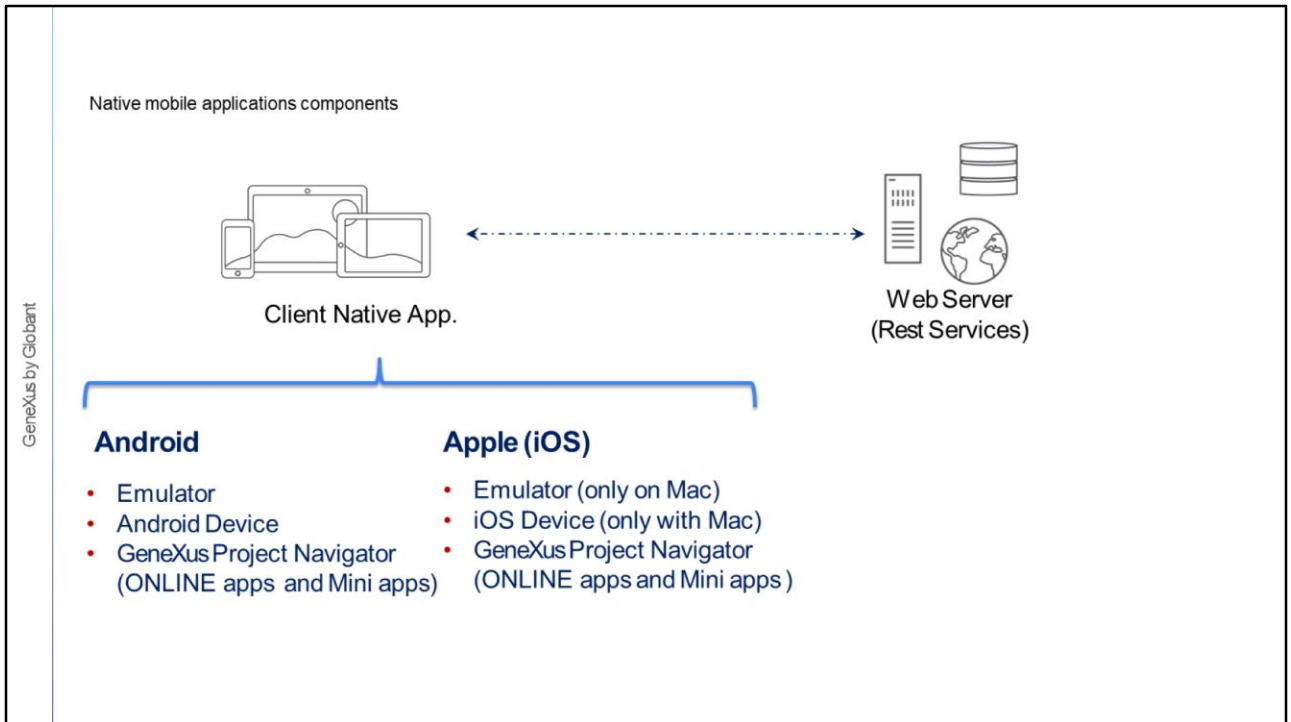


Prototyping a mobile native app



Rodolfo Roballo

En este video veremos cómo podemos probar nuestra aplicación Android o Apple en un emulador, en un dispositivo físico, o utilizando la aplicación GeneXus Project Navigator y conocer las ventajas y desventajas de cada una de estas posibilidades.



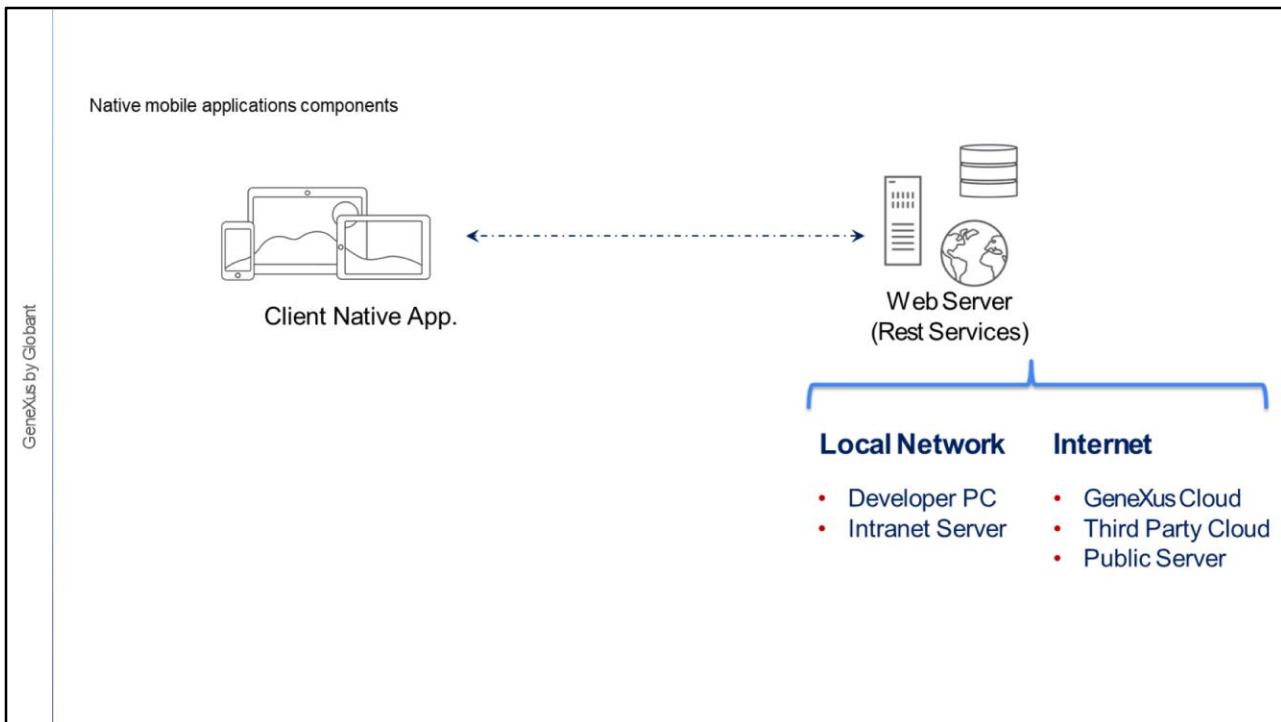
De acuerdo a lo que hemos visto cuando estudiamos la arquitectura de una aplicación nativa, para ejecutar una aplicación necesitamos dos componentes, la aplicación mobile nativa propiamente dicha y los servicios REST que atienden los pedidos de la aplicación, alojados en un servidor web.

En el caso de Android la aplicación cliente puede correr en un emulador como venimos utilizando en el curso, o podemos ejecutarla en un dispositivo físico Android estándar, o un dispositivo Android Huawei, conectados a nuestra computadora Windows a través de un puerto USB. En ambos casos la aplicación es compilada antes de ser ejecutada y el archivo .apk es instalado en el emulador o en el dispositivo físico mediante la conexión USB.

Pero también podemos ejecutar la aplicación sin necesidad de compilarla, utilizando el GeneXus Project Navigator. El GPN es una aplicación nativa hecha por GeneXus que podemos descargar desde el Google Play, instalarla en un dispositivo Android estándar y prototipar nuestra aplicación en forma interpretada sin necesidad de que el dispositivo esté conectado físicamente a nuestro computador de desarrollo. Este método no se utiliza para Huawei ya que para que la app acceda a los servicios propios de la plataforma, como Huawei Analytics, Maps, Notifications, etc., se requiere de una clave y ésta solamente se obtiene con la aplicación compilada.

También tenemos la limitación que con el GPN solamente podemos prototipar aplicaciones ONLINE. La buena noticia es que sí podemos prototipar Mini Apps. Hablaremos un poco más del GPN en unos momentos.

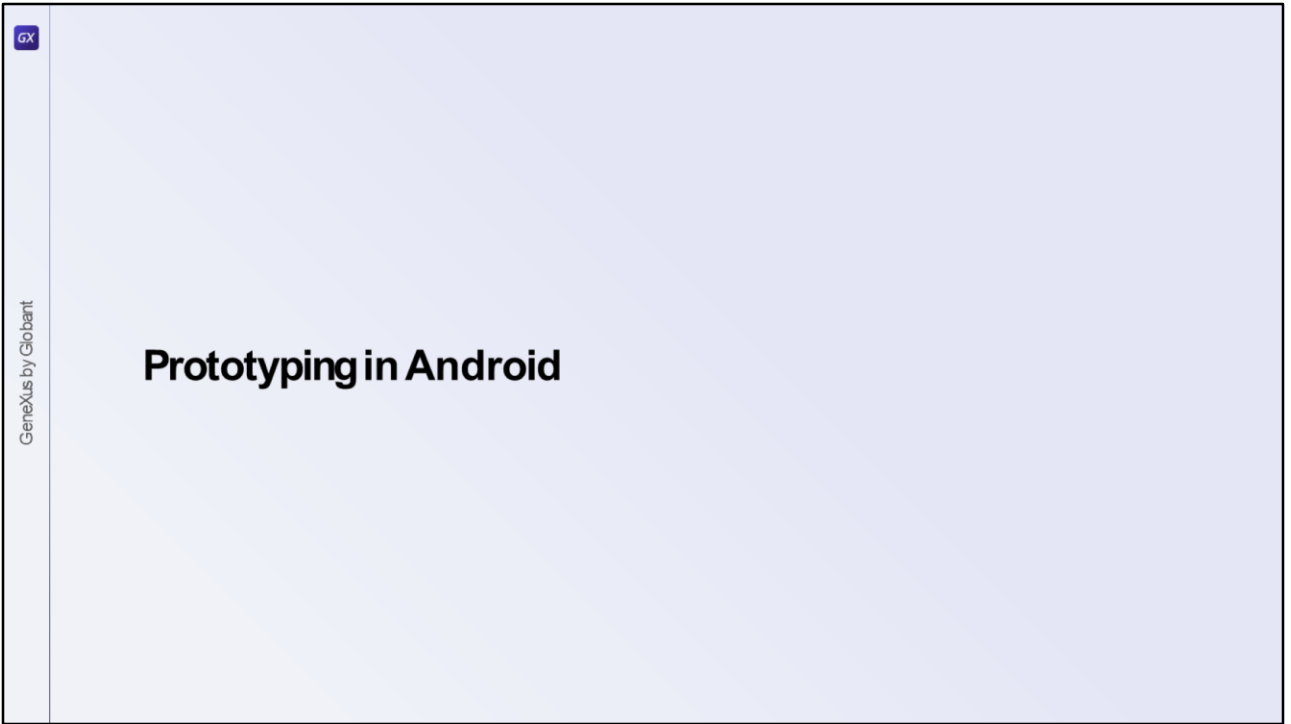
Para Apple, la aplicación puede correr también en un emulador o en un dispositivo físico pero necesitamos una computadora Mac para compilar y para ejecutar el emulador o para conectar un dispositivo físico vía USB. Además también en este caso contamos con la posibilidad de ejecutar la aplicación en forma interpretada en un dispositivo con iOS usando el GPN, (siempre y cuando la aplicación sea Online o Mini App) y en este caso no necesitamos una Mac, solo contar con un iPhone o iPad con el GPN instalado desde el Apple Store.



Respecto a los servicios, estos pueden correr en la red local, ya sea en la computadora del desarrollador o en algún otro servidor web disponible en la red; o pueden correr en un servidor en internet. En este caso el servidor puede ser el de la nube de GeneXus como estamos utilizando en el curso, siendo ésta la única posibilidad cuando usamos la versión trial o si estamos usando GeneXus Full los servicios REST pueden alojarse en una nube de terceros como la que provee Amazon o Google, o podemos publicar los servicios en un servidor propio que tenga acceso público.

Lo importante es que la aplicación mobile cliente tenga acceso a estos servicios, es decir, que pueda conectarse por wifi al servidor donde los servicios están instalados.

A continuación vamos a ver cada una de las alternativas y las cuestiones que deberemos considerar en cada caso.



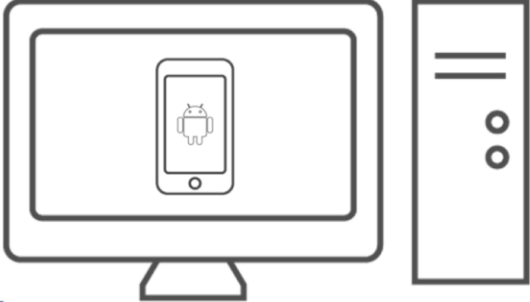
Hasta ahora hemos ejecutado la aplicación nativa en Android utilizando el emulador.

Veamos un poco más en detalle los mecanismos de prototipado en esta plataforma.

GeneXus by Globant

Prototyping in Android on Emulator

Generator: Frontend (Front end)	
Name	Frontend
Main Platform	Android
Dynamic Services URL	False
Services URL	https://trialapps3.genex...
SSL Pinning Pin Set	
Smart Devices Cache Management	On
Visual Testing Repository URL	
Generators	
Generate Android	True
Generate Huawei	False
Generate Apple	True
Generate Angular	False
Android Specific	
Android Execution Type	Emulator or Device
Android SDK directory	C:\Android-SDK
JDK Directory	C:\Program Files\Java\j...



Services URL

- Local Network
- Wifi or Lan assigned IP Address

Para generar en Android, como vimos en videos anteriores, solo debemos cumplir con los requerimientos necesarios para poder ejecutar la aplicación en esta plataforma y verificar que la propiedad Generate Android esté en el valor True que es su valor por defecto y Main Platform con el valor Android.

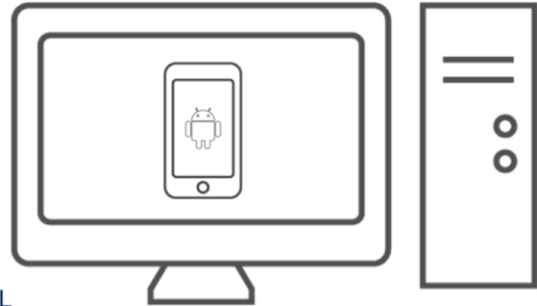
Como ya hemos visto, al ejecutar, GeneXus abre un emulador de Android, para poder probar la aplicación. Los servicios usando el emulador podrán estar alojados en cualquiera de las opciones que vimos, pero debemos verificar que el emulador tenga acceso a ese servidor.

La URL de los servicios se establece mediante la propiedad Services URL del nodo Front end.

En nuestro caso, esta propiedad esta inhabilitada porque la versión Trial no permite ejecutar en otro servidor que no sea el de la nube de GeneXus, pero si usamos la versión GeneXus Full en esta propiedad podremos ingresar la URL correspondiente.

Prototyping in Android on Emulator (cont.)

Generator: Frontend (Front end)	
Name	Frontend
Main Platform	Android
Dynamic Services URL	False
Services URL	https://trialapps3.genex...
SSL Pinning Pin Set	
Smart Devices Cache Management	On
Visual Testing Repository URL	
Generators	
Generate Android	True
Generate Huawei	False
Generate Apple	True
Generate Angular	False
Android Specific	
Android Execution Type	Emulator or Device
Android SDK directory	C:\Android-SDK
JDK Directory	C:\Program Files\Java\j...



Services URL

- Local Network
 - Wifi or Lan assigned IP Address
 - For Localhost use IP 10.0.2.2
- Internet Server (any)

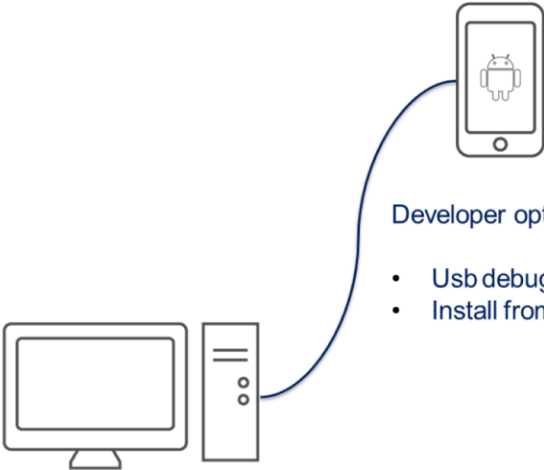
En caso de usar el nombre del host como parte de la URL, el servidor debe ser accesible desde el emulador (es decir debe ser un servidor público) ya que los emuladores no pueden resolver ni Localhost ni ningún nombre de la red interna, por lo que en esos casos debemos usar la IP. Por ejemplo si se ejecuta con el servidor en la PC de desarrollo, podemos usar la IP 10.0.2.2 que representa a Localhost, que sí será accesible desde el emulador.

Un caso especial es cuando necesitamos que nuestra aplicación deba conectarse a distintos servidores. En este caso en la propiedad Services URL debemos indicar un valor por default o dejarla vacía y especificar la propiedad Dynamic Services URL en True, y al ejecutar por primera vez la aplicación, se abrirá un objeto panel que construiremos para que nos solicite ese valor.

GeneXus by Globant

Prototyping in Android on Physical Device

Generator: Frontend (Front end)	
Name	Frontend
Main Platform	Android
Dynamic Services URL	False
Services URL	https://trialapps3.genex...
SSL Pinning Pin Set	
Smart Devices Cache Management	On
Visual Testing Repository URL	
Generators	
Generate Android	True
Generate Huawei	False
Generate Apple	True
Generate Angular	False
Android Specific	
Android Execution Type	Emulator or Device
Android SDK directory	C:\Android-SDK
JDK Directory	C:\Program Files\Java\j...



Developer options:

- Usb debugging ON
- Install from any source allowed

• Device drivers installed

Para ejecutar en un dispositivo físico con Android (y de aquí en adelante incluimos a Huawei, como caso particular de Android), debemos conectar al dispositivo con nuestro computador de desarrollo mediante un cable USB.

Cuando GeneXus detecta que hay un dispositivo con Android conectado al equipo de desarrollo mediante USB, no se abre el emulador y la aplicación se instala y ejecuta en el dispositivo.

Para que el equipo de desarrollo detecte correctamente que el dispositivo está conectado, deberán instalarse en el equipo los drivers adecuados, deberá habilitarse en el dispositivo la opción de Depuración de USB y permitir que se instalen aplicaciones desde cualquier fuente.

Como ya dijimos, para que la aplicación funcione correctamente, el dispositivo debe tener acceso a los servicios. Al tratarse de un dispositivo físico, lo mas común es que mediante Wifi conectemos el dispositivo a la misma red donde corren los servicios. En el caso de que estos servicios corran en un servidor accesible desde internet, además de Wifi podremos usar conexión de datos celulares (como 3G, LTE, etc.).

GeneXus by Globant

Prototyping in Android on Physical Device (cont.)

Generator: Frontend (Front end)

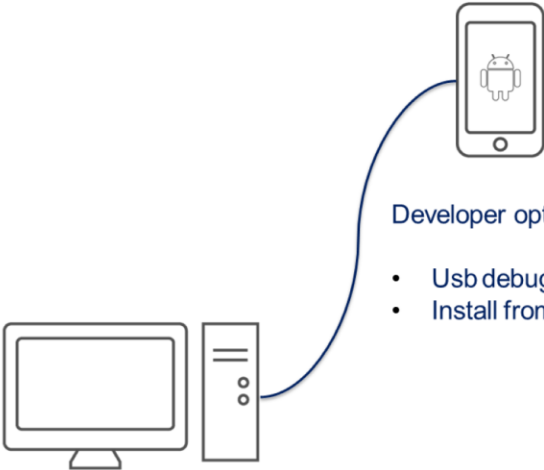
Name	Frontend
Main Platform	Android
Dynamic Services URL	False
Services URL	https://trialapps3.genex...
SSL Pinning Pin Set	
Smart Devices Cache Management	On
Visual Testing Repository URL	

Generators

Generate Android	True
Generate Huawei	False
Generate Apple	True
Generate Angular	False

Android Specific

Android Execution Type	Emulator or Device
Android SDK directory	C:\Android-SDK
JDK Directory	C:\Program Files\Java\j...



Developer options:

- Usb debugging ON
- Install from any source allowed

• **Device drivers installed**

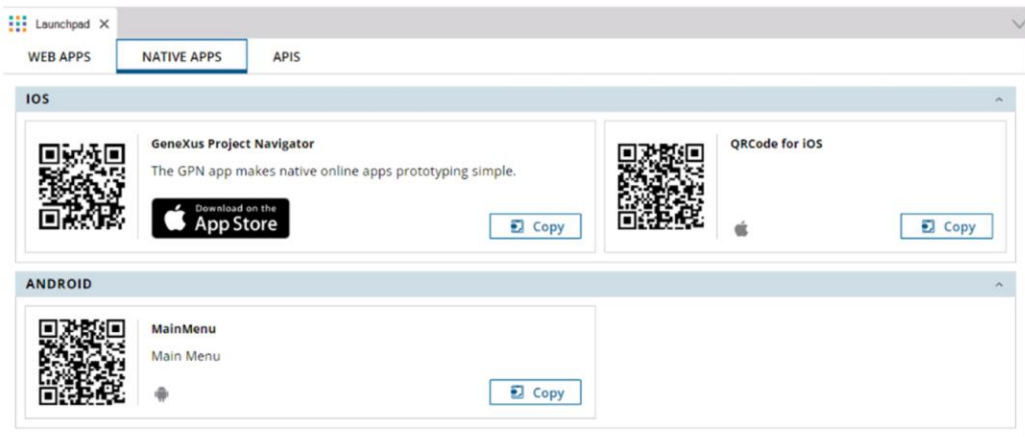
Si el servidor está en una red local, el servidor debe poder ser accedido por el dispositivo mediante Wifi con los permisos necesarios.

Si el servidor es localhost, es decir el servidor web está instalado en el equipo de desarrollo al cual está conectado el dispositivo por USB, esta conexión con cable no establece una conexión de red para que el dispositivo acceda al servidor localhost. La forma de que esta conexión se establezca es si el celular y el computador de desarrollo están en la misma red. Para que esto funcione el computador debe permitir conexiones remotas y además verificar que el firewall no bloquee la conexión.

En caso de que el servidor sea privado y esté remoto, se podría acceder a la red donde está el servidor mediante una VPN y configurar el equipo de desarrollo local como proxy, pero en definitiva el dispositivo debe estar dentro de la misma red que el host.

Resumiendo, si bien es posible prototipar local y que el dispositivo físico acceda al servidor web, lo más cómodo es prototipar en la nube de GeneXus ya que no importa donde esté el dispositivo, mientras tenga conexión a Internet va a poder llegar a los servicios REST del backend de la aplicación.

Prototyping in Android on Physical Device

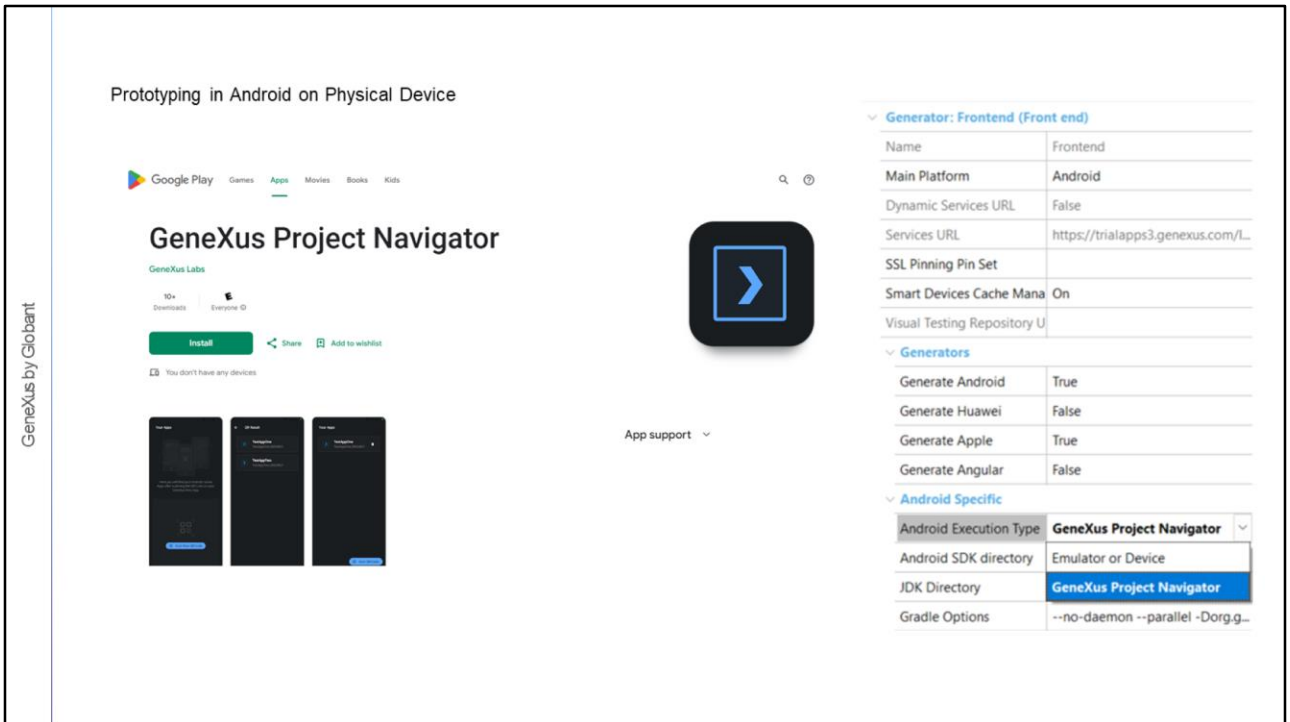


En lugar de tener al dispositivo conectado por cable, otra opción para ejecutar en un dispositivo físico es usar los códigos QR.

Para ello debemos escanear con el dispositivo el código QR de la aplicación compilada y proceder a instalar la misma. Es necesario que el dispositivo tenga acceso a esta URL que es la misma URL donde corren los servicios.

Para acceder a los códigos QR debemos visualizar la ventana del Launchpad y abrir la solapa Native Apps. El Launchpad aparece si ejecutamos el Developer Menu o si vamos por View/Other Tool Windows/Launchpad. Allí vamos a encontrar una sección IOS y otra sección ANDROID. En la de Android tendremos el QRCode para poder escanearlo.

Como en el caso anterior debemos tener la opción de instalación desde otras fuentes habilitada en el dispositivo.



Una tercera opción para prototipar en Android es utilizar la aplicación GeneXus Project Navigator, disponible en el Google Play Store e instalarla en el dispositivo.

Para ejecutar nuestra aplicación en el GPN, primero agregamos el proyecto escaneando el QRCode o ingresando a mano la URL de la propiedad ServicesURL.

La ejecución de la aplicación será interpretada, porque cuando damos F5 GeneXus no compila la aplicación (es decir no genera el apk) pero actualiza la metadata del proyecto en el servidor, la cual es refrescada automáticamente por el GPN, mostrando en el dispositivo los cambios que realizamos. Esto acorta los tiempos de desarrollo porque reduce el tiempo entre que hacemos los cambios y los vemos en ejecución.

Sin embargo prototipando con el GPN tenemos algunas limitaciones, que es cuando se requiere que la aplicación esté compilada, por ejemplo cuando queremos desarrollar aplicaciones offline, cuando queremos usar user controls o external objects, usar push notifications o acceder a ciertos recursos como fonts embebidos, imágenes con varias resoluciones y animaciones, entre otros.



Como mencionamos antes, con GeneXus podemos prototipar en un emulador y también en un dispositivo físico. Si bien prototipar en el emulador tiene la ventaja de que nos evita tener que contar con un dispositivo, tenemos también varias desventajas.

Para empezar con el emulador no tenemos acceso a recursos como el calendario, la cámara, el GPS o aplicaciones de terceros, que sí los tenemos en un dispositivo físico.

También perdemos performance, ya que los emuladores son usualmente más lentos que los dispositivos a la hora de cargar y ejecutar una aplicación.

Otra desventaja muy importante es que lo que se ve en el emulador no es exactamente lo que se ve en un dispositivo físico. Quizás las diferencias sean pequeñas, pero es muy importante probar la aplicación final en un dispositivo real.

Por último cabe mencionar que ejecutar la aplicación en el emulador no nos proporciona una "sensación" adecuada, ya que en realidad no se puede tener la experiencia de tocar la pantalla, deslizar el dedo, hacer gestos, etc.

Sin embargo aunque parezca que no hay razón para utilizar un emulador, es la forma más rápida de tener una aplicación funcionando sin contar con nada más que tu computador.

Prototyping in Apple

Veamos ahora las distintas maneras de prototipar una aplicación generada para iOS.

Prototyping in iOS

GeneXus by Globant

▼ Generator: Frontend (Front end)	
Name	Frontend
Main Platform	Apple
Dynamic Services URL	False
Services URL	https://trialapps3.g
SSL Pinning Pin Set	
Smart Devices Cache Mana	On
Visual Testing Repository U	
▼ Generators	
Generate Android	False
Generate Apple	True
Generate Angular	False
▼ Apple Specific	

▼ Apple Specific	
Required Xcode version	14.0
Flexible client version	1.2.1
iOS Deployment Target	12.0
tvOS Deployment Target	12.0
watchOS Deployment Ta	5.0
Apple Execution Type	GeneXus Project Navigator (...)
Execution Device	Simulator (Mac)
Mac Host	iOS Device (Mac)
Mac User	iTunes Sync (Local)
Mac Password	Build IPA (Local)
	GeneXus Project Navigator (Device)

Vayamos a revisar las propiedades del nodo Front end, ya que precisamos definir que la plataforma de generación sea Apple.

Por defecto la propiedad Generate Apple está en True, pero también la propiedad Generate Android está en True y la propiedad Main Platform por defecto es Android.

Lo primero que debemos hacer es configurar la propiedad Main Platform en el valor Apple. Si queremos podemos seguir generando Android, pero si solamente vamos a desarrollar una app para IOS entonces es preferible poner la propiedad Generate Android en False, con lo cual ahorraremos tiempo de generación.

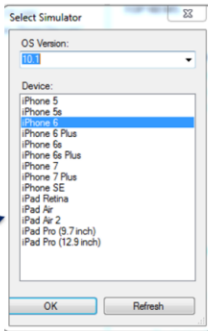
Luego debemos configurar en el grupo Apple Specific, algunas opciones para poder ejecutar la aplicación, que veremos a continuación.

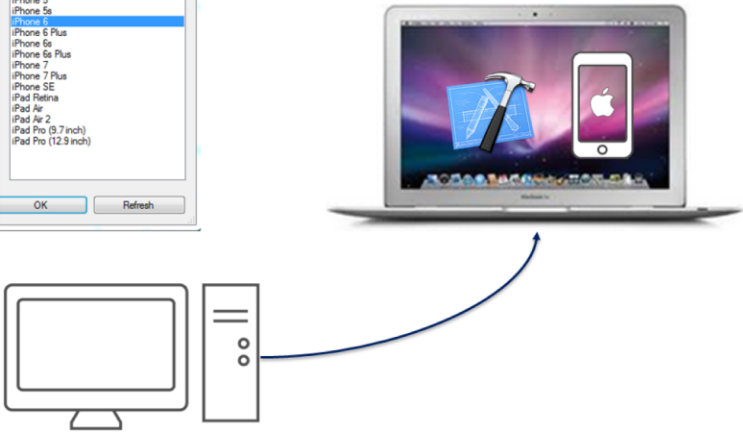
GeneXus by Globant

Prototyping in Apple: Simulator (Mac)

Apple Specific	
Required Xcode version	14.0
Flexible client version	1.2.1
iOS Deployment Target	12.0
tvOS Deployment Target	12.0
watchOS Deployment Target	5.0
Apple Execution Type	Simulator (Mac)
iOS Simulator	Last used simulator
Mac Host	192.168.30.129
Mac User	gxTraining
Mac Password	*****

- MAC with remote login enabled
- Secure Shell enabled





La propiedad Apple Execution Type, nos da varias opciones de Prototipación.

La primera opción del combo box nos muestra el valor Simulator (Mac), que nos permite que se pueda prototipar en un emulador de un dispositivo Apple ejecutando en una Mac.

Obviamente para esto tenemos que contar con un computador de ese tipo, pero no tiene porqué ser el equipo de desarrollo que esté ejecutando GeneXus, sino que puede ser un computador remoto.

Para configurar el acceso, en las propiedades del grupo IOS Specific, debemos dar valores a Mac Host, Mac User y Mac Password.

Para esto la Mac deberá estar en la misma red del equipo que tiene instalado GeneXus. Al ejecutar desde GeneXus, el código fuente se transfiere a la Mac, allí se compila, se abre el emulador y se instala la aplicación compilada en el mismo.

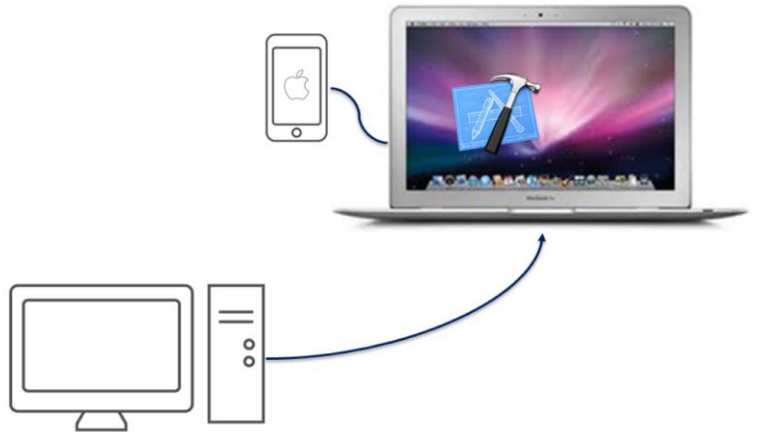
La propiedad iOS Simulator nos permite elegir los emuladores disponibles en la Mac.

La Mac debe permitir login remoto y tener habilitado SSH (Secure Shell).

Respecto a los servicios Rest, debemos contar con acceso a estos servicios desde la Mac, es decir que el servidor web donde están los servicios, debe estar en la misma red que el computador MAC.

Prototyping in Apple: iOS Device (Mac)

Apple Specific	
Required Xcode version	14.0
Flexible client version	1.2.1
iOS Deployment Target	12.0
tvOS Deployment Target	12.0
watchOS Deployment Target	5.0
Apple Execution Type	iOS Device (Mac)
Mac Host	192.168.30.129
Mac User	gxTraining
Mac Password	*****



- IOS developer
- MAC Certificate
- AppID for each app
- Assign Development Team Id property in main object

La segunda opción del combo del tipo de ejecución tiene el valor iOS Devices (MAC) nos permite usar un dispositivo físico Apple.

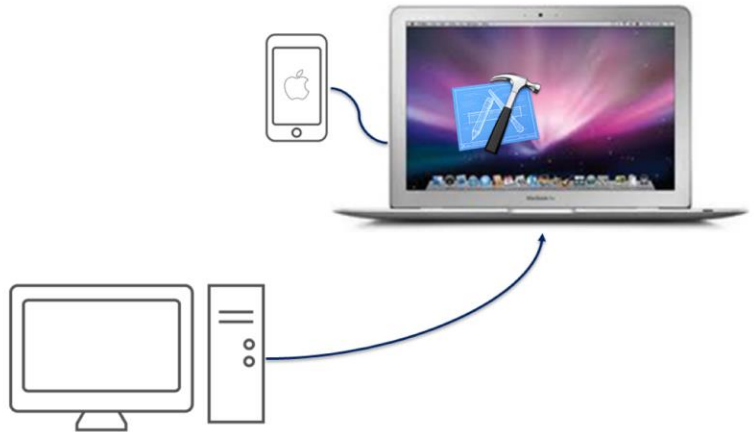
En este caso, necesitamos de un computador MAC y de un dispositivo Apple como iPhone, iPad, Apple TV o Apple Watch, conectado a la MAC por cable USB.

Utilizando esta opción, debemos completar las propiedades Mac Host ,Mac User y Mac Password para acceder al computador MAC, que debe estar en la misma red que el equipo de desarrollo.

Al ejecutar desde GeneXus se genera la aplicación y se transfieren los fuentes a la MAC. Al finalizar la transferencia, la aplicación es compilada en la MAC y el paquete de extensión .ipa se transfiere e instala en el dispositivo Apple conectado a la MAC.

Prototyping in Apple: IOS Device (Mac) (cont.)

Apple Specific	
Required Xcode version	14.0
Flexible client version	1.2.1
iOS Deployment Target	12.0
tvOS Deployment Target	12.0
watchOS Deployment Target	5.0
Apple Execution Type	iOS Device (Mac)
Mac Host	192.168.30.129
Mac User	gxTraining
Mac Password	*****



- IOS developer
- MAC Certificate
- AppID for each app
- Assign Development Team Id property in main object

Esta opción de Prototipación provee la forma más veloz y realista de prototipar, ya que la aplicación será idéntica a la que se podrá descargar luego del Apple Store.

No obstante, para poder compilar la aplicación, es necesario que el desarrollador esté registrado como Desarrollador iOS en el sitio de Apple, que la MAC tenga instalado un certificado de desarrollo y que el dispositivo donde se vaya a probar, se haya registrado dentro del certificado.

También es necesario crear una App ID por cada aplicación que queramos probar y crear un perfil de aprovisionamiento donde se vinculará toda esta información.

Además es necesario obtener el valor del campo Team-ID de la cuenta de desarrollador Apple y asignar ese valor en la propiedad Development Team Id del objeto main de la aplicación, para que luego sea incluido en proyecto Xcode generado.

GeneXus by Globant

GeneXus™
by Globant

Recent pages Statistics

Q [Sign in](#) [Login](#)

Native Mobile Applications Development
Table of contents

Page Tools ▾ Page Info ▾ Also seen in ▾ Other document versions ▾

Prototyping in iOS with a compiled application

GeneXus™
by Globant

Recent pages Statistics

Q [Sign in](#) [Login](#)

Native Mobile Applications Development
Table of contents

Page Tools ▾ Page Info ▾ Also seen in ▾

HowTo: Prototye an iOS Application on a Mac

Introduction

Native Mobile Generator

Welcome to Native Mobile Applications Development

My first Android application

My first iOS application

My first Offline Native Mobile application

Requirements and Installation

Architecture

Tabs offered in Panel and Work With objects

Layouts

Orders and Filters in Grids of Panels

Transaction Rules in Native Mobile

This documentation is valid for: [GeneXus 18 Help](#) [GeneXus 17 Help](#) [GeneXus 16 Help](#) [GeneXus 15 Help](#) [GeneXus X Evolution 3 Help](#) [GeneXus X Evolution 2 Help](#)

The purpose of this article is to explain the necessary steps to prototype your iPhone or iPad application on your Mac.

Step 1: Install requirements
Check Mac requirements on [Apple Requirements](#)

Step 2: Enable ssh
Go to System preferences.



Para más detalle sobre cómo seguir estos pasos te sugerimos visitar el wiki.

17

Prototyping in Apple: iTunes Sync (Local)

Apple Specific	
Required Xcode version	14.0
Flexible client version	1.2.1
iOS Deployment Target	12.0
tvOS Deployment Target	12.0
watchOS Deployment Target	5.0
Apple Execution Type	iTunes Sync (Local)
Mac Host	192.168.30.129
Mac User	gxTraining
Mac Password	*****

Download the latest version from the Microsoft Store.

Download the latest version from the Microsoft Store.

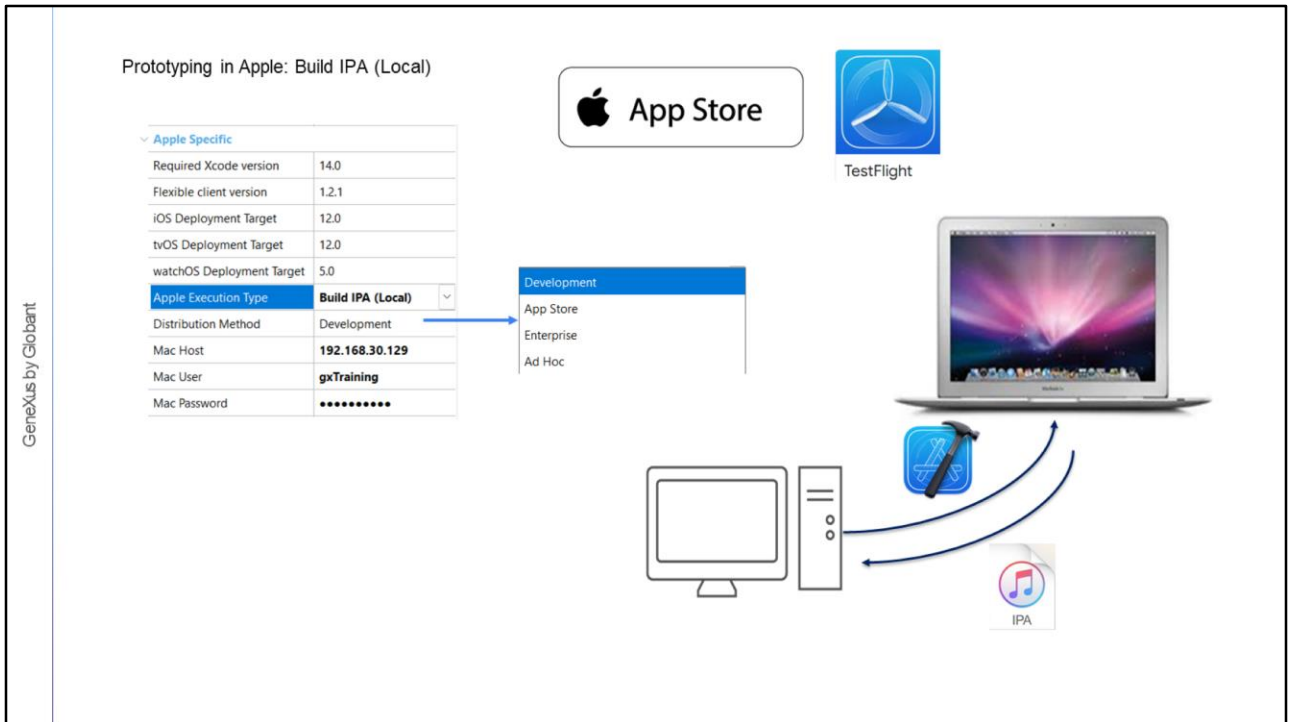
The latest entertainment apps now come installed with the latest macOS. Upgrade today to get your favorite music, movies, TV shows, and podcasts. You can join Apple Music and stream — or download and play offline — millions of songs, ad-free.



Eligiendo la opción iTunes Sync (Local), se transfiere el proyecto XCode a una Mac y allí se compila la aplicación generando un archivo *.ipa.

Luego el proceso copia el archivo .ipa de la Mac al equipo de desarrollo y lo deja en la carpeta especificada en la ventana de Output del IDE de GeneXus.

Por último en la máquina de desarrollo se abre automáticamente iTunes para mostrar la aplicación en el catálogo de aplicaciones.



Con la opción Build IPA (local) se transfiere el proyecto XCode a una Mac y se compila la aplicación usando el método especificado en la propiedad Distribution Method generando un archivo *.ipa. A continuación, el archivo se copia a la carpeta asociada al environment de la KB.

Si en la propiedad Distribution Method, se elije el valor Development, la aplicación es compilada y firmada con el perfil de aprovisionamiento de desarrollo para ser enviada al equipo de Test.

Si se elige App Store, la aplicación es compilada y firmada con el perfil de aprovisionamiento de desarrollo para ser publicada en TestFlight y en Apple Store.

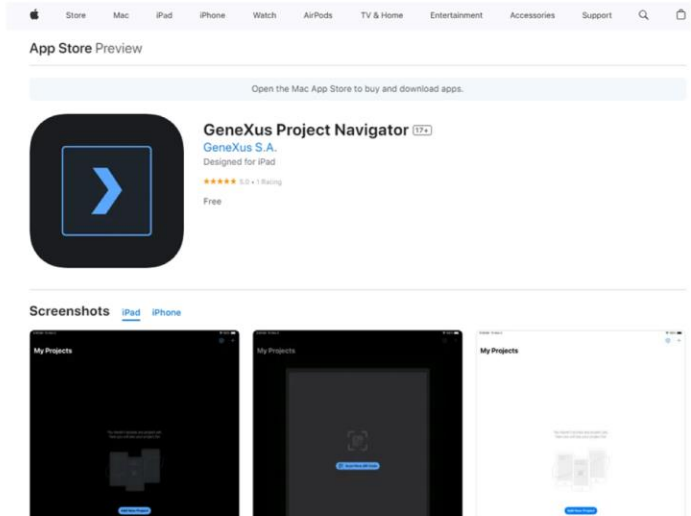
TestFlight es una aplicación de Apple que le permite a los desarrolladores de aplicaciones poder invitar a evaluadores internos o externos para que prueben sus aplicaciones y, de esta manera, recopilar comentarios y datos relacionados con la funcionalidad de su proyecto.

Si seleccionamos Enterprise, la aplicación es compilada y firmada para distribución interna.

Y si elegimos Ad Hoc, la aplicación es compilada y firmada con un perfil de aprovisionamiento creado para un caso concreto, para distribuirse luego a los testers de la aplicación.

Prototyping in Apple on GPN

▼ Apple Specific	
Required Xcode version	14.0
Flexible client version	1.2.1
iOS Deployment Target	12.0
tvOS Deployment Target	12.0
watchOS Deployment Target	5.0
Apple Execution Type	GeneXus Project Navigator (Device) ▼
Execution Device	
Mac Host	192.168.30.129
Mac User	gxTraining
Mac Password	*****

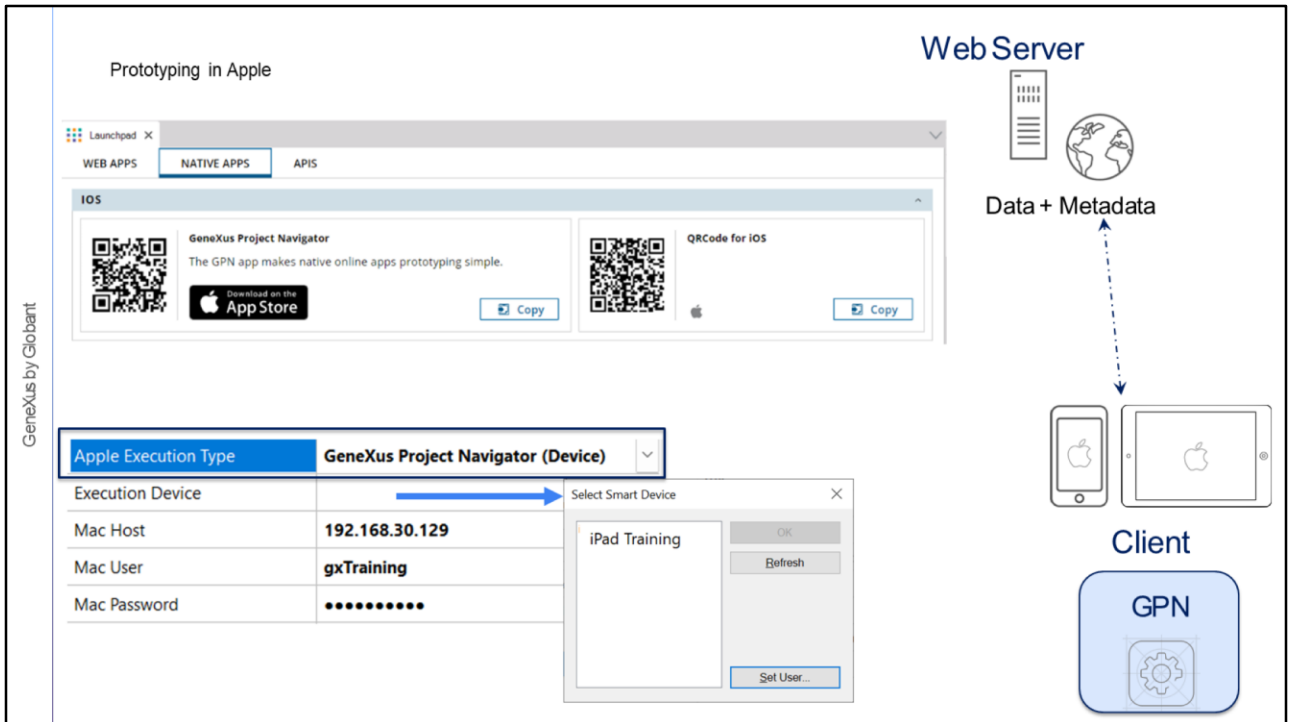


La última opción del combo y que es la opción por defecto, dice GeneXus Project Navigator (Device).

Esto nos permite prototipar en un dispositivo físico utilizando el GeneXus Project Navigator, que se deberá descargar del Apple Store.

En este caso no necesitamos contar con un computador MAC para compilar la aplicación y su comportamiento es análogo a que lo que ya vimos para Android.

Las limitaciones del GPN para prototipar en la plataforma Apple, son las mismas que ya mencionamos.



El link para descargar el GPN desde el Apple Store está disponible en el Launchpad. Allí también aparece el QRCode de la URL de la aplicación, que utilizaremos desde el GPN instalado en el dispositivo, para registrar la aplicación.

Cuando usamos la opción GeneXus Project Navigator (Device), debemos asignar el dispositivo donde vamos a prototipar. En la propiedad Execution Device podremos elegir alguno de los dispositivos asociados a nuestra cuenta de GeneXus Account. Para eso debemos primero abrir el cuadro de diálogo de la propiedad Execution Device, presionar el botón Set User, loguearnos con nuestra cuenta GeneXus y luego elegir el dispositivo de la lista.

Una vez que catalogamos la aplicación en el GPN (ingresando la URL manualmente o a través del QRCode), cada vez que ejecutamos desde GeneXus, la aplicación se actualiza automáticamente con los cambios realizados en GeneXus, gracias a una Push Notification recibida desde el server.

Esta forma de prototipar tiene la ventaja de que vemos la aplicación en un dispositivo Apple real, sin necesidad de tener una MAC, ni de registrarnos como desarrolladores en el sitio de Apple.



En este video vimos la flexibilidad que disponemos para prototipar aplicaciones nativas, tanto en Android como en Apple.

En el resto del curso utilizaremos el emulador Android para prototipar nuestra aplicación.