details`.
- En el archivo `car-details.component.ts`, importa `ActivatedRoute` y `car.service` para obtener los detalles del coche a mostrar.

Vende tu coche

- Implementar un formulario para que los usuarios vendan su coche, incluyendo campos para la marca, modelo, año, precio, descripción e imagen. El formulario debe incluir validación usando las características incorporadas de Angular.
- Crea un nuevo componente llamado `"SellCarComponent"` usando el CLI de Angular.

- Dentro del SellCarComponent, crea una plantilla con elementos de formulario para cada campo: marca, modelo, año, precio, descripción e imagen.
- Añade validación a cada campo utilizando las directivas de validación de formularios integradas en Angular (por ejemplo, requerido, longitud mínima/máxima, patrón).
- Implementa una función de envío en el SellCarComponent que enviará los datos del formulario a un servidor o los guardará en una base de datos.
- Si quieres permitir al usuario subir una imagen, necesitarás usar una librería como ng2-file-upload.

Sobre Nosotros

- Cree un componente about que muestre información sobre el sitio web y sus creadores.
- Crea un nuevo componente llamado "AboutComponent" usando el CLI de Angular.
- Dentro del AboutComponent, crea una plantilla que muestre información sobre el sitio web y sus creadores. Esto podría incluir texto, imágenes y enlaces a perfiles de redes sociales.

Componente Contacto

- Crea un componente de contacto que contenga un formulario para que los usuarios envíen solicitudes de contacto, incluyendo campos para su nombre, correo electrónico, asunto y mensaje. El formulario debe incluir validación utilizando las funciones integradas de Angular.

- Cree un nuevo componente llamado "ContactComponent" utilizando el CLI de Angular.
- Dentro del ContactComponent, crea una plantilla con elementos de formulario para cada campo: nombre, email, asunto y mensaje.
- Añade validación a cada campo utilizando las directivas de validación de formularios integradas en Angular (por ejemplo, requerido, email).
- Implementa una función de envío en el ContactComponent que enviará los datos del formulario a un servidor o los guardará en una base de datos.

Routing

- Utilizar rutas para navegar entre las diferentes páginas.
- Define las rutas para cada página en el archivo app-routing.module.ts.
- Importa los componentes para cada página y añádelos a la matriz de rutas.
- En el archivo app.component.html, añade una barra de navegación con enlaces a cada página. Utiliza la directiva routerLink de Angular para navegar entre las páginas.

Estilos

- Añade estilos a la aplicación usando CSS o un framework CSS.
- Puedes añadir estilos CSS a componentes individuales añadiendo una propiedad styleUrls a los metadatos del componente y creando un archivo CSS.

- Alternativamente, puedes añadir estilos globales a toda la aplicación creando un archivo `styles.css` y añadiéndolo al archivo `angular.json` bajo el array "styles".
- Si quieres usar un framework CSS como Bootstrap o Material Design, puedes instalarlo usando npm e incluir los archivos CSS en tu aplicación. Puede que también necesites importar archivos JavaScript para que ciertos componentes funcionen correctamente.

Rúbrica

Si el programa da un error de ejecución o no es sintácticamente correcto entonces se le pondrá un 0 a la práctica. Se valorará el uso de código que siga SOLID/KISS/DRY y las buenas prácticas comentadas en clase.

Navbar	1
Inicio	0.5
Coches en venta	3
Vende tu coche	2
Sobre nosotros	1
Contacto	1
GitHub	1
SOLID/KISS/DRY	0.5
TOTAL	10

Entregable

El entregable será la URL del GitHub donde tendréis el proyecto incluyendo HTML, CSS y Javascript.