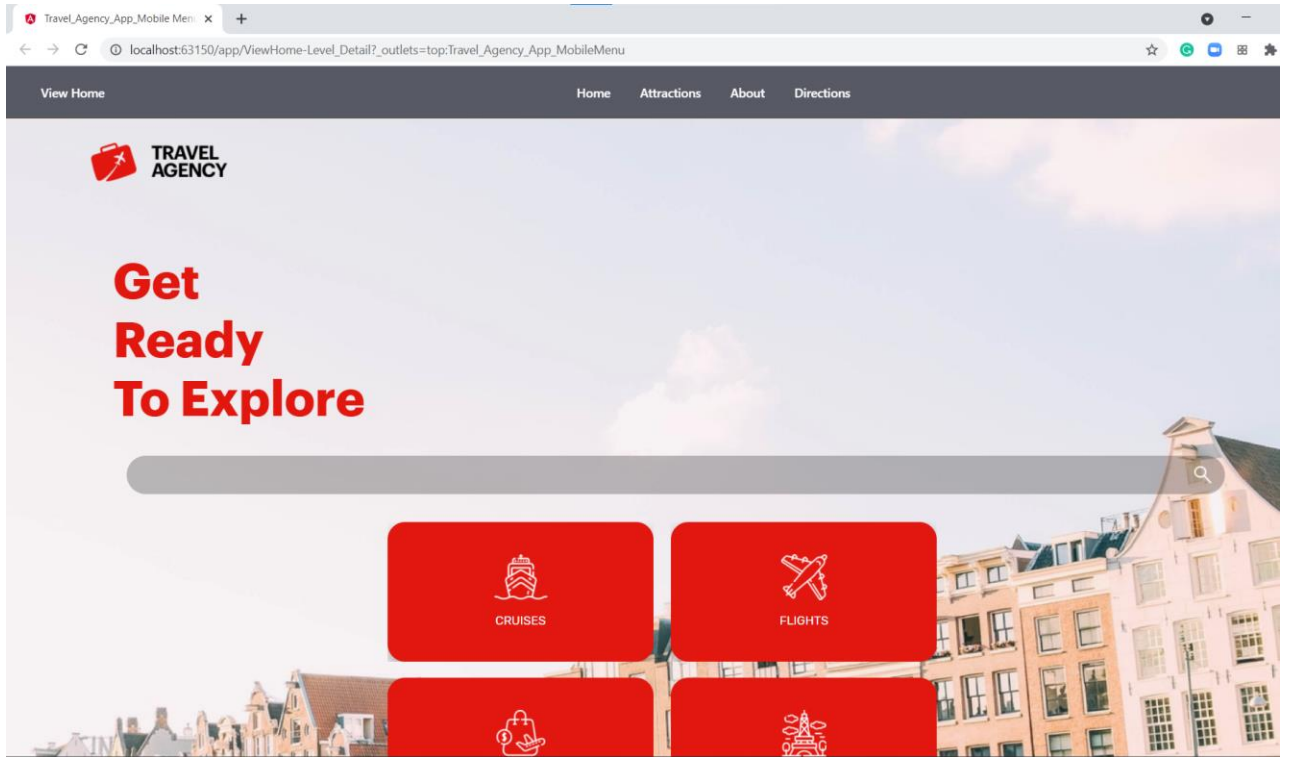


Primeros pasos con Angular

GeneXus™

En el desarrollo del curso, vamos a mostrar cómo implementamos una aplicación para los clientes de una Agencia de Viajes y todos nuestros ejemplos los haremos completando su desarrollo. Pero primero echemos un vistazo a la aplicación ya en funcionamiento.

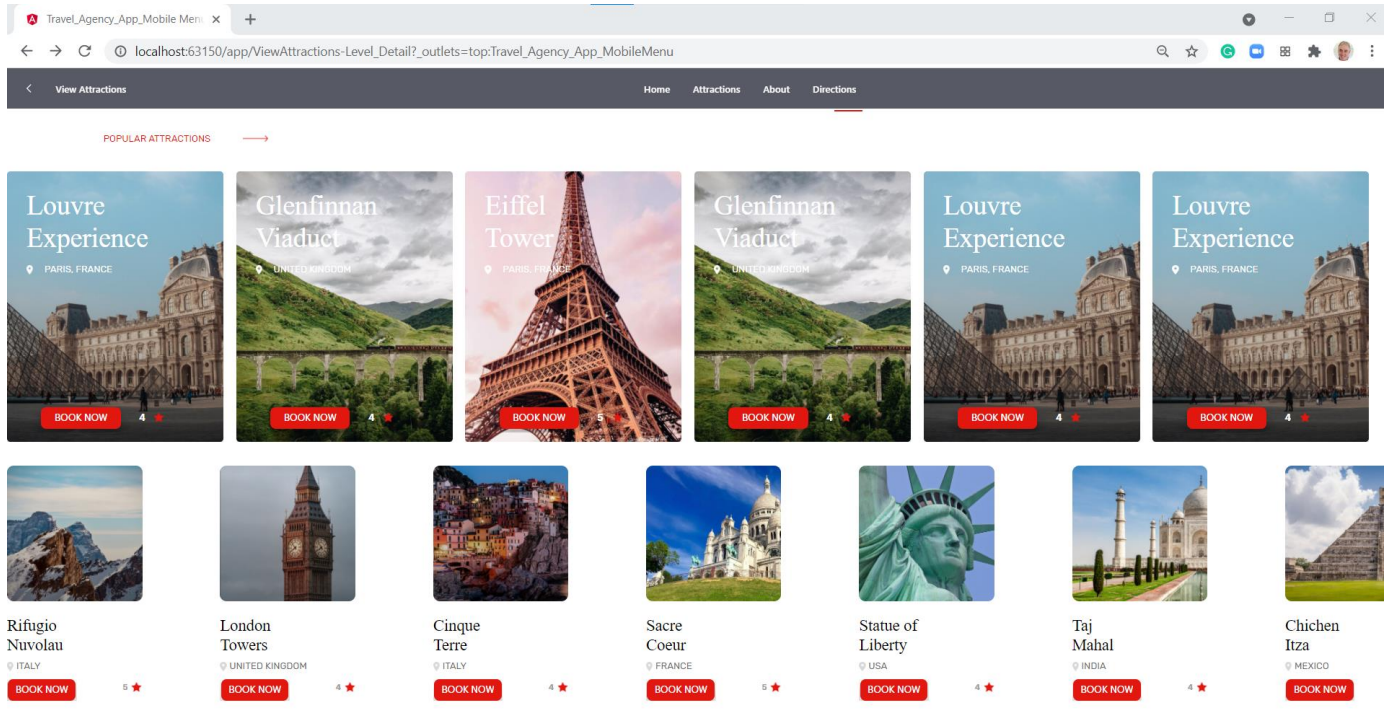


Abrimos GeneXus, presionamos F5 y vemos que se abre el navegador web donde vemos la pantalla inicial de la aplicación.

En la parte superior apreciamos el logo de la agencia y en la parte gris una barra con botones de acceso rápido a los datos de las atracciones. En el centro de la página vemos grandes botones que nos permiten acceder a las distintas partes de la aplicación para ver información de cruceros, vuelos, paquetes turísticos y atracciones.

The screenshot shows a mobile application interface for 'View Attractions'. At the top, there is a navigation bar with the title 'View Attractions' and four menu items: 'Home', 'Attractions', 'About', and 'Directions'. Below the navigation bar, the main content area features the heading 'The new age of EXPLORATION'. Underneath, there are four icons representing different travel services: 'CRUISES', 'FLIGHTS', 'PACKAGES', and 'ATTRACTIONS'. The 'ATTRACTIONS' icon is highlighted with a red underline. Below the icons, there is a section titled 'POPULAR ATTRACTIONS' with a right-pointing arrow. This section displays five attraction cards in a horizontal row. Each card includes the attraction name, location, a 'BOOK NOW' button, and a rating with a star icon. The attractions shown are: 'Louvre Experience' (Paris, France, 4 stars), 'Glenfinnan Viaduct' (United Kingdom, 4 stars), 'Eiffel Tower' (Paris, France, 5 stars), 'Glenfinnan Viaduct' (United Kingdom, 4 stars), and 'Louvre Experience' (Paris, France, 4 stars).

Si hacemos clic en este último botón, accedemos a una página donde podemos navegar horizontalmente entre las atracciones más populares. Para cada atracción vemos su imagen, su ubicación, su calificación y un botón para agendar una visita.



Travel_Agency_App_Mobile Men x +

localhost:63150/app/ViewAttractions-Level_Detail?_outlets=top:Travel_Agency_App_MobileMenu

View Attractions Home Attractions About Directions

POPULAR ATTRACTIONS →

- Louvre Experience PARIS, FRANCE BOOK NOW 4 ★
- Glenfinnan Viaduct UNITED KINGDOM BOOK NOW 4 ★
- Eiffel Tower PARIS, FRANCE BOOK NOW 5 ★
- Glenfinnan Viaduct UNITED KINGDOM BOOK NOW 4 ★
- Louvre Experience PARIS, FRANCE BOOK NOW 4 ★
- Louvre Experience PARIS, FRANCE BOOK NOW 4 ★
- Rifugio Nuvolau ITALY BOOK NOW 5 ★
- London Towers UNITED KINGDOM BOOK NOW 4 ★
- Cinque Terre ITALY BOOK NOW 4 ★
- Sacre Coeur FRANCE BOOK NOW 5 ★
- Statue of Liberty USA BOOK NOW 4 ★
- Taj Mahal INDIA BOOK NOW 4 ★
- Chichen Itza MEXICO BOOK NOW 4 ★

Más abajo en la página vemos otra lista con todas las atracciones para visitar. Aquí también podemos movernos horizontalmente y para cada atracción vemos su país, calificación y botones para fijar una visita.

Travel_Agency_App_Mobile Men x +

localhost:63150/app/ViewLouvre_About-Level_Detail?...outlets=top:Travel_Agency_App_MobileMenu

View Louvre About Home Attractions About Directions

Louvre Experience

Visiting the LOUVRE MUSEUM

ABOUT DIRECTIONS DATES PRICING

The museum packs 30,000 artworks into a 60,000-square-meter exhibition space in three sections: the Denon, Richelieu, and Sully wings. Each wing has more than 70 rooms displaying paintings and objects of art, plus there are enormous halls filled with sculptures. It's impossible to see the entire collection in one day or even in a week. To get the most out of your time spent at the Louvre, focus on a checklist of essential art works using this guide.

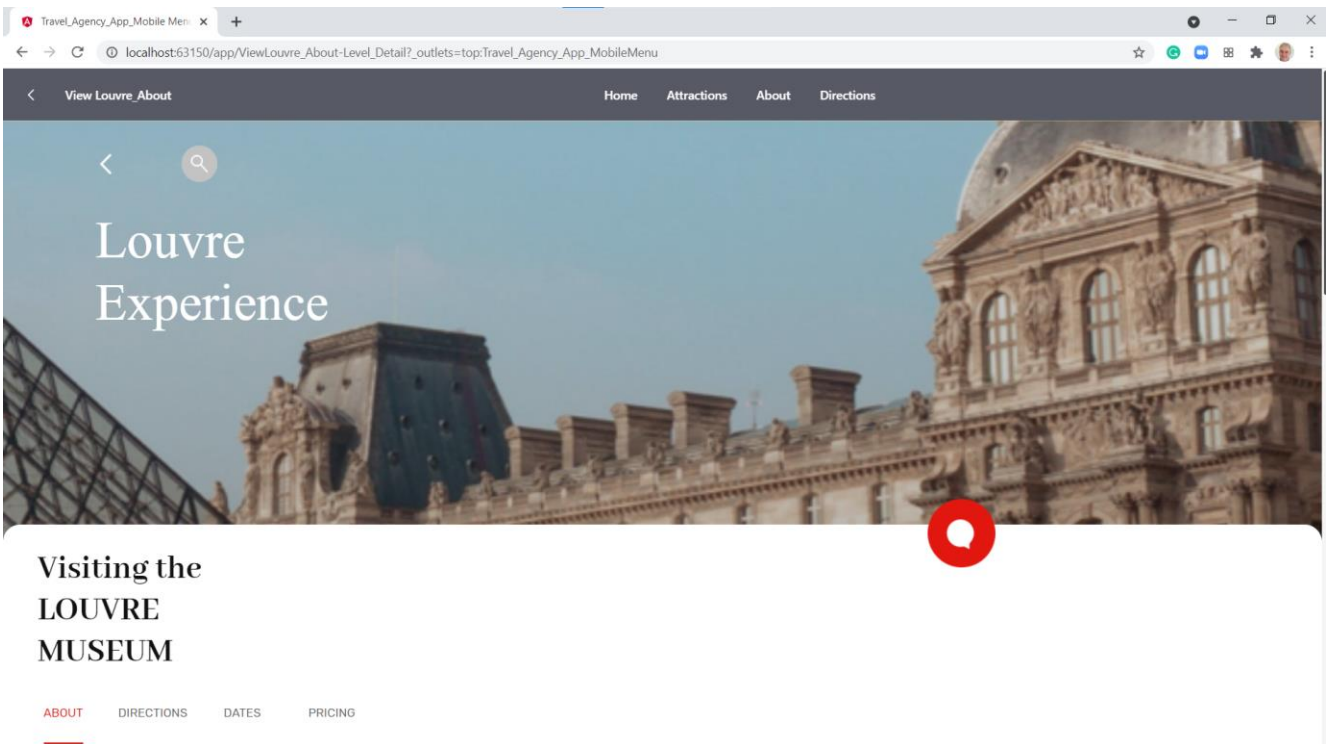
Si hacemos clic en una atracción popular, por ejemplo, en el Louvre, vemos que se abre una página con el detalle de la atracción donde podemos leer una breve descripción con imágenes representativas.

Vemos también un submenú, que además de estos datos generales (marcado con About) podemos acceder a otros datos de la atracción como sus direcciones, fechas de las visitas y sus precios.

The screenshot shows a mobile application interface for the Louvre Museum. At the top, there is a navigation bar with the title "View Louvre_Directions" and menu items: "Home", "Attractions", "About", and "Directions". Below the navigation bar is a header image of the Louvre Museum with a red speech bubble icon in the top right corner. The main content area is titled "Visiting the LOUVRE MUSEUM" and includes a sub-menu with "ABOUT", "DIRECTIONS", "DATES", and "PRICING". The "DIRECTIONS" option is selected. Below the sub-menu, the text reads: "Louvre Museum", "Rue de Rivoli, 75001 Paris, France", "Metro", "Palais-Royal Musée du Louvre (lines 1 and 7) and Pyramides (line 14)", "Bus", and "no. 21, 24, 27, 39, 48, 68, 69, 72, 81, 95". At the bottom, there is a map showing the Louvre Museum location with a red pin. The map includes labels for "Place du Ca...", "Société des Amis du Louvre", "Musée du Louvre - Département des...", "Louvre Museum Landmark art museum", and "Louvre - Rivoli M".

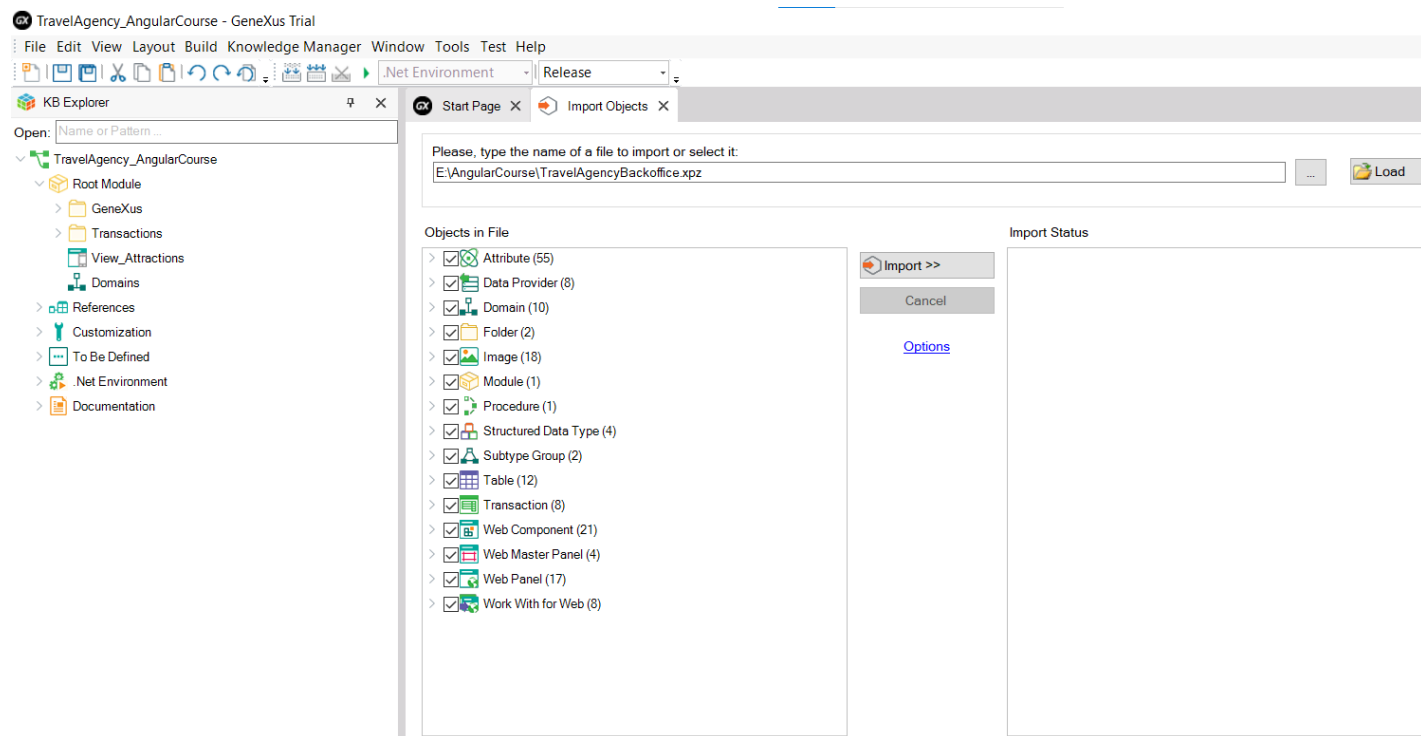
Si seleccionamos direcciones, vemos la calle donde está el museo Louvre, su apartado postal, ciudad y país. También vemos las líneas y paradas del metro y las líneas de buses para acceder al museo.

Mas abajo, vemos en un mapa la localización precisa de la entrada.



Si presionamos el botón de la barra superior que dice Attractions, volvemos a la lista de atracciones. Y si presionamos home, volvemos a la pantalla inicial.

En los siguientes videos vemos como iremos completando estas funcionalidades, al principio sin tener mucho en cuenta cuestiones de aspecto como colores, formas de botones y estilo de fuentes ya que nos concentraremos en el desarrollo de lo que tiene que hacer la aplicación y de su lógica y una vez cumplido esto, veremos cómo importar a nuestra aplicación, las definiciones de estética, llevadas a cabo por un profesional en diseño que nos ayudará a conseguir la apariencia deseada.



Volvamos a GeneXus y empecemos a desarrollar la aplicación que vimos. Lo primero que hacemos es crear una KB de nombre `TravelAgency_AngularCourse`.

Ahora importamos el modelo de datos ya pronto, con todas las transacciones que utilizaremos, a las cuales ya le hemos aplicado el patrón `WorkWith for Web`, por lo que tendremos disponible el backoffice de la aplicación de la agencia de viajes.

Este backoffice web ya está hecho de antes y lo importamos para utilizarlo tal cual está, ya que su implementación no es el objetivo del curso. Lo que vamos a desarrollar durante el curso es el frontend web orientado a customer-facing, en Angular.









Vamos a Knowledge Manager/Import y seleccionamos al archivo `TravelAgencyBackoffice.xpz`

Este xpz contiene además data providers de carga de datos, por lo que cuando ejecutemos se creará la base de datos, con sus tablas ya con datos.

Primero seleccionemos al web panel Home como startup object y ahora presionamos F5 para familiarizarnos con las entidades de la agencia de viajes mientras ejecutamos el backoffice de la aplicación.

Attractions

+ INSERT

Name	Country Name	Category Name	Photo	City Name		
Christ the Redemmer	Brazil	Monument		Rio de Janeiro	UPDATE	DELETE
Cinque Terre	Italy	Tourist site		Maranola	UPDATE	DELETE
Eiffel Tower	France	Monument		Paris	UPDATE	DELETE
Forbidden city	China	Tourist site		Beijing	UPDATE	DELETE
Glenfinnan Viaduct	Scotland	Tourist site		Glenfinnan	UPDATE	DELETE
London Towers	England	Monument		London	UPDATE	DELETE
Long Bridge	United States	Tourist site		San Francisco	UPDATE	DELETE
Louvre	France	Museum		Paris	UPDATE	DELETE

Si vamos a las atracciones turísticas, vemos que atracción pertenece a un país y a una ciudad y tiene información detallada de la atracción.

Si volvemos al Home podemos navegar por los otros datos de la agencia de viajes. Si tiene dudas acerca de los objetos transacción, el patrón work with o los data providers, le sugerimos repasar dichos conceptos en el curso GeneXus Core.

The screenshot shows the GeneXus IDE interface. On the left, a design canvas displays a grid with five columns: 'AttractionId', 'AttractionName', 'AttractionPhoto', 'CityName', and 'CountryName'. The 'MainTable' is visible in the application bar. The Properties window on the right shows the configuration for the 'Panel: View_Attractions', including its name, description, and main program status. The KB Explorer on the right shows the project structure, highlighting the 'Front end' environment. The Properties window for the 'Generator: Frontend (Front end)' shows settings for generating the application, including 'Generate Angular' set to True and 'Main Platform' set to Angular.

Panel: View_Attractions	
Name	View_Attractions
Description	View_Attractions
Module/Folder	Root Module
Qualified Name	View_Attractions
Object Visibility	Public
Main program	True
Master Panel	(none)
Caption	View_Attractions
Miscellaneous	
Network	
Connectivity Support	Online
Deep Link Name	
Main object properties	
Application Title	

Generator: Frontend (Front end)	
Name	Frontend
Generate Android	False
Generate Apple	False
Main Platform	Angular
Dynamic Services URL	False
Services URL	https://trialapp
SSL Pinning Pin Set	
Smart Devices Cache	On
Generate Angular	True
Angular Specific	
Setup Command	npm install
Run Command	npm start
Build Mode	Prototype
Default Platform	Best fit

Volvamos a GeneXus y comencemos con la implementación de la primera pantalla de la aplicación.

La agencia de viajes nos solicita construir un panel que muestre una lista horizontal de las atracciones disponibles y cuando seleccionamos una, ver los detalles de la atracción.

Para eso creamos un objeto Panel, que le ponemos de nombre View_Attractions.

Arrastramos al form un grid y seleccionamos los atributos AttractionId, AttractionName, AttractionPhoto, CityName y CountryName. Vamos a las propiedades del panel y lo seteamos como objeto main.

Ahora en el Knowledgebase Navigator hacemos doble clic en el environment .NET, seleccionamos Front end y vamos a sus propiedades. Seteamos Generate Angular en True y luego Generate Android y Generate Apple en False. Observamos que la propiedad Main Platform quedó con el valor Angular.

Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

New security releases now available for 16.x, 14.x, and 12.x release lines

Download for Windows (x64)

14.17.4 LTS

Recommended For Most Users

16.6.1 Current

Latest Features

<Angular Generator prerequisites for development environment

This documentation is valid for:

[GeneXus Beta](#)

For [Angular applications development](#), your development environment should include the following:

- **Node.js**
Get Node.js from nodejs.org (download the LTS version). After the installation of Node.js, to check your version run the following command in a terminal/console window:

```
node -v
```

Version has to be v12.17 or higher

Antes de generar nuestra aplicación en Angular, debemos instalar algunas aplicaciones. Esto lo podemos ver si vamos al wiki, y buscamos: Angular prerequisites.

Vemos que tenemos que instalar el Node.js desde la página nodejs.org. Aquí descargamos e instalamos la versión LTS. Yo ya lo hice en mi máquina.

For [Angular applications development](#), your development environment should include the following:

- **Node.js**

Get Node.js from nodejs.org (download the LTS version). After the installation of Node.js, to check your version run the following command in a terminal/console window:

```
node -v
```

Version has to be v12.17 or higher

- **npm package manager**

For the environment setup, npm packages will be installed (automatically) using the [npm client](#) command line interface, so you must have an npm package manager. Node.js includes it by default. So, after installing Node.js, check that you have the npm client installed, by running the following command in a terminal/console window:

```
npm -v
```

- **Angular CLI**

To install the [Angular CLI](#) using npm: open a terminal/console window and enter the following command:

```
npm install -g @angular/cli
```

Una vez instalada, para chequear la versión abrimos una ventana de comando y escribimos: **node -v**

Vemos que en mi caso la versión que tengo es la 14.16.1. Para generar Angular con GeneXus, la versión debe ser la 12.17 o superior.

Luego tenemos que instalar un cliente del administrador de paquetes npm, pero como ya tenemos instalado Node.js, esto ya lo incluye. Para verificar que haya quedado correctamente instalado, escribimos: **npm -v** y obtenemos la versión, en mi caso la 6.14.12.

Por último, tenemos que instalar la herramienta de interface de comandos en línea de Angular (la Angular Command Line Interface) y para eso escribimos este comando.

Ahora sí, tenemos todo pronto para generar y ejecutar nuestra aplicación en Angular.

```

Administrator: Windows PowerShell
.69 kB
node_modules_genexus_web-controls-library_dist_esm_gx-map-line_entry_js.js | - | 2
.61 kB
node_modules_genexus_web-controls-library_dist_esm_gx-gauge-range_entry_js.js | - | 2
.30 kB
node_modules_genexus_web-controls-library_dist_esm_gx-query-viewer-parameter_entry_js.js | - | 1
.63 kB
node_modules_genexus_web-controls-library_dist_esm_gx-audio_entry_js.js | - | 1
.59 kB
node_modules_genexus_web-controls-library_dist_esm_gx-query-viewer-element-format_entry_js.js | - | 1
.46 kB
node_modules_genexus_web-controls-library_dist_esm_gx-query-viewer-format-style_entry_js.js | - | 1
.45 kB
node_modules_genexus_web-controls-library_dist_esm_gx-query-viewer-element_entry_js.js | - | 1
.41 kB

Build at: 2021-08-03T13:15:59.371Z - Hash: 9d0ed77ac7a19ddd9fa8 - Time: 17217ms

Warning: E:\Models\TravelAgency_AngularCourse\CSharpModel\mobile\Angular\View_Attractions\node_modules\@genexus\web-standard-functions\dist\lib-esm\types\guid.js depends on 'uuid'. CommonJS or AMD dependencies can cause optimization bailouts.
For more info see: https://angular.io/guide/build#configuring-commonjs-dependencies

** Angular Live Development Server is listening on localhost:54960, open your browser on http://localhost:54960/ **

√ Compiled successfully.
- Generating browser application bundles...

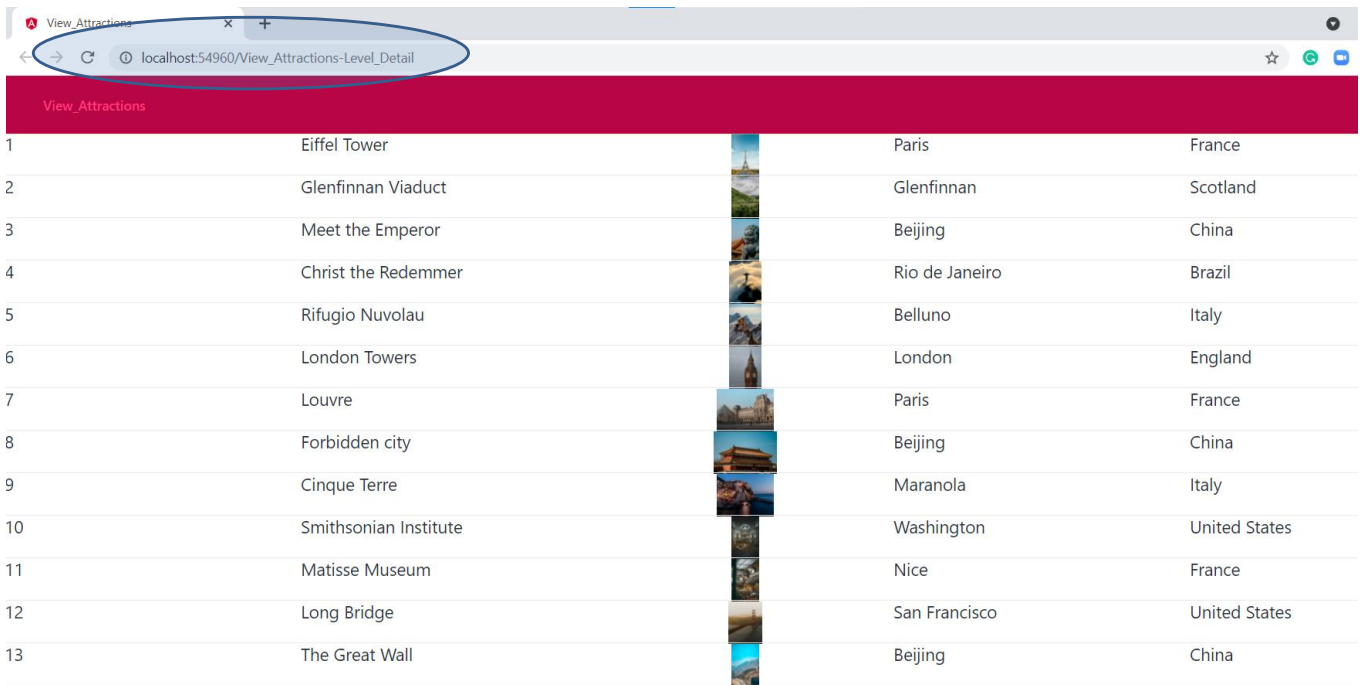
```














Volvemos a GeneXus, vamos al panel View_Attractions, damos botón derecho y seleccionamos Run.

Vemos que se abre una ventana de comandos y que se ejecutan varias líneas escritas por el generador Angular.

Luego de unos minutos esa ventana se cierra y se abre otra ventana de comando que muestra el resultado de la compilación.

Cuando finaliza se abre automáticamente el navegador web y vemos la lista de atracciones turísticas.



View_Attractions				
1	Eiffel Tower		Paris	France
2	Glenfinnan Viaduct		Glenfinnan	Scotland
3	Meet the Emperor		Beijing	China
4	Christ the Redemmer		Rio de Janeiro	Brazil
5	Rifugio Nuvolau		Belluno	Italy
6	London Towers		London	England
7	Louvre		Paris	France
8	Forbidden city		Beijing	China
9	Cinque Terre		Maranola	Italy
10	Smithsonian Institute		Washington	United States
11	Matisse Museum		Nice	France
12	Long Bridge		San Francisco	United States
13	The Great Wall		Beijing	China

Ya tenemos nuestra primera aplicación Angular funcionando.

Si observamos la URL vemos que se ejecutó en localhost, dos puntos y un puerto asignado dinámicamente. Esto corresponde al servidor web instanciado para ejecutar la aplicación Angular. La aplicación ejecutada es View_Attractions-Level_Detail, que corresponde al grid de atracciones que incluimos en el panel.

View_Attractions x Navigation View x

Pattern:

View_Attractions
Level_Detail_Grid1

Data Provider View_Attractions_Level_Detail_Grid1 Navigation Report

Name: View_Attractions_Level_Detail_Grid1
Description: View_Attractions_Level_Detail_Grid1
Status: No Generation Required
Output Devices: None

Environment: C# Default (C#)
Spec. Version: 17_0_4-151606
Form Class: HTML
Program Name: View_Attractions_Level_Detail_Grid1
Parameters: in: &start, in: &count, in: &gxid, out: View_Attractions_Level_Detail_Grid1Sdt

LEVELS

For Each Attraction (Line: 2)

Order: AttractionId
Index: IATTRACTION
Navigation filters: Start from: FirstRecord
 Loop while: NotEndOfTable
Join location: Server
Optimizations: Server Paging

Attraction (AttractionId)	INTO	AttractionPhoto Uri
AttractionId	AttractionName	AttractionPhoto CountryId CityId
Country (CountryId)	INTO	CountryName
CountryCity (CountryId, CityId)	INTO	CityName

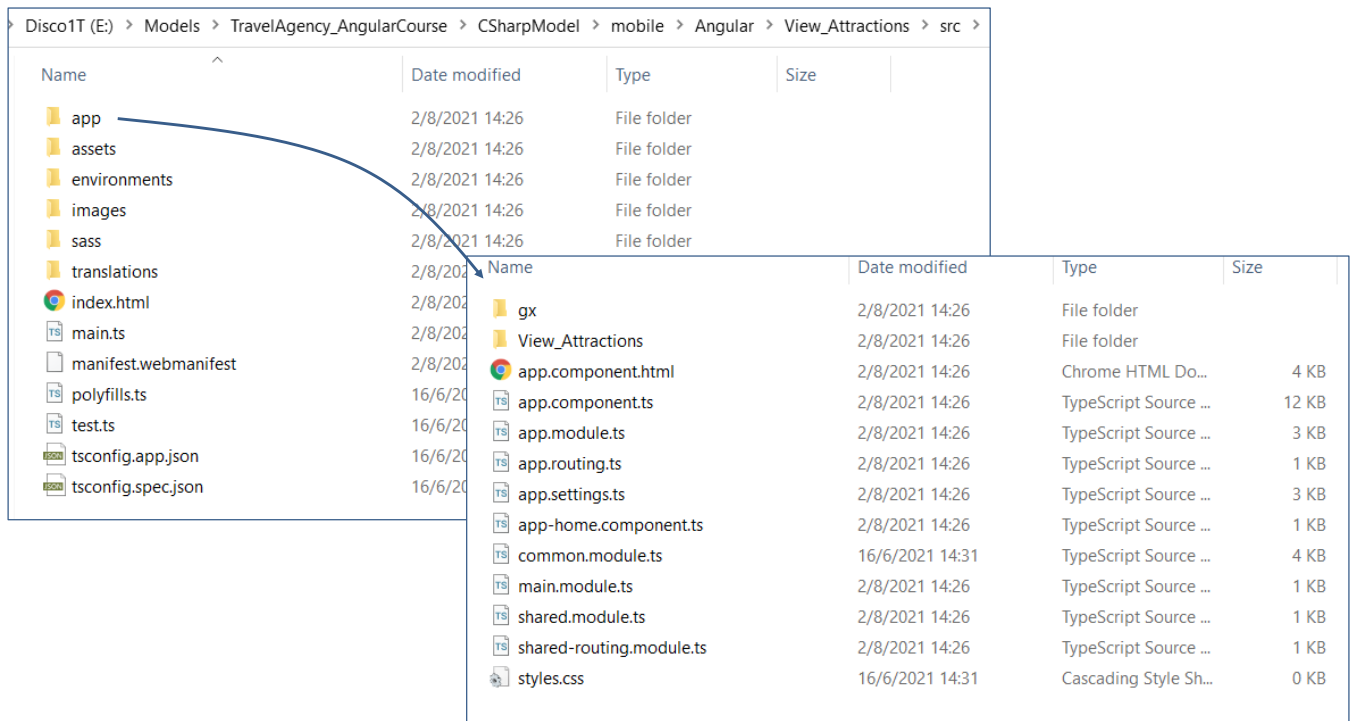
Si volvemos a GeneXus y vemos el listado de navegación, vemos que el panel View_Attractions contiene un nodo llamado Level_Detail_Grid1 y que tiene el símbolo de un data provider.

Si hacemos clic sobre el mismo, vemos que es el listado de navegación del data provider View_Attractions_Level_Detail_Grid1 que es el responsable de acceder a la base de datos y recuperar la información de las atracciones turísticas.

Si vemos donde dice Levels, encontramos un For Each Attraction ya que como nuestra grid tenía atributos, GeneXus encontró una tabla base y creó un for each implícito para acceder a los datos de la base de datos.

Vemos que efectivamente se navegó la tabla Attraction para traer los datos de la atracción y luego se accedió a la tabla Country para recuperar el nombre del país y a la tabla CountryCity para obtener el nombre de la ciudad.

Más adelante veremos qué toma en cuenta GeneXus para determinar la tabla base del grid y cómo se carga la información a través del data provider cuyo listado de navegación estamos viendo.

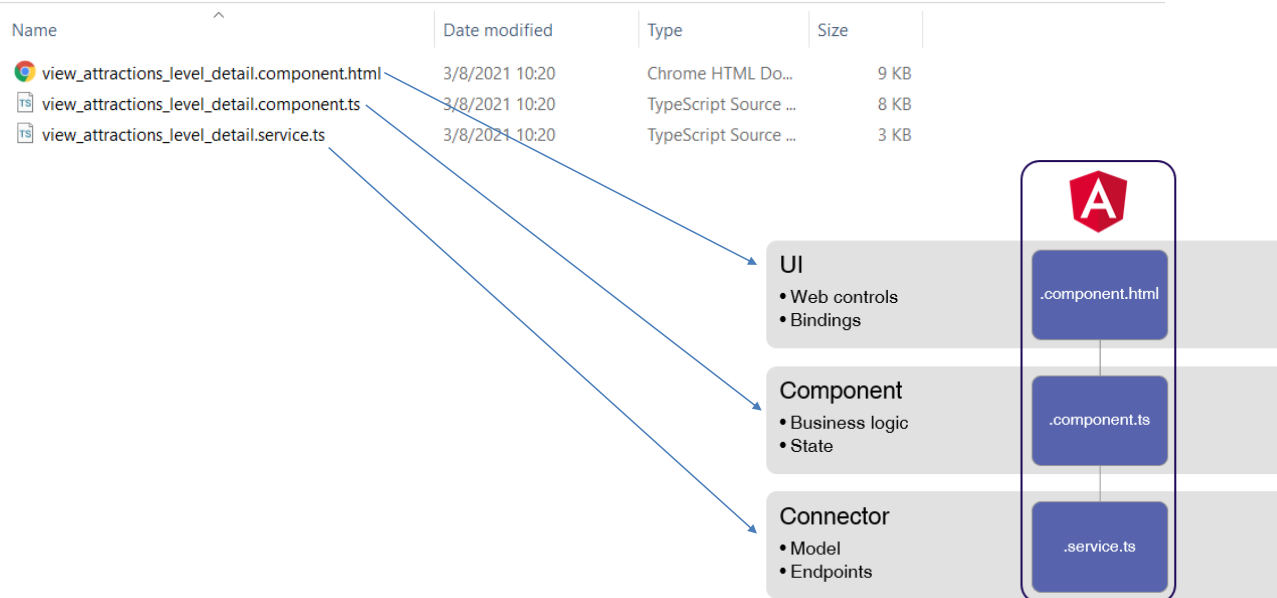


Analizamos ahora qué genera GeneXus, cuando se ejecuta el generador Angular.

Si vamos a Tools/Explore Target Environment y luego vamos a mobile/Angular, encontramos una carpeta View_Attractions. Esto es porque es el objeto main que ejecutamos. Si ahora seleccionamos src/app, vemos las partes que integran la aplicación.

Si hacemos clic en View_Attractions vemos las partes que incluyen al componente Angular, que son 3 archivos.

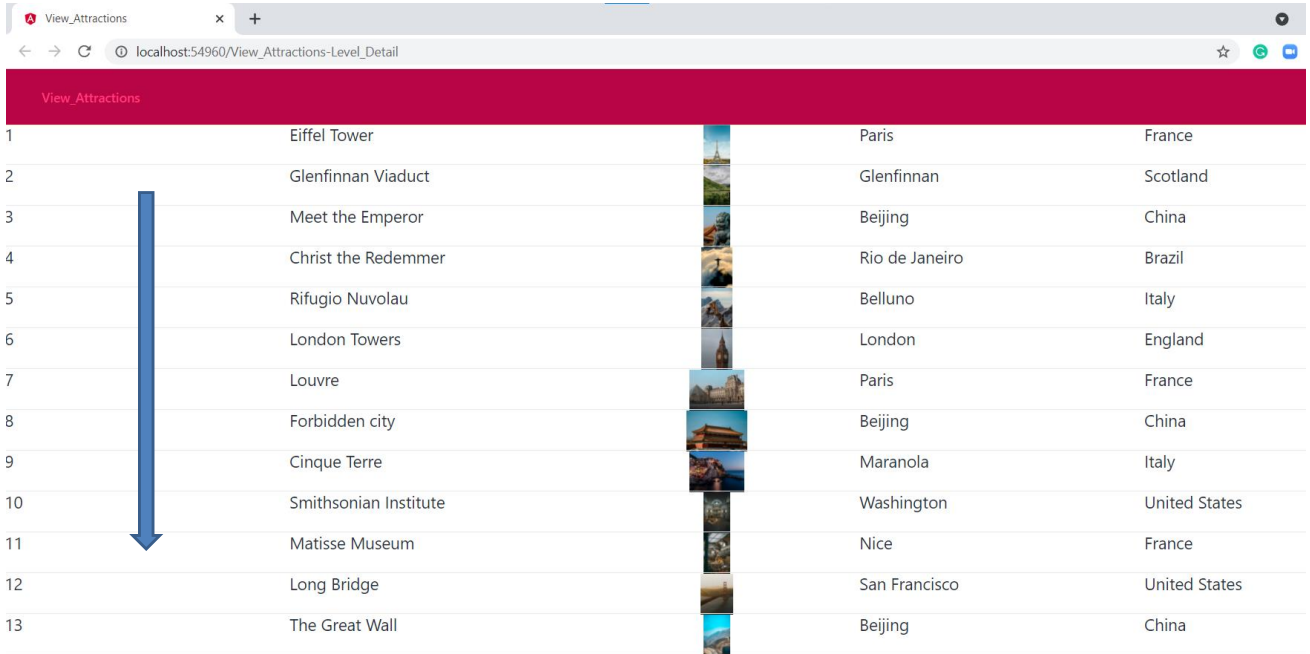
Disco1T (E:) > Models > TravelAgency_AngularCourse > CSharpModel > mobile > Angular > View_Attractions > src > app > View_Attractions
















El archivo que termina en **components.html** contiene la definición de la interfase de usuario de la aplicación y es un template de Angular que representa el layout del panel. Aquí también se incluyen elementos que Angular interpreta para asociar los controles en pantalla con los datos de la base de datos.

Luego tenemos al archivo que termina en **components.ts**, que es un archivos TypeScript donde está definida la lógica del panel que implementamos. Aquí es donde están definidos los eventos y el estado de los controles de la pantalla.

Por último, tenemos al archivo que termina en **service.ts** que es el responsable de interactuar con los servicios del servidor web. Aquí se producen las comunicaciones con los servicios REST, se obtiene los datos de la base de datos y se generan las estructuras de datos que el panel va a utilizar.



The screenshot shows a web browser window with the URL `localhost:54960/View_Attractions-Level_Detail`. The page title is `View_Attractions`. The table displays 13 rows of attraction data, each with a number, name, image, location, and country.

Number	Attraction Name	Image	Location	Country
1	Eiffel Tower		Paris	France
2	Glenfinnan Viaduct		Glenfinnan	Scotland
3	Meet the Emperor		Beijing	China
4	Christ the Redemmer		Rio de Janeiro	Brazil
5	Rifugio Nuvolau		Belluno	Italy
6	London Towers		London	England
7	Louvre		Paris	France
8	Forbidden city		Beijing	China
9	Cinque Terre		Maranola	Italy
10	Smithsonian Institute		Washington	United States
11	Matisse Museum		Nice	France
12	Long Bridge		San Francisco	United States
13	The Great Wall		Beijing	China

Si volvemos a la pantalla de atracciones, vemos una grilla que muestra los datos de las atracciones hacia abajo, que es la modalidad por defecto.

En el desarrollo del curso, vamos a ver cómo cambiar esta visualización para poder paginar hacia la derecha, así como otros controles de pantalla que nos permitirán construir la aplicación con una interfaz de usuario más atractiva.

GeneXus™

training.genexus.com
wiki.genexus.com
training.genexus.com/certifications