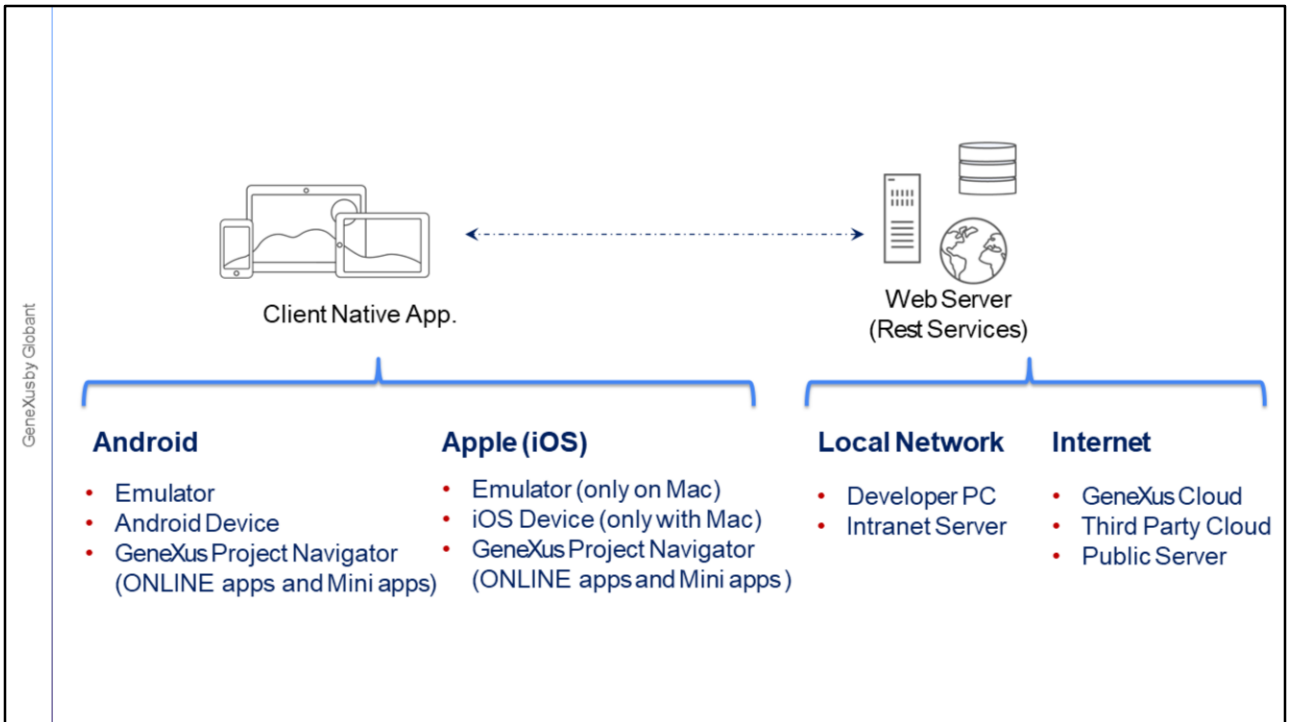


# Prototyping a mobile native app



Rodolfo Robalo

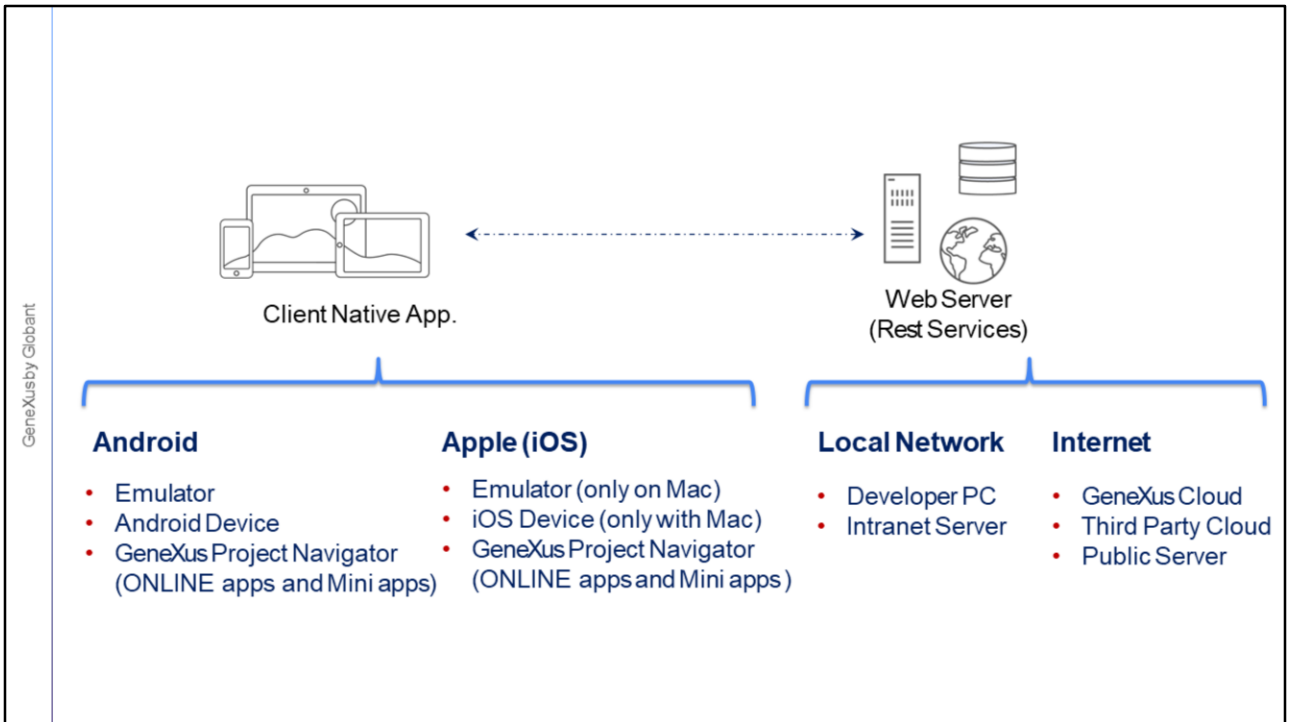
Neste vídeo veremos como podemos testar nossa aplicação Android ou Apple em um emulador, em um dispositivo físico ou utilizando a aplicação GeneXus Project Navigator e conhecer as vantagens e desvantagens de cada uma dessas possibilidades.



De acordo com o que vimos quando estudamos a arquitetura de uma aplicação nativa, para executar uma aplicação precisamos de dois componentes, a aplicação mobile nativa propriamente dita e os serviços REST que atendem as requisições da aplicação, hospedados em um servidor web.

No caso do Android, a aplicação cliente pode ser executada em um emulador como utilizamos no curso, ou podemos executá-la em um dispositivo físico Android padrão, ou um dispositivo Android Huawei, conectados ao nosso computador Windows através de uma porta USB. Em ambos os casos, a aplicação é compilada antes de ser executada e o arquivo .apk é instalado no emulador ou no dispositivo físico através da conexão USB.

Mas também podemos executar a aplicação sem precisar compilá-la, utilizando o GeneXus Project Navigator. O GPN é uma aplicação nativa feita por GeneXus que podemos baixar do Google Play, instalar em um dispositivo Android padrão e prototipar nossa aplicação de forma interpretada sem necessidade de o dispositivo estar conectado fisicamente ao nosso computador de desenvolvimento. Falaremos um pouco mais sobre GPN em alguns instantes.

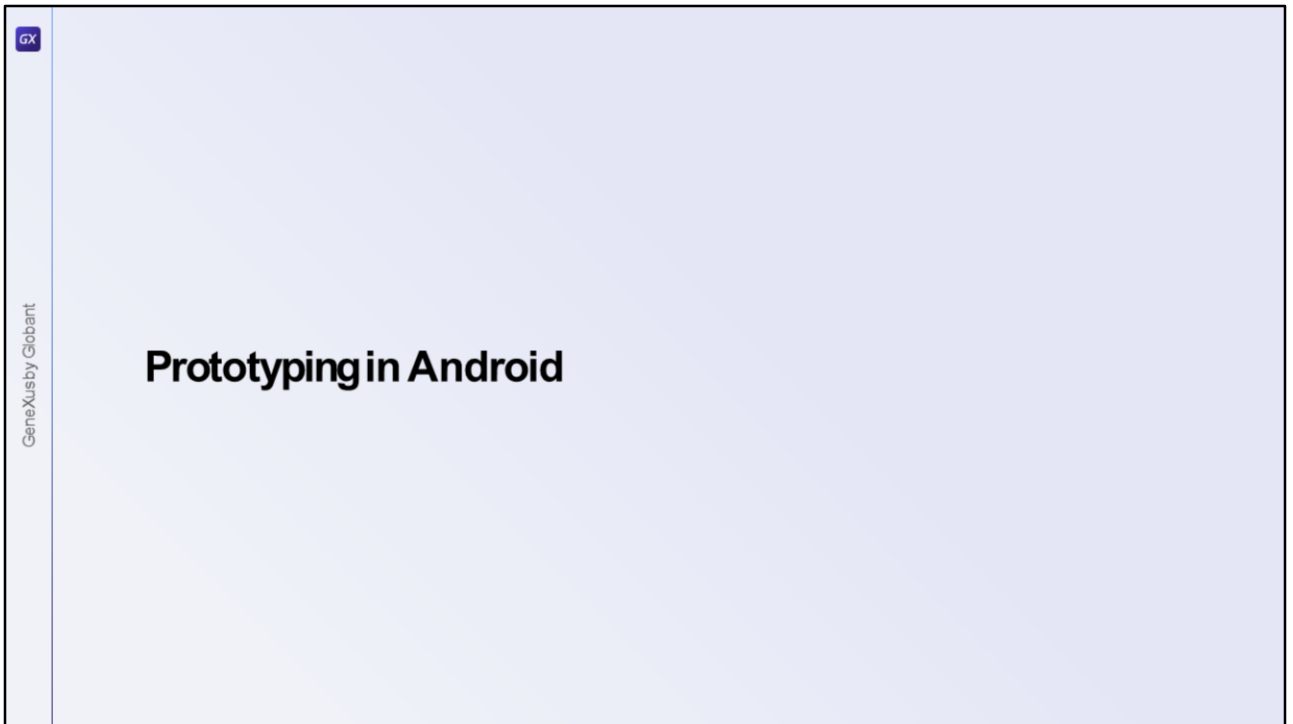


Para Apple, a aplicação também pode ser executada em um emulador ou em um dispositivo físico, mas precisamos de um computador Mac para compilar e para executar o emulador ou para conectar um dispositivo físico via USB. Além disso, neste caso também temos a possibilidade de executar a aplicação de forma interpretada em um dispositivo com iOS usando o GPN, (desde que a aplicação seja Online ou Mini App) e neste caso não necessitamos de um Mac, apenas ter um iPhone ou iPad com o GPN instalado a partir da Apple Store.

Quanto aos serviços, estes podem executar na rede local, seja no computador do desenvolvedor ou em algum outro servidor web disponível na rede; ou podem ser executados em um servidor na internet. Neste caso, o servidor pode ser o da nuvem de GeneXus como estamos utilizando no curso, sendo esta a única possibilidade quando usamos a versão trial ou se estamos usando GeneXus Full, os serviços REST podem ser hospedados em uma nuvem de terceiros como a fornecida pela Amazon ou Google, ou podemos publicar os serviços em um servidor próprio que tenha acesso público.

O importante é que a aplicação mobile cliente tenha acesso a estes serviços, ou seja, que possa conectar-se via Wi-Fi ao servidor onde os serviços estão instalados.

A seguir veremos cada uma das alternativas e as questões que devemos considerar em cada caso.



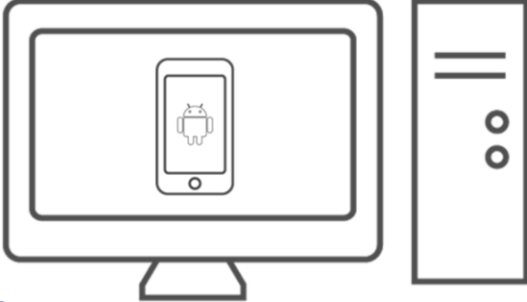
Até agora executamos a aplicação nativa no Android utilizando o emulador.

Vejamos um pouco mais em detalhe os mecanismos de prototipagem nesta plataforma.

GeneXusby 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
  - For Localhost use IP 10.0.2.2
- Internet Server (any)

Para gerar em Android, como vimos em vídeos anteriores, basta atender aos requisitos necessários para poder executar a aplicação nesta plataforma e verificar se a propriedade Generate Android está com o valor True, que é seu valor padrão, e Main Platform com o valor Android.

Como já vimos, ao executar, GeneXus abre um emulador de Android, para poder testar a aplicação. Os serviços usando o emulador poderão estar hospedados em qualquer uma das opções que vimos, mas devemos verificar se o emulador tem acesso a esse servidor.

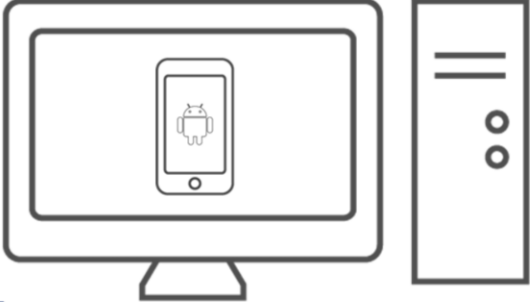
A URL dos serviços é definida usando a propriedade Services URL do nó Front end.

No nosso caso, esta propriedade está desabilitada porque a versão Trial não permite a execução em outro servidor que não seja o da nuvem de GeneXus, mas se utilizarmos a versão GeneXus Full nesta propriedade podemos inserir a URL correspondente.

GeneXusby 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
  - For Localhost use IP 10.0.2.2
- Internet Server (any)

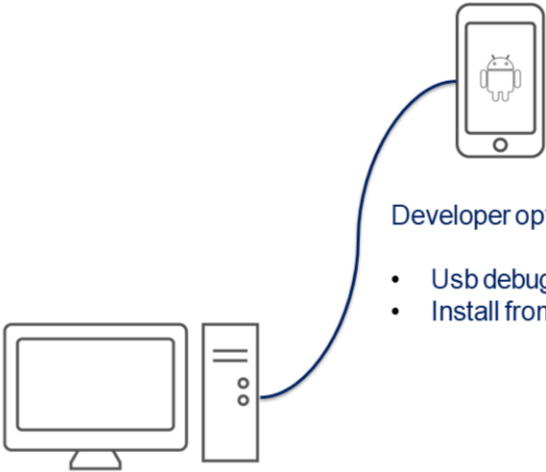
No caso de usar o nome do host como parte da URL, o servidor deverá ser acessível a partir do emulador (ou seja, deve ser um servidor público), pois os emuladores não podem resolver nem Localhost nem qualquer nome da rede interna, de modo que nesses casos devemos usar o IP. Por exemplo, se for executado com o servidor no PC de desenvolvimento, podemos usar o IP 10.0.2.2 que representa Localhost, que será acessível a partir do emulador.

Um caso especial é quando necessitamos que nossa aplicação deva se conectar a diferentes servidores. Neste caso, na propriedade Services URL devemos indicar um valor default ou deixá-la vazia e especificar a propriedade Dynamic Services URL como True, e ao executar pela primeira vez a aplicação, será aberto um objeto panel que construiremos para que nos solicite esse valor.

GeneXusby 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...



- Device drivers installed

Developer options:

- Usb debugging ON
- Install from any source allowed

Para executar em um dispositivo físico com Android (e daqui em diante incluímos Huawei, como caso particular de Android), devemos conectar o dispositivo ao nosso computador de desenvolvimento através de um cabo USB.

Quando GeneXus detecta que existe um dispositivo com Android conectado ao equipamento de desenvolvimento via USB, não é aberto o emulador e a aplicação é instalada e executada no dispositivo.

Para que o equipamento de desenvolvimento detecte corretamente que o dispositivo está conectado, deverão ser instalados no equipamento os drivers adequados, deverá ser habilitada no dispositivo a opção de Depuração de USB e permitir que sejam instaladas aplicações a partir de qualquer fonte.

Como já dissemos, para que a aplicação funcione corretamente, o dispositivo deve ter acesso aos serviços. Por se tratar de um dispositivo físico, o mais comum é que através de Wifi conectemos o dispositivo à mesma rede onde executam os serviços. No caso em que estes serviços executem em um servidor acessível a partir da internet, além de Wifi poderemos usar conexão de dados celulares (como 3G, LTE, etc.).

GeneXusby 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

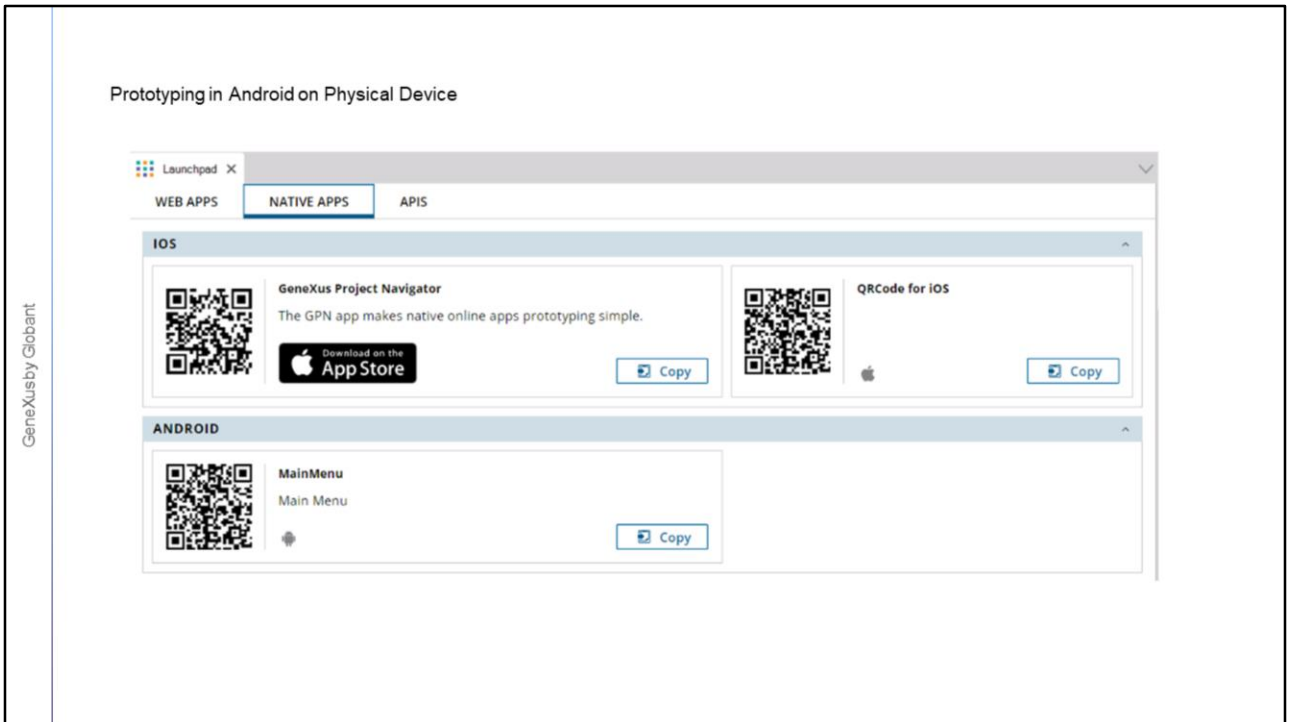
Se o servidor estiver em uma rede local, o servidor deverá estar acessível pelo dispositivo via Wifi com as permissões necessárias.

Se o servidor for localhost, ou seja, o servidor web estiver instalado no equipamento de desenvolvimento ao qual está conectado o dispositivo via USB, esta conexão com cabo não estabelece uma conexão de rede para que o dispositivo acesse o servidor localhost. A forma como essa conexão é estabelecida é se o celular e o computador de desenvolvimento estão na mesma rede. Para que isso funcione, o computador deve permitir conexões remotas e também verificar se o firewall não bloqueia a conexão.

No caso em que o servidor seja privado e esteja remoto, poderia ser acessada a rede onde está o servidor por meio de uma VPN e configurar o equipamento de desenvolvimento local como proxy, mas, em suma, o dispositivo deve estar dentro da mesma rede que o host.

Resumindo, embora seja possível prototipar local e que o dispositivo físico acesse o servidor web, o mais conveniente é prototipar na nuvem de GeneXus, pois não importa onde esteja o dispositivo, desde que tenha conexão com a Internet ele poderá chegar aos serviços REST do backend da aplicação.



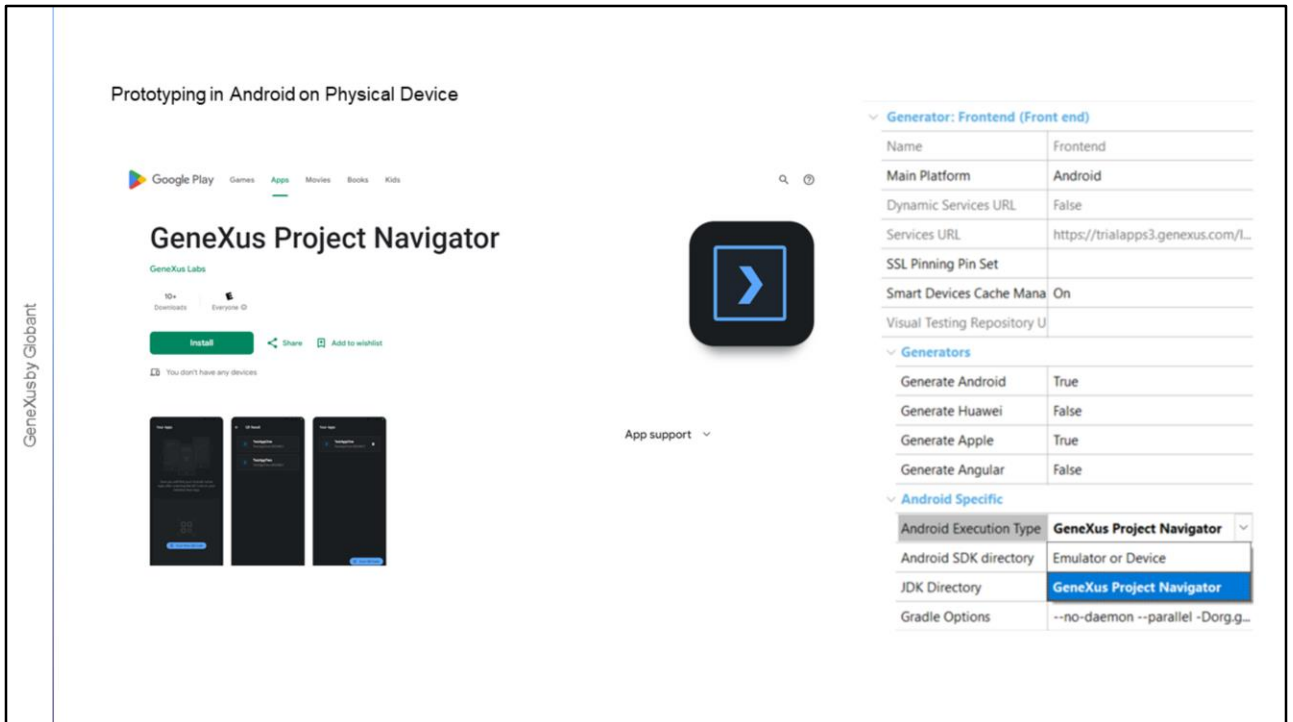


Em vez de ter o dispositivo conectado por cabo, outra opção para executar em um dispositivo físico é usar os códigos QR.

Para isso, devemos escanear com o dispositivo o código QR da aplicação compilada e proceder para instalar. É necessário que o dispositivo tenha acesso a esta URL, que é a mesma URL onde rodam os serviços.

Para acessar os códigos QR devemos visualizar a janela do Launchpad e abrir a aba Native Apps. O Launchpad aparece se executarmos o Developer Menu ou se formos por View/Other Tool Windows/Launchpad. Lá encontraremos uma seção IOS e outra seção ANDROID. Na de Android teremos o QRCode para podermos escaneá-lo.

Como no caso anterior, devemos ter a opção de instalação a partir de outras fontes habilitada no dispositivo.



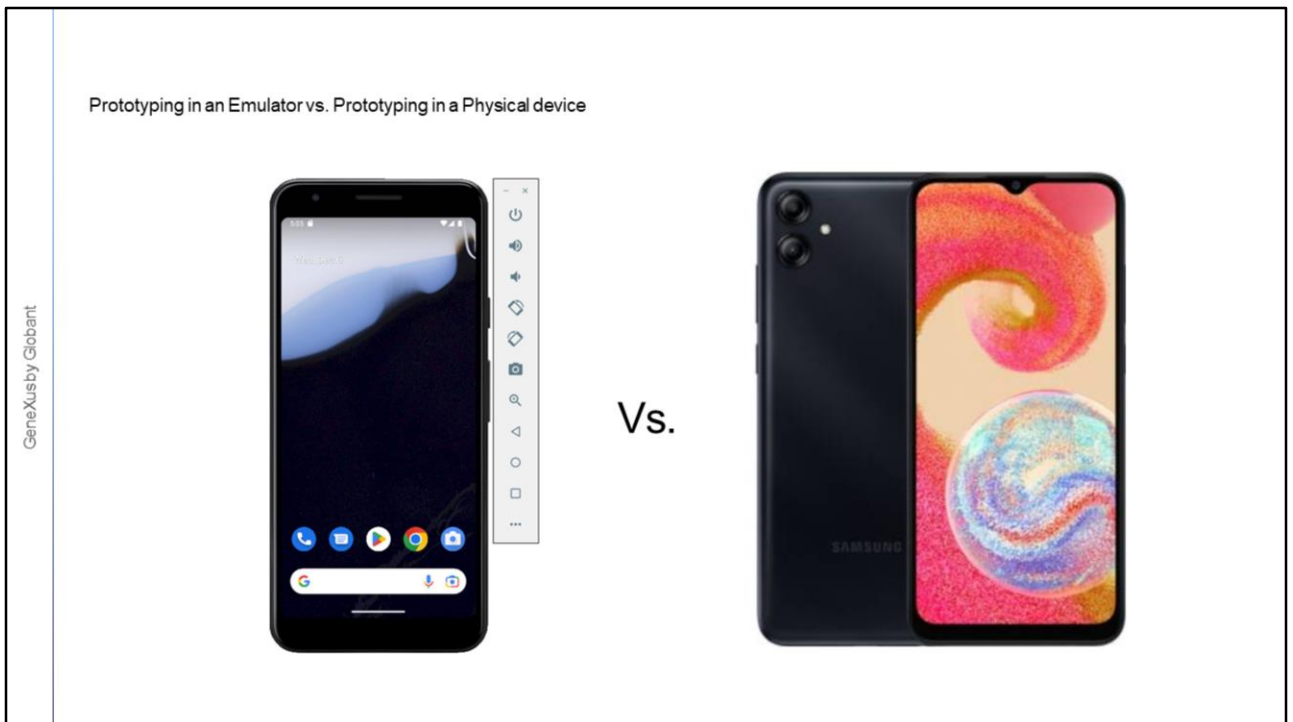
Uma terceira opção para prototipar em Android é utilizar a aplicação GeneXus Project Navigator, disponível no Google Play Store, e instalá-la no dispositivo.

Para executar nossa aplicação no GPN, primeiro adicionamos o projeto escaneando o QRCode ou inserindo manualmente a URL da propriedade Services URL.

A execução da aplicação será interpretada, pois ao pressionarmos F5, GeneXus não compila a aplicação (ou seja, não gera o apk), mas atualiza os metadados do projeto no servidor, que são atualizados automaticamente pelo GPN, mostrando no dispositivo as alterações que realizamos. Isso reduz os tempos de desenvolvimento porque reduz o tempo entre fazer as alterações e vê-las em execução.

Porém, prototipando com o GPN temos algumas limitações, que é quando é necessário que a aplicação esteja compilada, por exemplo quando queremos desenvolver aplicações offline, quando queremos usar user controls ou external objects, usar push notifications ou acessar determinados recursos como fontes incorporadas, imagens com várias resoluções e animações, entre outras.

O GeneXus Project Navigator não é utilizado para Huawei, pois para que a aplicação acesse os serviços próprios da plataforma, como Huawei Analytics, Maps, Notifications, é necessária uma chave e esta só é obtida com a aplicação compilada. A boa notícia é que podemos prototipar Mini Apps.



Como mencionamos anteriormente, com GeneXus podemos prototipar em um emulador e também em um dispositivo físico. Embora prototipar no emulador tenha a vantagem de evitar a necessidade de ter um dispositivo, temos também várias desvantagens.

Para começar, com o emulador não temos acesso a recursos como o calendário, a câmera, o GPS ou aplicações de terceiros, que temos em um dispositivo físico.

Também perdemos desempenho, já que os emuladores costumam ser mais lentos que os dispositivos na hora de carregar e executar uma aplicação.

Outra desvantagem muito importante é que o que é visto no emulador não é exatamente o que é visto em um dispositivo físico. Talvez as diferenças sejam pequenas, mas é muito importante testar a aplicação final num dispositivo real.

Por último, vale mencionar que executar a aplicação no emulador não nos proporciona uma “sensação” adequada, já que na realidade não é possível ter a experiência de tocar na tela, deslizar o dedo, fazer gestos, etc.

No entanto, embora possa parecer que não há razão para utilizar um emulador, é a maneira mais rápida de ter uma aplicação funcionando sem contar com nada mais que seu computador.



Vejamos agora as diferentes maneiras de prototipar uma aplicação gerada para IOS.

## Prototyping in iOS

GeneXusby 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	

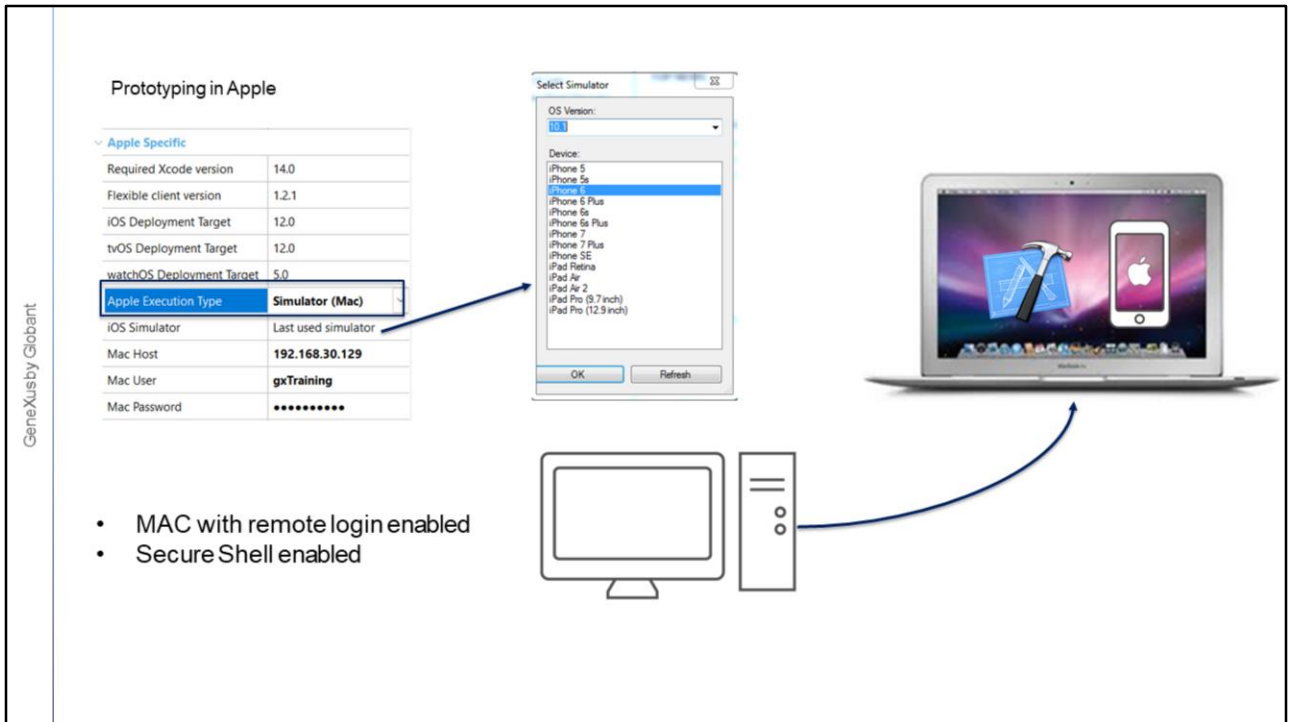
▼ Apple Specific	
Required Xcode version	14.0
Flexible client version	1.2.1
iOS Deployment Target	12.0
tvOS Deployment Target	12.0
watchOS Deployment Ta	5.0
Apple Execution Type	GeneXus Project Navigator (...)
Execution Device	Simulator (Mac)
Mac Host	iOS Device (Mac)
Mac User	iTunes Sync (Local)
Mac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

Vamos revisar as propriedades do nó Front end, pois precisamos definir que a plataforma de geração seja Apple.

Por padrão a propriedade Generate Apple está em True, mas também a propriedade Generate Android está em True e a propriedade Main Platform por padrão é Android.

A primeira coisa que devemos fazer é configurar a propriedade Main Platform com o valor Apple. Se quisermos, podemos continuar gerando Android, mas se vamos desenvolver apenas uma app para IOS então é preferível definir a propriedade Generate Android como False, o que economizará tempo de geração.

Em seguida devemos configurar no grupo Apple Specific, algumas opções para podermos executar a aplicação, que veremos a seguir.



A propriedade Apple Execution Type nos oferece várias opções de Prototipação.

A primeira opção do combo box nos mostra o valor Simulator (Mac), que nos permite prototipar em um emulador de um dispositivo Apple executando em um Mac. Obviamente para isso temos que ter um computador deste tipo, mas não precisa ser o equipamento de desenvolvimento que está executando GeneXus, mas pode ser um computador remoto.

Para configurar o acesso, nas propriedades do grupo IOS Specific, devemos fornecer valores para Mac Host, Mac User e Mac Password.

Para isso, o Mac deverá estar na mesma rede do equipamento que possui GeneXus instalado. Ao executar a partir de GeneXus, o código fonte é transferido para o Mac, lá é compilado, é aberto o emulador e é instalada a aplicação compilada nele.

A propriedade iOS Simulator nos permite escolher os emuladores disponíveis no Mac.

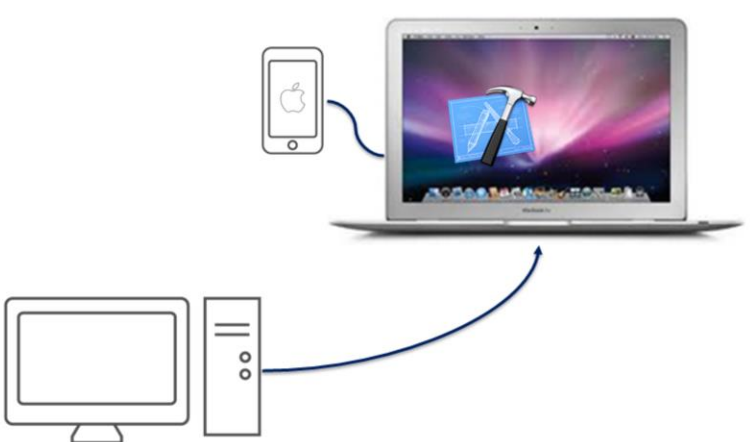
O Mac deve permitir login remoto e ter habilitado SSH (Secure Shell).

Em relação aos serviços Rest, devemos ter acesso a estes serviços a partir do Mac, ou seja, o servidor web onde estão os serviços deve estar na mesma rede que o computador MAC.

GeneXusby Globant

### Prototyping in Apple

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

A segunda opção do combo do tipo de execução tem o valor iOS Device (MAC) e nos permite usar um dispositivo físico Apple.

Neste caso, precisamos de um computador MAC e de um dispositivo Apple como iPhone, iPad, Apple TV ou Apple Watch, conectado ao MAC via cabo USB.

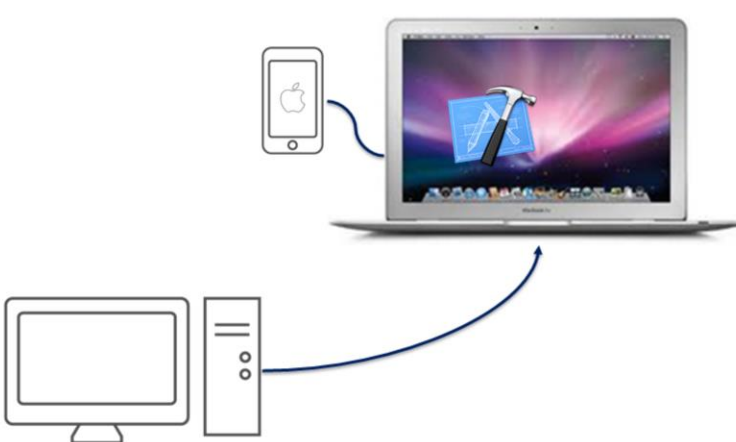
Utilizando esta opção, devemos preencher as propriedades Mac Host, Mac User e Mac Password para acessar o computador MAC, que deve estar na mesma rede que o equipamento de desenvolvimento.

Ao executar a partir de GeneXus, é gerada a aplicação e são transferidos os fontes para o MAC. Ao finalizar a transferência, a aplicação é compilada no MAC e o pacote de extensão .ipa é transferido e instalado no dispositivo Apple conectado ao MAC.

GeneXusby Globant

### Prototyping in Apple

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 opção de Prototipação fornece a maneira mais rápida e realista de prototipar, já que a aplicação será idêntica à que poderá ser baixada posteriormente na Apple Store.

No entanto, para poder compilar a aplicação, é necessário que o desenvolvedor esteja registrado como Desenvolvedor iOS no site da Apple, que o MAC possua instalado um certificado de desenvolvimento e que o dispositivo onde será testado tenha sido registrado dentro do certificado.

Também é necessário criar um App ID para cada aplicação que queremos testar e criar um perfil de provisionamento onde será vinculada toda esta informação.

Além disso, é necessário obter o valor do campo Team-ID da conta de desenvolvedor Apple e atribuir esse valor à propriedade Development Team Id do objeto main da aplicação, para que então seja incluído no projeto Xcode gerado.



GeneXusby Globant

GeneXus™ by Globant

Recent pages Statistics

Page Tools Page Info Also seen in Other document versions

## Prototyping in iOS with a compiled application

GeneXus™ by Globant

Recent pages Statistics

Page Tools Page Info Also seen in

## HowTo: Prototye an iOS Application on a Mac

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

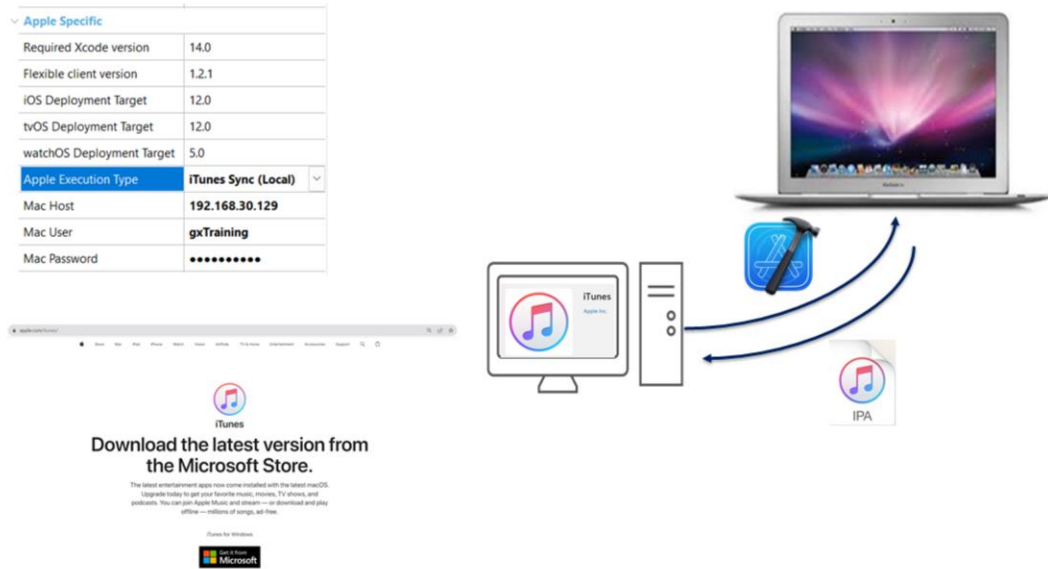
Check Mac requirements on [Apple Requirements](#)

### Step 2: Enable ssh

Go to System preferences.



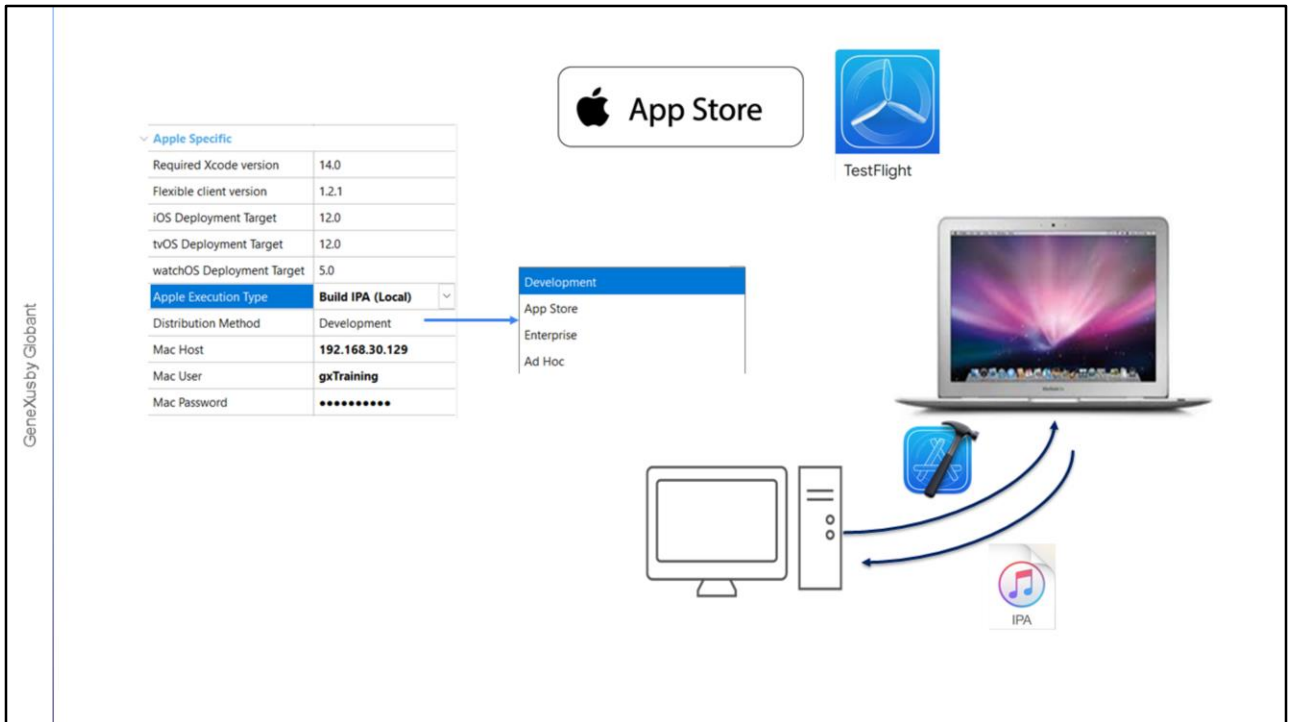
Para mais detalhes sobre como seguir esses passos, sugerimos que você visite a wiki.



Escolhendo a opção iTunes Sync (Local), é transferido o projeto XCode para um Mac e lá é compilada a aplicação, gerando um arquivo de extensão .ipa.

Em seguida o processo copia o arquivo .ipa do Mac para o equipamento de desenvolvimento e o deixa na pasta especificada na janela de Output do IDE de GeneXus.

Por último, na máquina de desenvolvimento é aberto automaticamente o iTunes para mostrar a aplicação no catálogo de aplicações.



Com a opção Build IPA (local) é transferido o projeto XCode para um Mac e é compilada a aplicação usando o método especificado na propriedade Distribution Method, gerando um arquivo \*.ipa. Em seguida o arquivo é copiado para a pasta associada ao environment da KB.

Se na propriedade Distribution Method, for escolhido o valor Development, a aplicação é compilada e assinada com o perfil de aprovisionamento de desenvolvimento para ser enviada à equipe de Teste.

Se for escolhida App Store, a aplicação é compilada e assinada com o perfil de aprovisionamento de desenvolvimento para ser publicada no TestFlight e na Apple Store.

TestFlight é uma aplicação da Apple que permite aos desenvolvedores de aplicações convidar avaliadores internos ou externos para que testem suas aplicações e, desta forma, coletar comentários e dados relacionados à funcionalidade de seu projeto.

Se selecionarmos Enterprise, a aplicação é compilada e assinada para distribuição interna.

E se escolhermos Ad Hoc, a aplicação é compilada e assinada com um perfil de aprovisionamento criado para um caso concreto, para ser distribuída depois aos testadores da aplicação.

GeneXusby Globant

### Prototyping in Apple on GPN

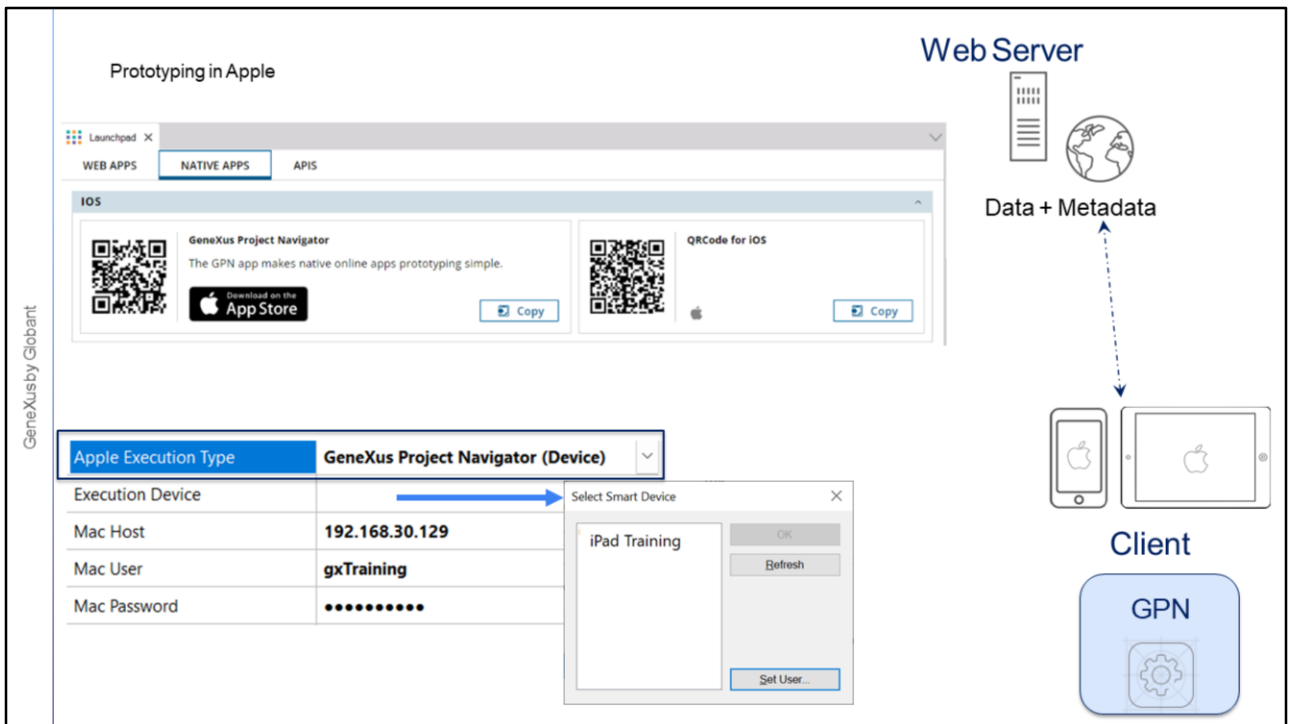
Apple Specific	
Required Xcode version	14.0
Flexible client version	1.2.1
iOS Deployment Target	12.0
tvOS Deployment Target	12.0
watchOS Deployment Target	5.0
Apple Execution Type	GeneXus Project Navigator (Device) ▾
Execution Device	
Mac Host	192.168.30.129
Mac User	gxTraining
Mac Password	*****

A última opção do combo e que é a opção padrão, diz GeneXus Project Navigator (Device).

Isto nos permite prototipar em um dispositivo físico utilizando o GeneXus Project Navigator, que deverá ser baixado da Apple Store.

Neste caso não precisamos ter um computador MAC para compilar a aplicação e seu comportamento é análogo ao que já vimos para Android.

As limitações do GPN para prototipar na plataforma Apple são as mesmas que já mencionamos.



O link para baixar o GPN da Apple Store está disponível no Launchpad. Lá também aparece o QRCode da URL da aplicação, que utilizaremos a partir do GPN instalado no dispositivo para registrar a aplicação.

Quando usamos a opção GeneXus Project Navigator (Device), devemos atribuir o dispositivo onde vamos prototipar. Na propriedade Execution Device podemos escolher algum dos dispositivos associados à nossa conta de GeneXus Account. Para isso, devemos primeiro abrir a caixa de diálogo da propriedade Execution Device, pressionar o botão Set User, fazer login com nossa conta GeneXus e depois escolher o dispositivo da lista.

Uma vez que catalogamos a aplicação no GPN (inserindo a URL manualmente ou através do QRCode), cada vez que executamos a partir de GeneXus, a aplicação é atualizada automaticamente com as alterações realizadas em GeneXus, graças a uma Push Notification recebida a partir do server.

Esta forma de prototipar tem a vantagem de vermos a aplicação em um dispositivo Apple real, sem necessidade de ter um MAC, nem de nos registrarmos como desenvolvedores no site da Apple.



Neste vídeo vimos a flexibilidade que temos para prototipar aplicações nativas, tanto no Android quanto na Apple.

No restante do curso utilizaremos o emulador Android para prototipar nossa aplicação.