

# Prototipación y Puesta en producción de una aplicación Angular

GeneXus™

Veamos a continuación distintas formas de prototipar una aplicación Angular y cómo ponerla en producción, es decir cómo hacer el despliegue para que quede funcionando en las instalaciones del cliente.

# Prototipación de una aplicación Angular

La etapa en la que comenzamos a construir la aplicación y ajustamos su funcionamiento de acuerdo a los requerimientos solicitados, se denomina Prototipación.

Como una aplicación Angular tiene un front-end y un back-end, veamos a continuación las opciones de prototipación que disponemos para cada parte.

Comencemos por la prototipación del back-end.

## Configuración del back-end

Generator: Default (.NET Framework)

Name	Default
User Interface	Web

> General

> Services

> Data Access Information

> Event Handling

> Management

> User interface

> Specification

> Web information

> Build Process

> Execution

Deploy to cloud	Yes
Deploy Server URL	https://trialapps3.genexus.com
Deploy Virtual Direct	Id450b84f44dd60d0d14afee61c...
IIS Version	IIS8 or higher
Web Root	https://trialapps3.genexus.com/Id4...

Execution

Deploy to cloud	No
Web Server	Internet Information Server
IIS Version	IIS8 or higher
Web Root	https://localhost/TravelAgency_...

Net Environment

- Back end
  - Default (.NET Framework)
  - Data Stores
  - Services
- Front end
  - Web (.NET Framework)
  - Web (Angular)
- Deployment

Hasta ahora hemos utilizado los mecanismos por defecto para prototipar. Como usamos GeneXus Trial, la parte del backend se prototipa automáticamente en la nube de GeneXus y no podemos cambiar el valor de la propiedad Deploy to Cloud.

Si tuviéramos la versión Full podríamos elegir el valor de esta propiedad y prototipar en la Nube con el valor Yes, o poner la propiedad Deploy to cloud en No y prototipar en un servidor web instalado en nuestra máquina local, con el DBMS instalado también localmente, por lo que el acceso a los datos de la base de datos, será únicamente en nuestro computador.

## Configuración del front-end

Generator: Frontend (Front end) - Angular Specific	
Setup Command	npm install -f
Run Command	npm start
Build Mode	Prototype
Default Platform Hint	Best fit
Deploy Command	npm run deploy
Run Target	Default

Para el front-end, vemos que por defecto se instancia un servidor local (localhost) en un puerto que se asigna automáticamente.

Sin embargo, tenemos otras posibilidades para hacer la prototipación.

Si vamos a las propiedades del generador Angular (en las propiedades del environment / Front end), vemos que tenemos disponible una propiedad Run Target. Si presionamos el combo vemos varios valores posibles:

**Default:** Por defecto la aplicación Angular se ejecuta en el Angular Dev Server, en forma local (localhost) y en un puerto aleatorio. En un futuro, cuando se seleccione Deploy To Cloud en el Backend, la opción Default se comportará como GeneXusDeployToCloud.

**Angular Dev Server:** Es el servidor Angular Http Server de la Plataforma Angular que corre localmente en la máquina de desarrollo. Es un servidor simple diseñado para desarrolladores de aplicaciones de página única (SPA).

**Local Web Server:** La aplicación se transpila (es decir se traduce de un lenguaje fuente a otro lenguaje fuente, en este caso de TypeScript a JavaScript) y se copia al servidor web local del Environment GeneXus (por ejemplo Tomcat en Java o Internet Information Server en .Net). La aplicación es ejecutada directamente bajo la URL del backend web.

**GeneXusDeployToCloud:** Indica que la aplicación va a ser subida a la

nube de DeployToCloud. En este caso la aplicación se ejecutará en un servidor AWS S3 (Amazon Simple Storage Service) que es un servicio ofrecido por Amazon Web Services que proporciona almacenamiento de objetos a través de una interfaz de servicio web y son accedidas mediante la red de entrega de contenido (CDN) de Amazon Cloudfront. Éste es un servicio web que agiliza la distribución de contenido web estático y dinámico como archivos .html, .css, .js y archivos de imágenes, a los usuarios.

**AWS S3:** Prepara el proyecto angular para ser subido **manualmente** a AWS S3 mediante línea de comandos.

## Prototipación del front-end en la nube

The screenshot shows a web browser displaying a travel agency prototype. The browser's address bar shows the URL: `apps-angular.genexus.com/ld8babb83c36540ea021a6ee5f4f586435/View_Attractions_MoreInfo/View_Attractions_MoreInfo-Level_Detail`. The page header includes a red suitcase icon and the text "TRAVEL AGENCY". Below the header, there are four image cards representing travel destinations:

- Eiffel Tower**: Paris, France
- Glenfinnan Viaduct**: Glenfinnan, Scotland
- Meet the Emperor**: Beijing, China
- Christ the Redemmer**: Rio de Janeiro, Brazil

On the left side of the browser window, there is a "Properties" panel for the "Generator: Frontend (Front end) - Angular Specific". The "Run Target" property is set to "GeneXus DeployToCloud".

Property	Value
Setup Command	npm install -f
Run Command	npm start
Build Mode	Prototype
Default Platform Hint	Best fit
Deploy Command	npm run deploy
Run Target	GeneXus DeployToCloud

Vamos a probar ahora prototipar el front-end en la nube, en lugar de en nuestro servidor local.

Para eso, cambiamos el valor de la propiedad Run target, del valor Default que usamos, al valor GeneXus DeployToCloud y ejecutamos el panel View\_Attractions\_MoreInfo.

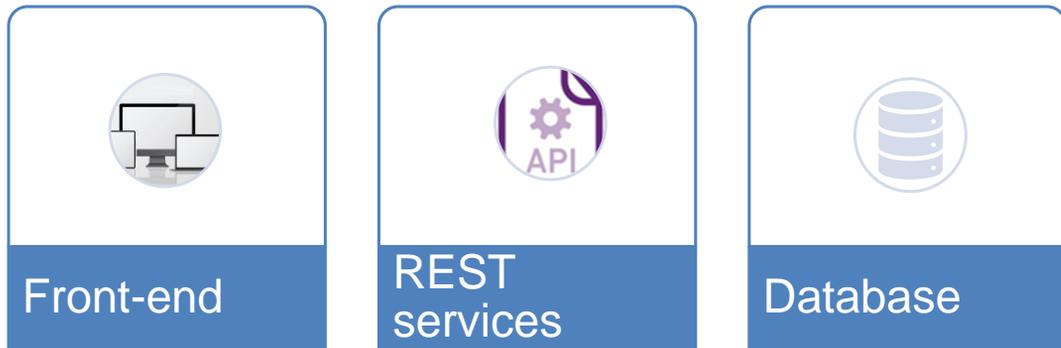
Vemos que ahora, no se prototipa más en forma local en nuestro computador ya que la url no dice localhost, sino que se está prototipando en la nube. Esto hace que no sea necesario instalar el software para instanciar el servidor web local para Angular.

Esta opción permite que cuando se prototipe el backend en la nube, también el front-end pueda prototiparse en la nube, de forma que no tenga que instalarse ningún recurso en la máquina local, facilitándose el setup del entorno de desarrollo.

# Puesta en producción de una aplicación Angular

Una vez que terminamos de desarrollar la aplicación en nuestro computador, debemos ponerla a funcionar en las instalaciones del cliente, ya sea un cliente interno (dentro de nuestra propia empresa) o un cliente externo. En nuestro ejemplo sería en los servidores de la agencia de viajes.

## Etapas de la puesta en producción de una aplicación en Angular



El proceso de puesta en producción de una aplicación en Angular involucra tres etapas:

- La distribución de los objetos del front-end a ser desplegados en el servidor HTTP
- La distribución de los servicios REST a ser desplegados en el servidor de aplicaciones
- La distribución de las estructuras de la base de datos

## Despliegue del Front-end

Comencemos por el front-end

## Despliegue del front-end

```

C:\Users\rroballo>
E:\>cd \Models\TravelAgency_AngularCourse\CSharpModel\mobile\Angular\ViewHome
E:\Models\TravelAgency_AngularCourse\CSharpModel\mobile\Angular\ViewHome>ng build --configuration
production
Your global Angular CLI version (12.1.4) is greater than your local version (12.0.2). The local Angular
CLI version is used.

To disable this warning use "ng config -g cli.warnings.versionMismatch false".
* Browser application bundle generation complete.
* Copying assets complete.
* Index html generation complete.
* Service worker generation complete.

Initial Chunk Files | Names | Size
main.2d913552e09bd696904a.js | main | 559.57 kB
polyfills.24ae5569982c0b2a2934.js | polyfills | 55.52 kB
carminesd.06808c03de34eac34c35.css | carminesd | 18.83 kB
| Initial Total | 633.92 kB

Lazy Chunk Files | Names | Size
875.ad6d136803ac7291b088.js | - | 228.66 kB
772.7473e439feb0e13bc897.js | - | 177.50 kB
11.5c6585f38edbf97f5b05.js | - | 153.83 kB
849.30a5b9eb1b70cf436bce.js | - | 135.72 kB
carmineios.css | carmineios | 35.33 kB
carmineandroid.css | carmineandroid | 29.48 kB

```

Disco1(T) > Models > TravelAgency\_AngularCourse > CSharpModel > mobile > Angular > ViewHome

Name	Date modified	Type
dist	15/11/2021 11:00	File folder
node_modules	15/11/2021 10:35	File folder
scripts	15/11/2021 10:33	File folder
src	15/11/2021 10:33	File folder
test	15/11/2021 10:33	File folder



Generator: Frontend (Front end) - Angular Specific	
Setup Command	npm install -f
Run Command	npm start
Build Mode	Distribution
Default Platform Hint	Prototype
Deploy Command	Development
Run Target	Distribution

La puesta en producción de los objetos del front-end se realiza a través del Angular Command Line Interface.

Para eso abrimos una ventana de comando, nos posicionamos en la carpeta de la KB donde está nuestro objeto main (en nuestro caso ViewHome) y escribimos el comando: `ng build --configuration production`. Este proceso crea, bajo la carpeta donde está el objeto main, una carpeta "dist" que tiene todo lo necesario para el front-end.

En lugar de escribir una línea de comando, podemos hacer esto directamente desde el IDE de GeneXus.

Si vamos a las propiedades del generador Angular, vemos que está la propiedad Build Mode que permite que el desarrollador configure el modo en que va a ejecutarse la aplicación Angular.

El valor por defecto es Prototype, en la que se importan a nuestro proyecto las mínimas dependencias necesarias para que ejecute la aplicación Angular.

En el modo Development se importan además paquetes que sirven para testing, debug y nos ayudan en el desarrollo de la aplicación.

Y en el modo Distribution es cuando queremos generar los paquetes para hacer un deploy de la aplicación. Al seleccionar este modo, se ejecutará automáticamente el comando `ng build --configuration production`, que vimos antes.

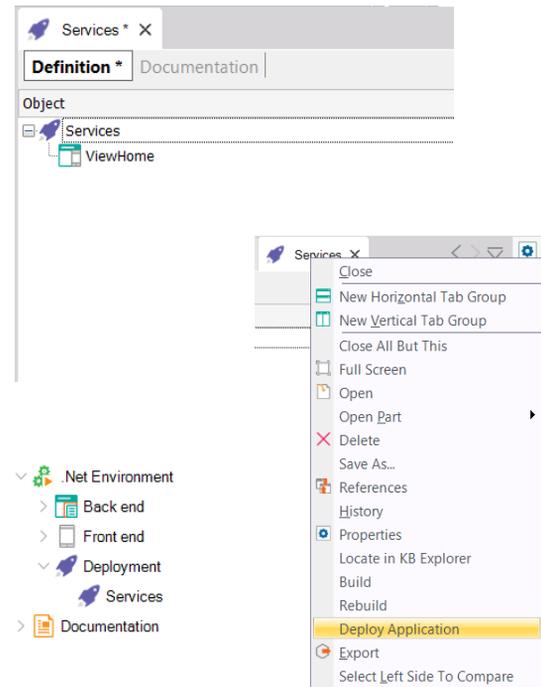
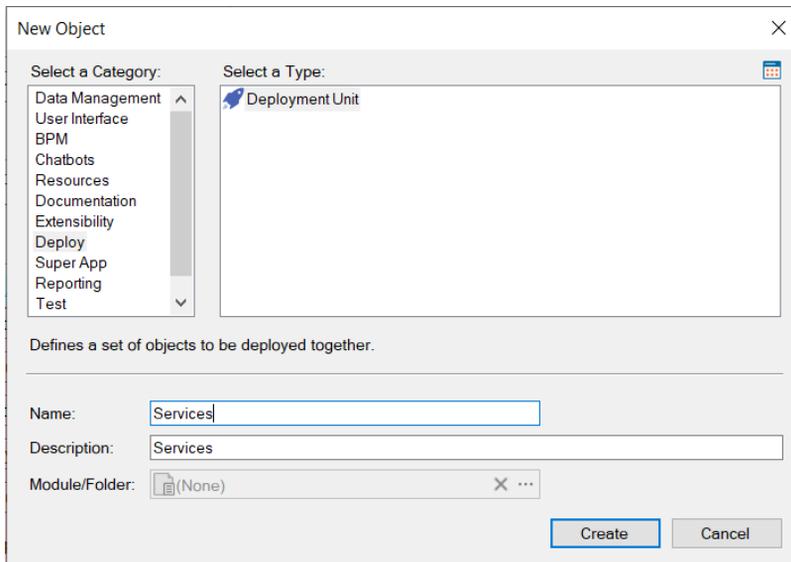
Luego de que se realiza el proceso de distribución y se genera la carpeta "dist", se copia esta carpeta completa al servidor HTTP del cliente, por ejemplo, al Internet Information Services o al Apache Tomcat.

## Despliegue de los servicios REST

Veamos ahora como desplegamos los servicios REST, a través del objeto main.

---

## Despliegue de los servicios REST

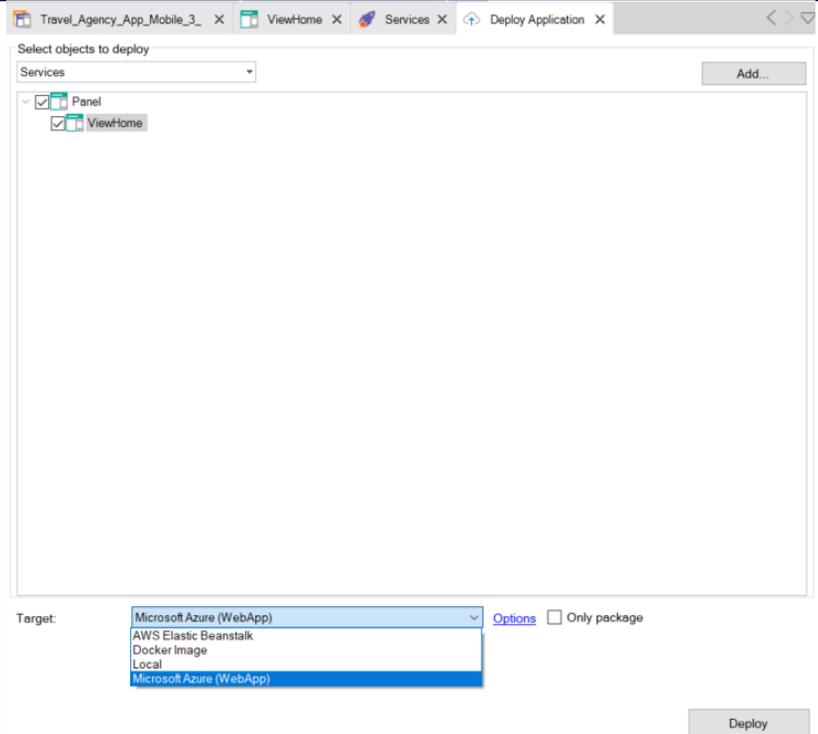
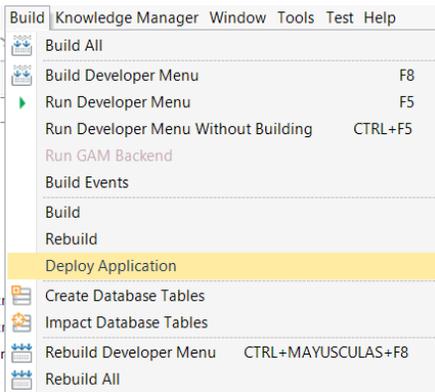


El despliegue de los servicios REST, se hace creando un objeto Deployment Unit, al cual le ponemos por ejemplo el nombre Services y luego arrastramos sobre nodo Services a nuestro objeto main.

Si vamos al KB Explorer vemos que bajo el nodo Deployment del environment, apareció la deployment unit que creamos.

Para hacer el despliegue damos botón derecho sobre la deployment unit y elegimos Deploy application.

## Despliegue de los servicios REST



Otra opción es usar la Application Deployment Tool a la que accedemos desde el menú Build, eligiendo Deploy Application. Vemos que automáticamente quedará seleccionada la deployment unit Services, que ya contiene a nuestro objeto main.

Luego con el combobox Target elegimos el tipo de despliegue que haremos (local, a una imagen de Docker, o a un servidor externo, como por ejemplo Microsoft Windows Azure o AWS Elastic Beanstock) y presionamos el botón Deploy.

## Despliegue de la base de datos

La distribución de la base de datos se realiza exportando la reorganización al servidor de base de datos de producción.

## Despliegue de la base de datos

The image shows the GeneXus Build menu and the Export Reorganization dialog box. The Build menu is open, showing options like Build All, Build ViewHome, Run ViewHome, Run ViewHome Without Building, Run GAM Backend, Build Events, Build, Rebuild, Run, Run Without Building, Run With This Only, Build With This Only, Set As Startup Object, Create Database Tables, Impact Database Tables, Rebuild ViewHome, Rebuild All, Deploy Application, Export Reorganization (highlighted), and Deploy through GeneXus Cloud. The Export Reorganization dialog box is open, showing the filename 'Reorganization\_20211115144621.zip', user 'umqfgYcRu2H2hU0p', and password '\*\*\*\*\*'. The 'Zip it!' button is visible. The Output window shows the following text:

```

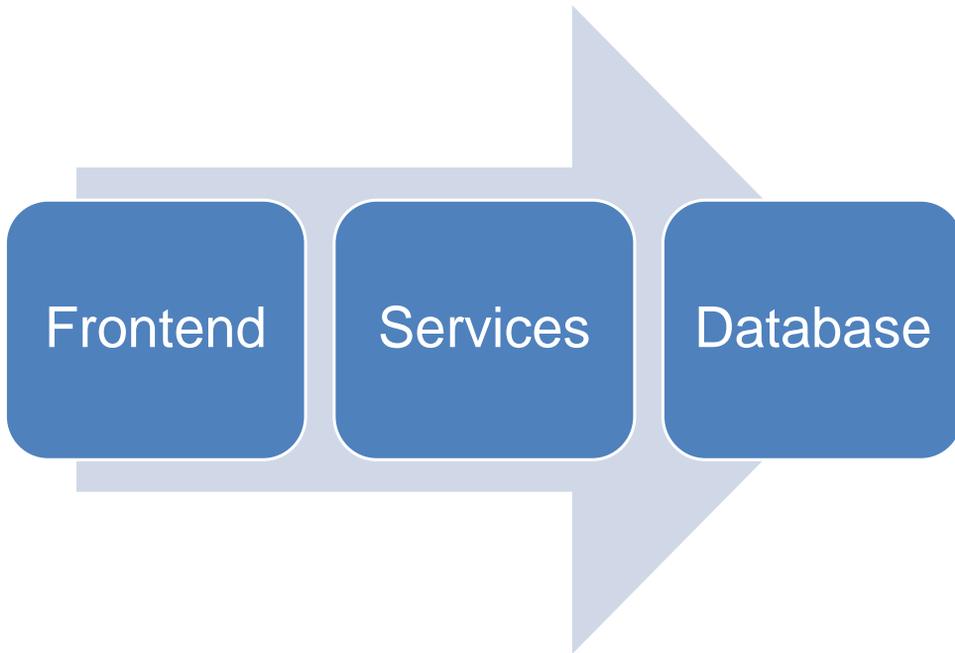
===== Export Reorganization started =====
Reorganization successfully exported at 'E:\Models\TravelAgency_AngularCourse\CSharpModel\Reorgs\Reorganization_20211115144621\Reorganization_20211
Success: Export Reorganization
  
```

Durante el proceso de Build, GeneXus crea (y por defecto ejecuta) los programas de reorganización.

Para exportar la reorganización, podemos ir a Build / Export reorganization y esto crea un paquete (archivo zip para .NET, como el del ejemplo y .jar para Java) con los archivos necesarios para ejecutar los programas de reorganización, que se han creado en el último proceso de compilación.

Ese paquete se puede enviar al servidor de base de datos de producción para que se descomprima y ejecute allí, para crear o reorganizar las estructuras de la base de datos correspondientes.

Puede obtener más datos sobre cómo exportar reorganizaciones buscando el artículo de nombre "Export Reorganization" en el Wiki: <https://wiki.genexus.com/commwiki/servlet/wiki?34476>



En este video vimos como se puede poner en producción una aplicación generada en Angular, en sus tres componentes, el front-end, los servicios rest y las reorganizaciones de la base de datos.

La puesta en producción es parte del proceso de desarrollo, especialmente en un ciclo de desarrollo DevOps, donde es posible automatizar este proceso junto otros que permiten tener la nueva versión de la aplicación funcionando en producción, en el menor tiempo posible, de forma ágil y continua.

# GeneXus™

[training.genexus.com](http://training.genexus.com)  
[wiki.genexus.com](http://wiki.genexus.com)  
[training.genexus.com/certifications](http://training.genexus.com/certifications)