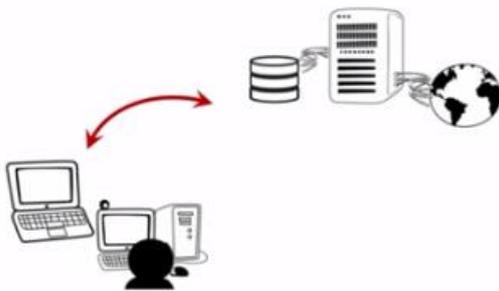


Características de las aplicaciones móviles



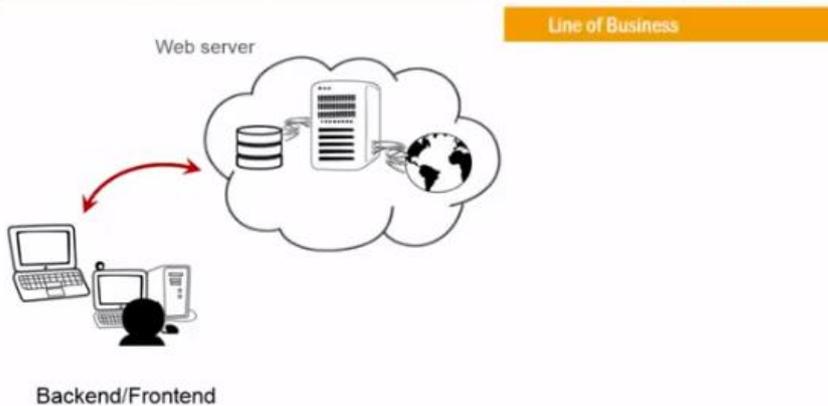
En el video de introducción vimos la aplicación a la que queremos llegar.

Ahora vamos a abordar las principales características de las aplicaciones para Smart Devices.

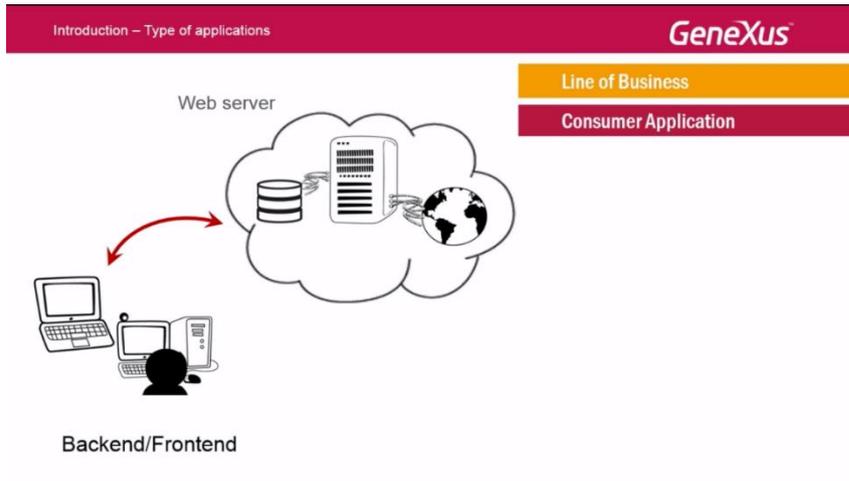


Toda aplicación para Smart Devices que desarrollemos tendrá necesariamente una parte ejecutándose en un servidor web.

No sólo en el caso de una aplicación Line of Business



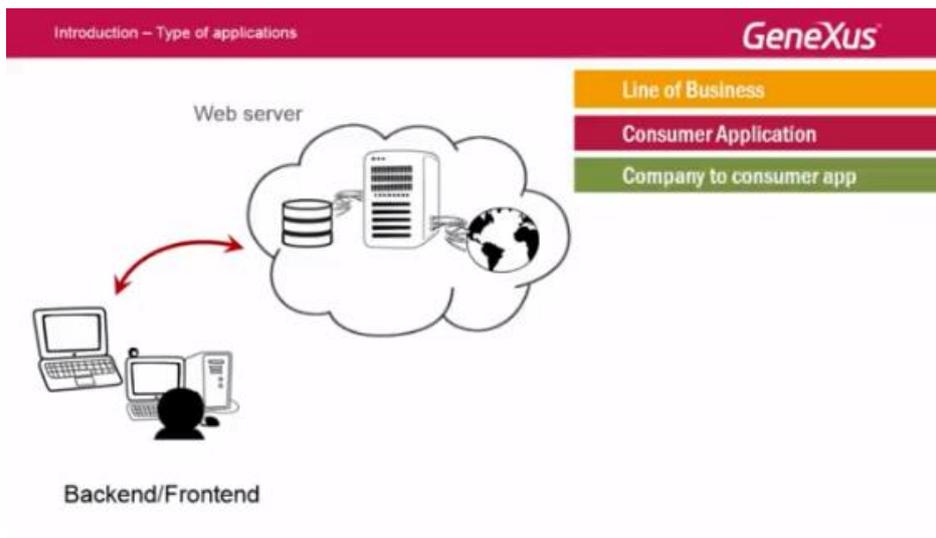
es decir una aplicación para Smart Device, que es sólo una parte de la aplicación empresarial; sino también en el de las Consumer Applications:



es decir aplicaciones hechas para ser subidas a los mercados de las distintas tecnologías, y ser descargadas por los usuarios tanto en forma paga como gratuita.

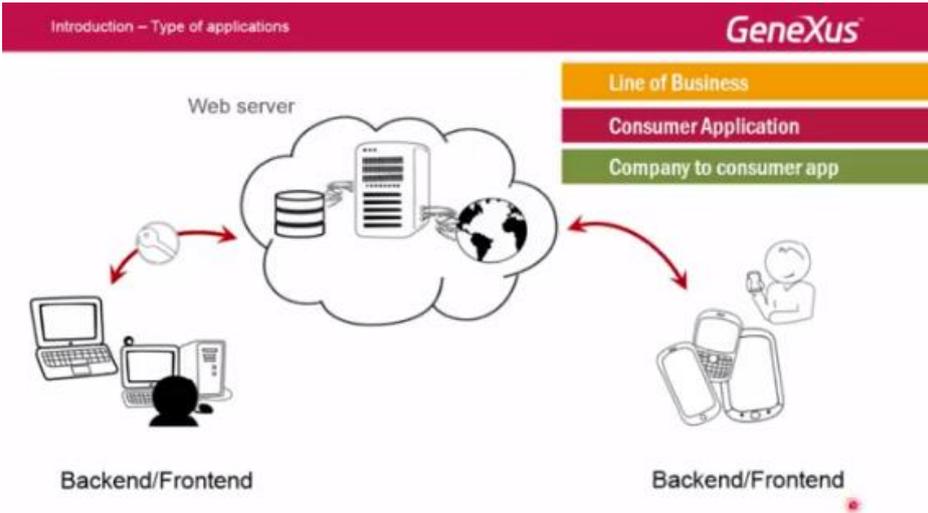
Aquí el foco está en la persona.

Entre estas dos, se encuentran las Company to consumer applications:

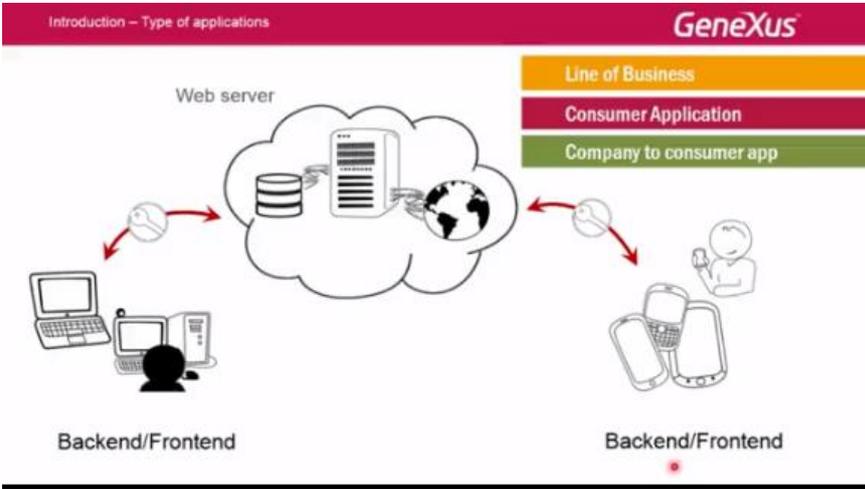


aplicaciones desarrolladas por la empresa para sus clientes y para ampliar su base de usuarios.

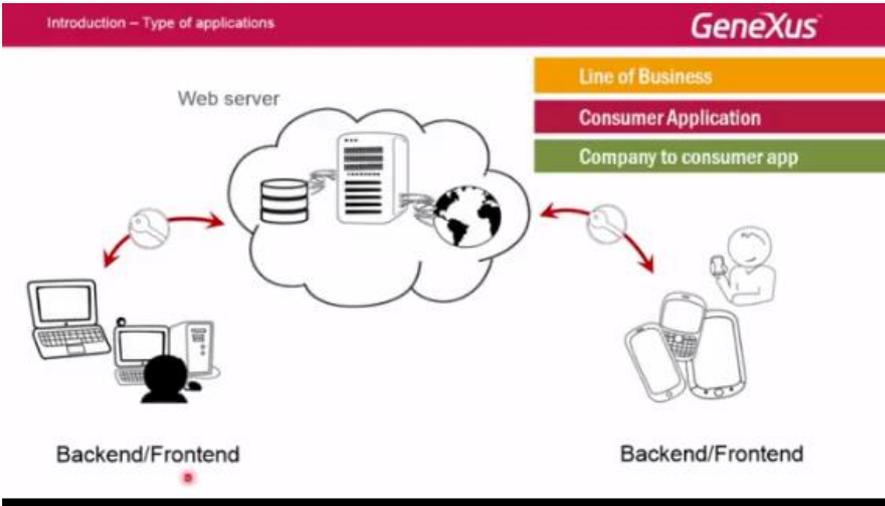
Nuestra aplicación EvenDay puede considerarse un híbrido, puesto que tendrá una parte cuyo foco esta puesto en los asistentes del evento en cuestión: el "Frontend":



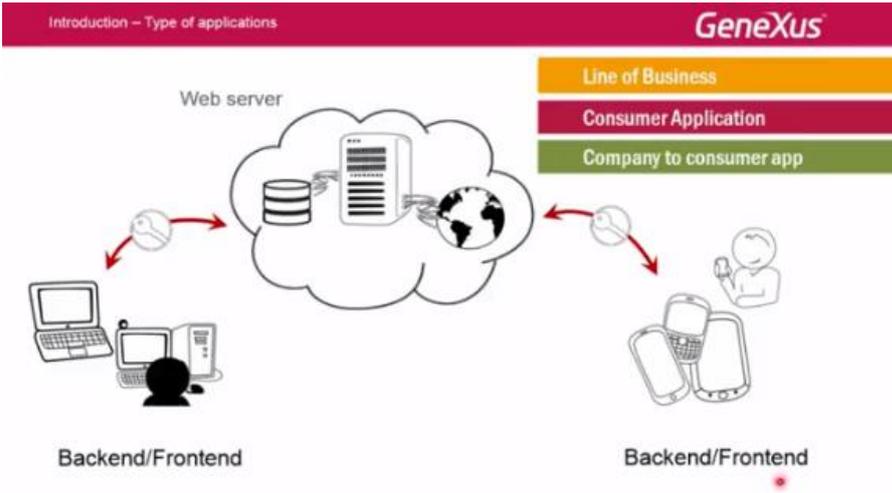
y otra parte que será utilizada por los organizadores del evento para poder modificar datos de la base de datos de forma móvil: el “Backend”:



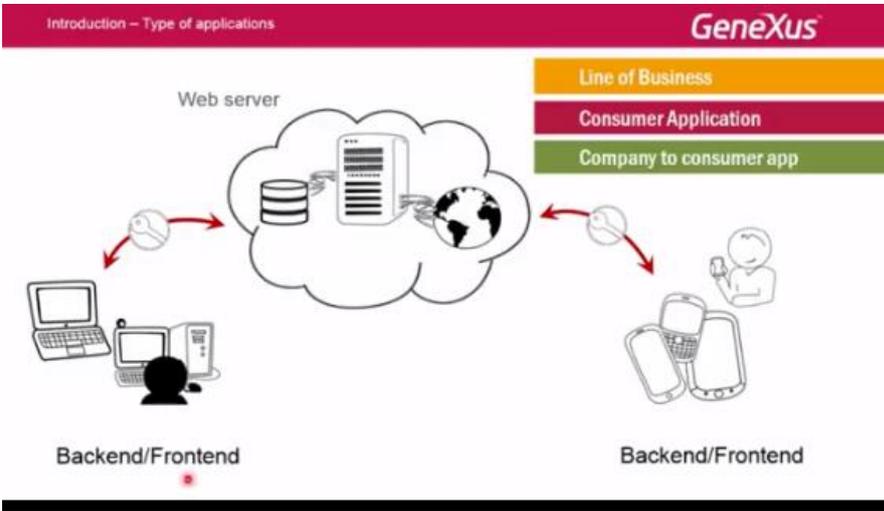
En definitiva, así como queremos publicar las conferencias, oradores y demás a través de un sistema web



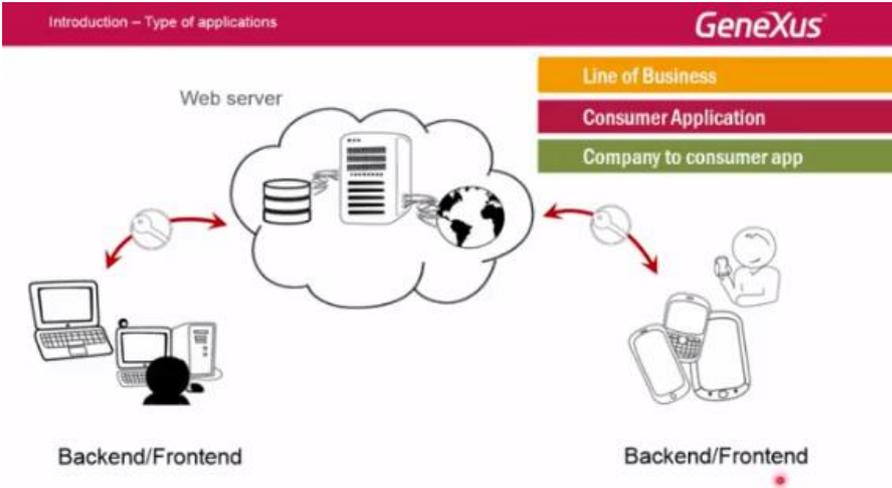
también queremos brindar esas funcionalidades y más a través de una aplicación móvil:



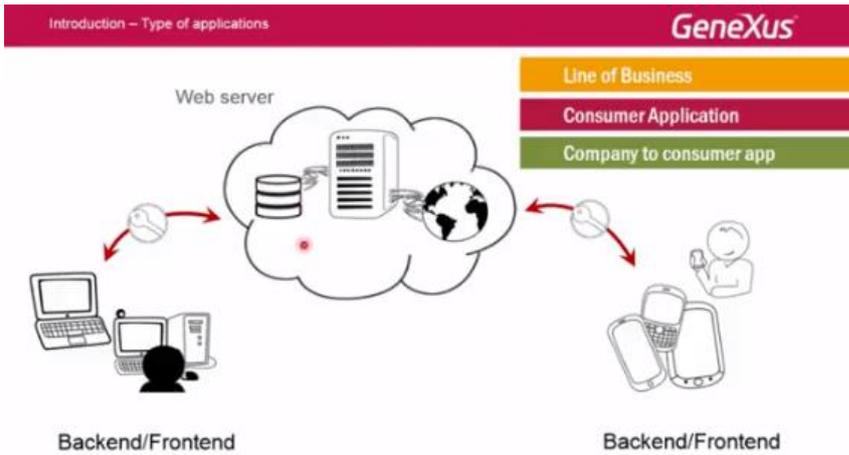
Y de igual manera, así como necesitamos un sistema backend web para que los usuarios autorizados puedan administrar la información del sistema:



podemos necesitar también que todas o parte de esas manipulaciones de los datos, puedan realizarse por esos mismos usuarios autorizados en la aplicación para el Smart Device:

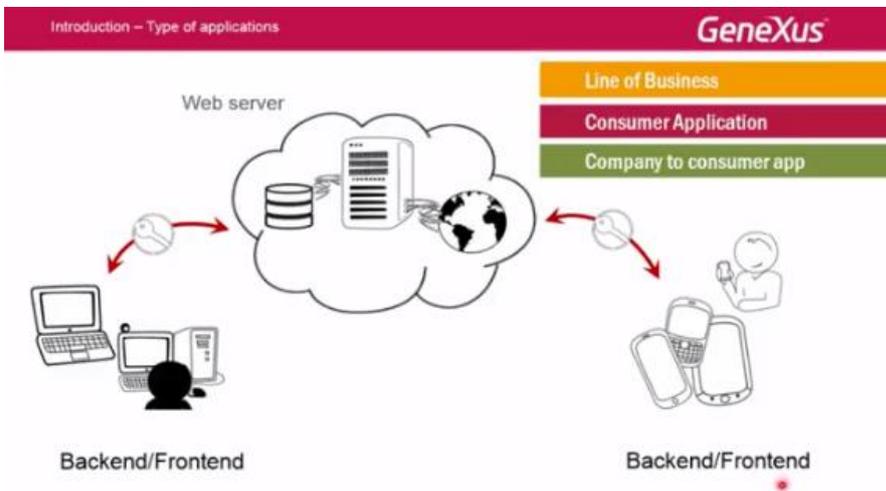


De este modo se podrán estar haciendo cambios en los datos de la base de datos centralizada

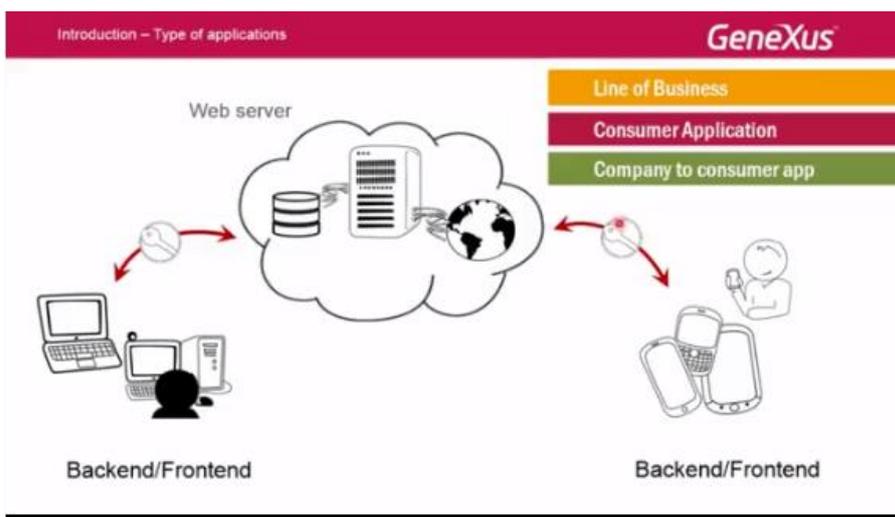


desde el mismo lugar del evento, un minuto antes de que una conferencia de inicio.

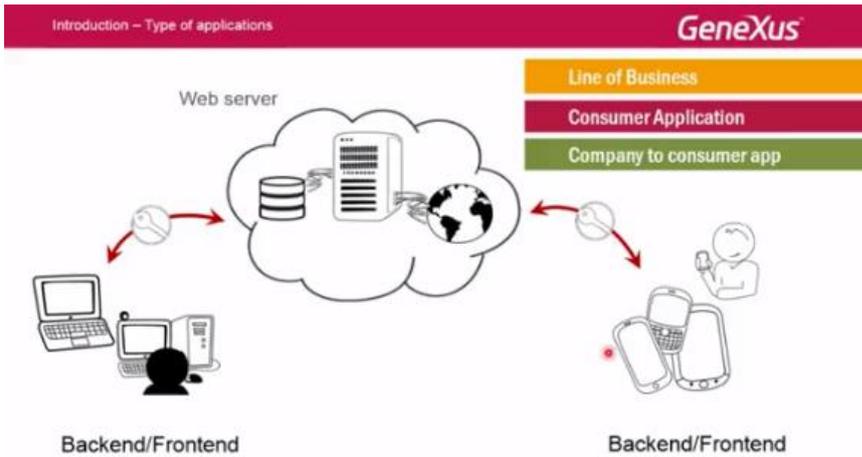
No habrá casi diferencias funcionales entre la implementación del backend y la del fronted.



La diferencia estará dada que para los usuarios autorizados

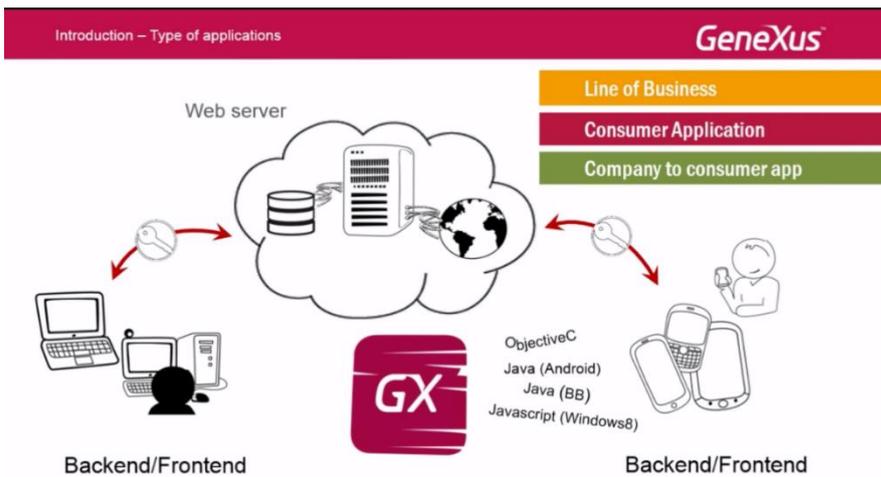


para los que deberemos tener seguridad, se permitirán las operaciones de CRUD (Create, update, deteled)



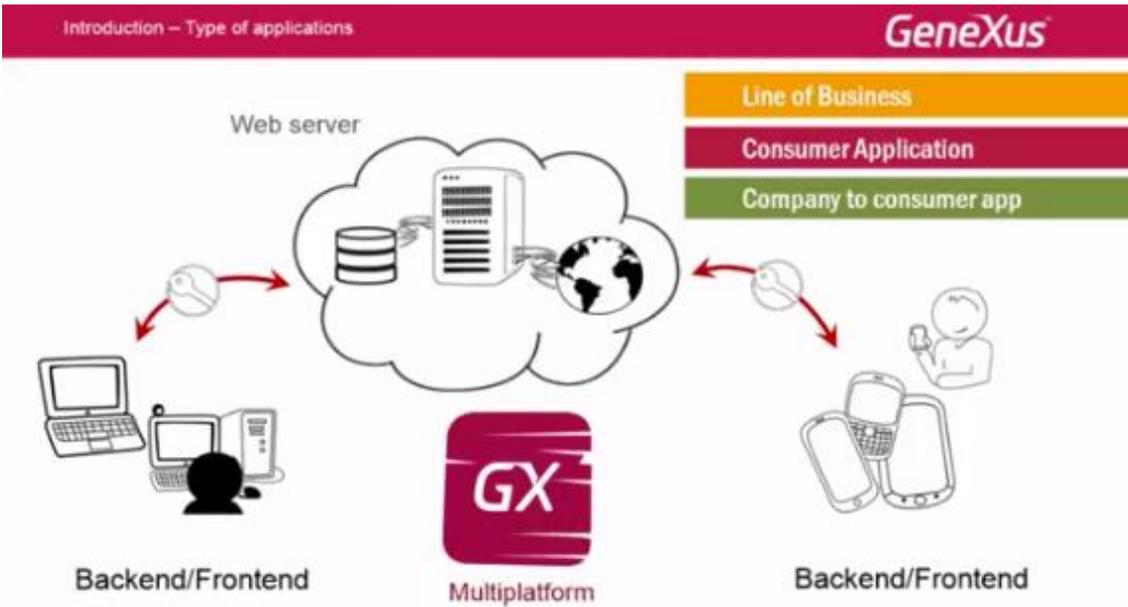
sobre algunos datos a través de pantallas que para usuarios no autorizados no estarán disponibles.

Así como la parte web se desarrollaba en GeneXus y este luego la implementaba en alguno de los lenguajes disponibles: Ruby, Java o .NET, lo que significaba que GeneXus era múltiple plataforma, la parte móvil se desarrollará análogamente

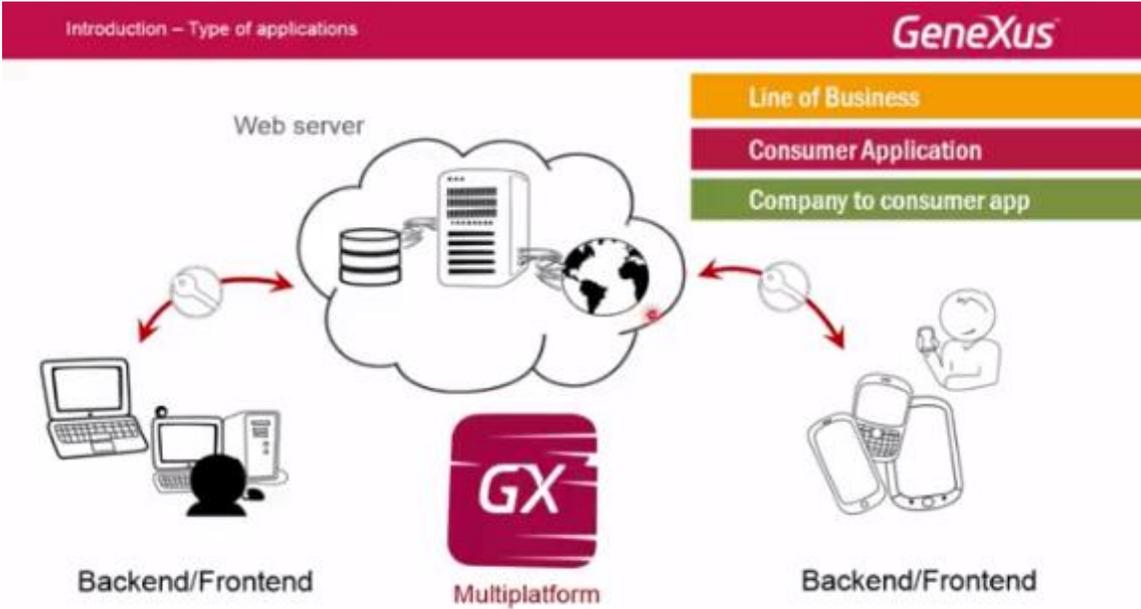


siguiendo la lógica de GeneXus que luego la implementará en el lenguaje de programación de la plataforma elegida y siguiendo los estándares de interfaz y comportamiento de esa plataforma.

Las plataformas sostenidas son las más extendidas: iOS, Android, Blackberry y Window8. Por tanto, GeneXus sigue siendo multiplataforma también para lo que hace a las aplicaciones para dispositivos inteligentes.

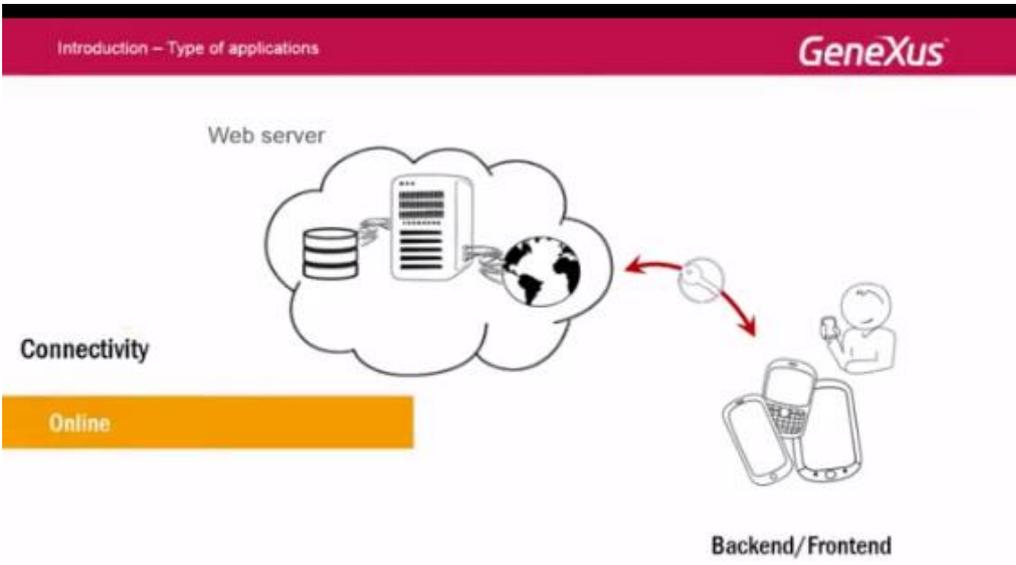
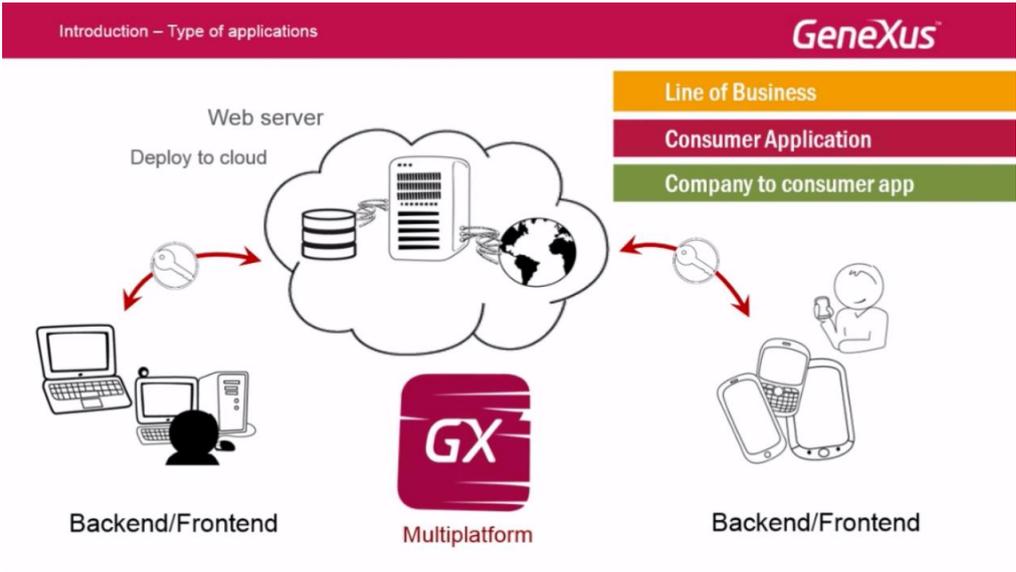


Para simplificar la tarea de prototipación, GeneXus nos ofrece la posibilidad de desarrollar la parte web en una nube

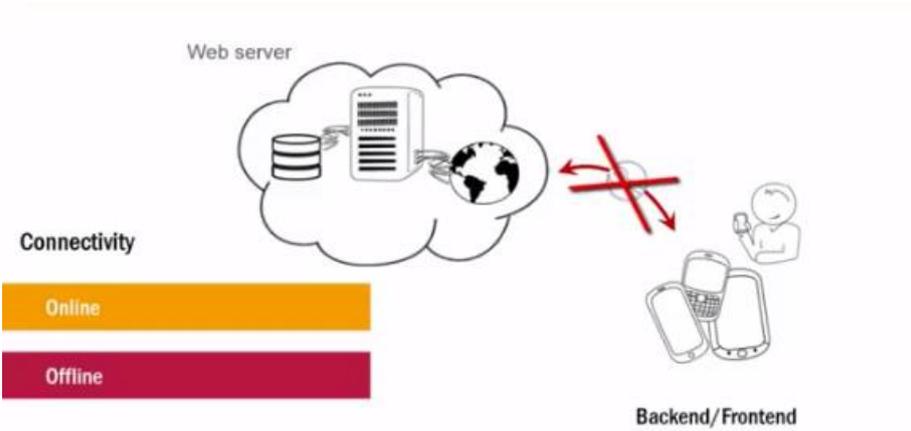


de modo que siempre que tengamos acceso a internet podremos estar probando la aplicación desde cualquier lugar y todo el software, programas y base de datos, estarán hospedados allí sin que tengamos que preocuparnos por la infraestructura.

Esto se hace a través de la propiedad de **Deploy to Cloud** del generador web



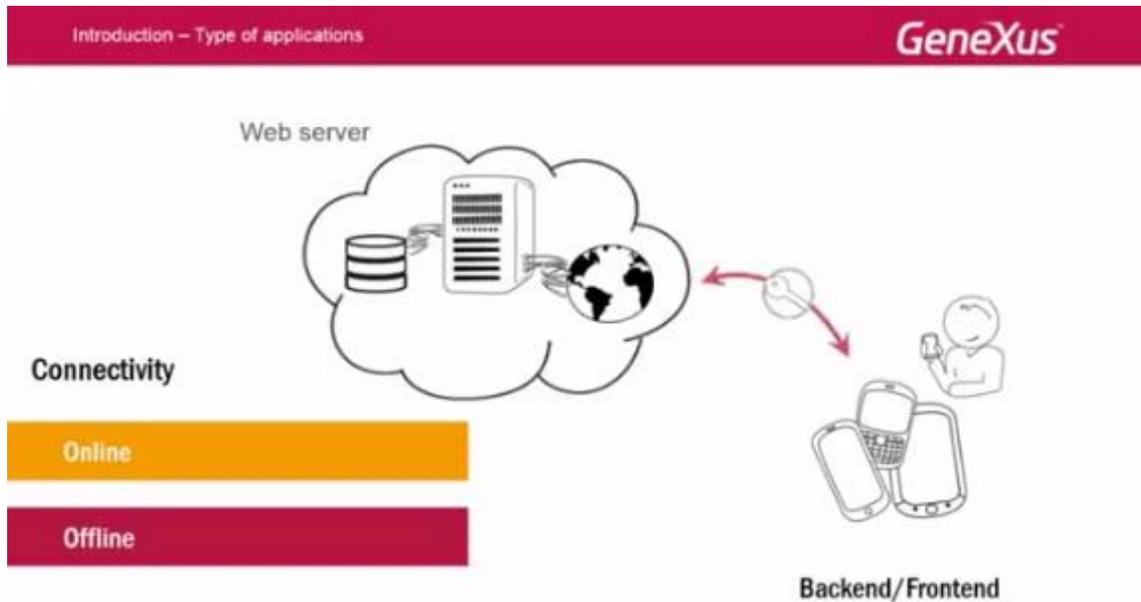
Un escenario bien importante para los dispositivos inteligentes, sobre para algunos tipos de aplicaciones, es permitir que la aplicaciones o parte de ella se siga ejecutando cuando se encuentra desconectada de Internet.



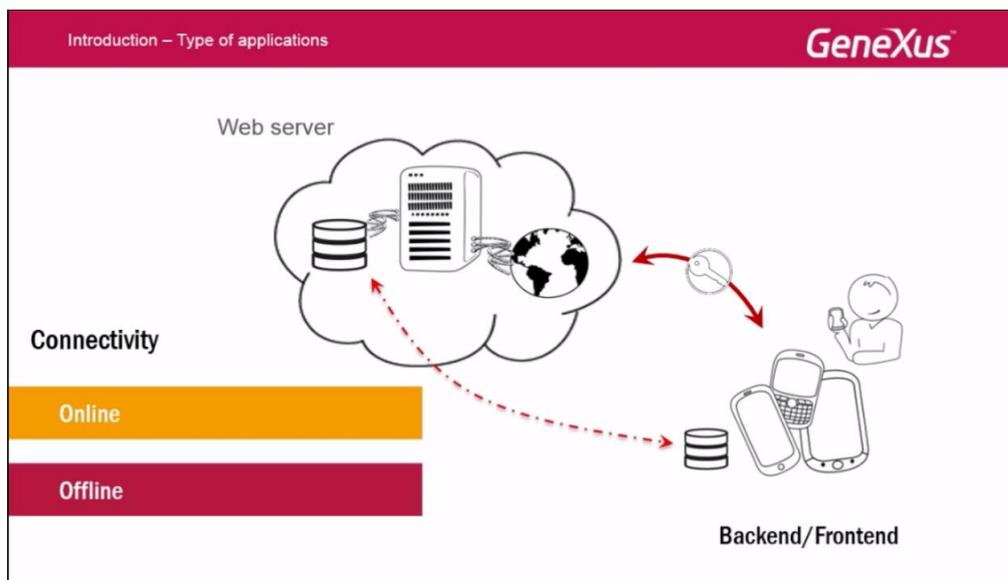
Piéñese por ejemplo cuando se está trabajando en zonas rurales o cuando un vendedor tiene que ir hacer órdenes de compra en la calle, en los diferentes locales.

En el caso de nuestra aplicación vamos a querer que el usuario pueda seguir viendo **toda la agenda de conferencias** y **toda la información relacionada** incluso cuando pierda la conexión.

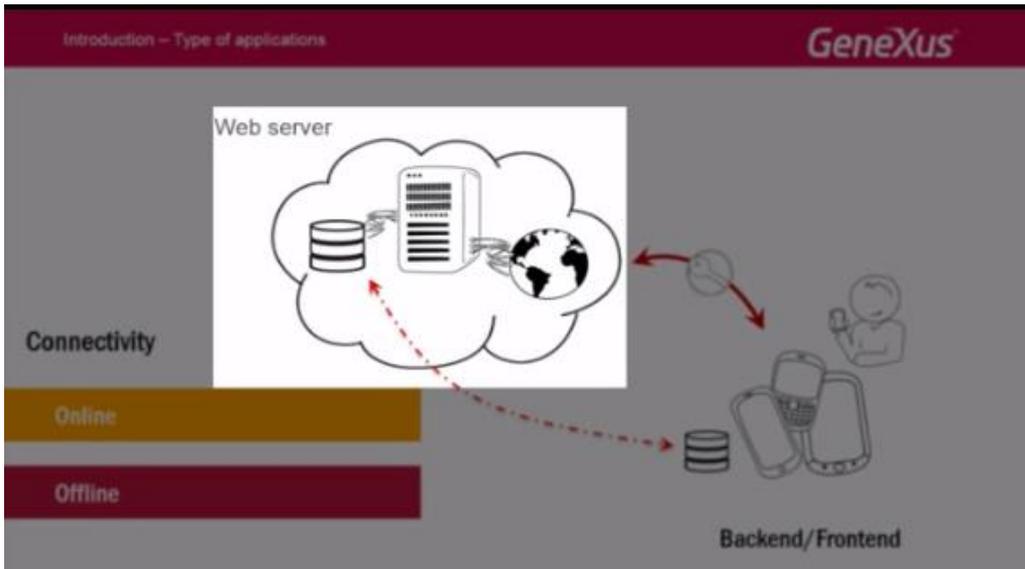
Luego cuando ésta se reestablece automáticamente la aplicación actualizará sus datos,



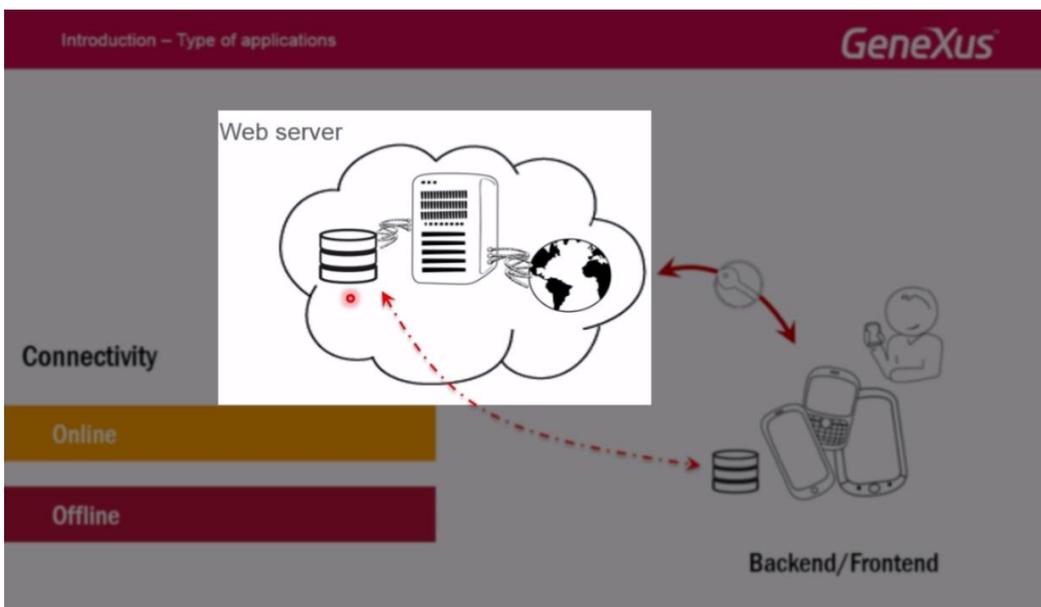
estarán en una base local en el dispositivo sincronizándose con el server, recibiendo la información de éste y también enviando información, como por ejemplo cuando el usuario haya marcado algunas conferencias como favoritas -estando desconectado- y esa información queremos que se envíe al servidor en el momento de recuperar la conexión.



Sin embargo, habrá tareas que requerirán necesariamente el acceso al servidor web



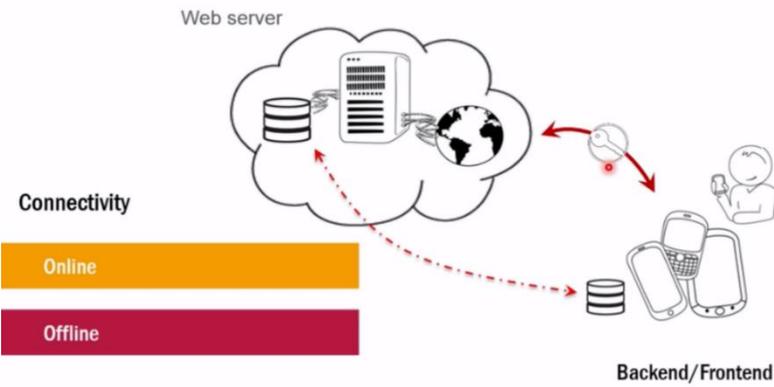
ya sea por su sensibilidad, tareas que deben ser validadas en la base de datos centralizada...



como por lo rápido que cambian los datos.

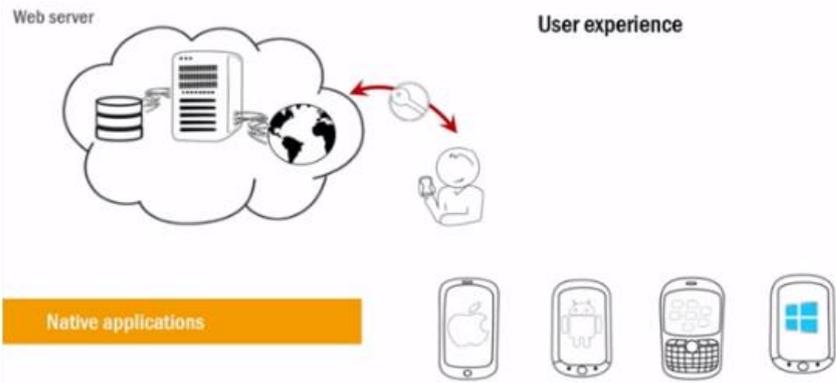
Estas tareas deberán ejecutarse **online**.

En nuestro caso, el login deberá ser con conexión

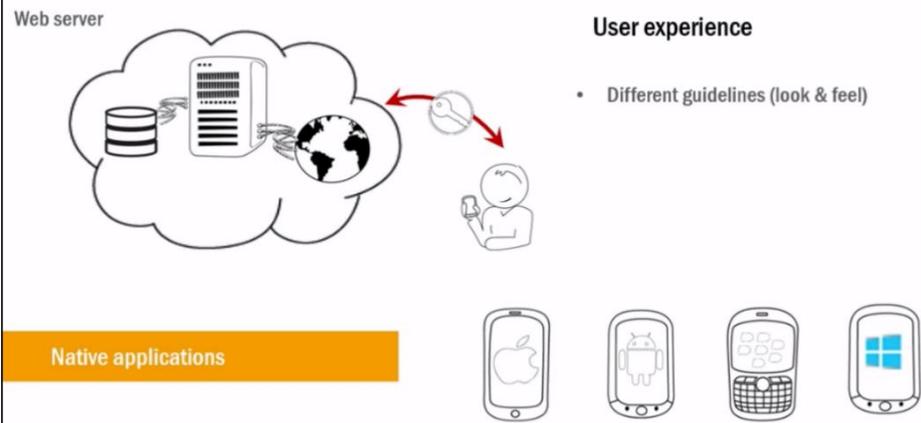


y el panel que muestran los tweets es deseable que también lo sea.

Por tanto, podremos elegir qué objetos de la aplicación pueden ejecutarse offline y qué objetos no.

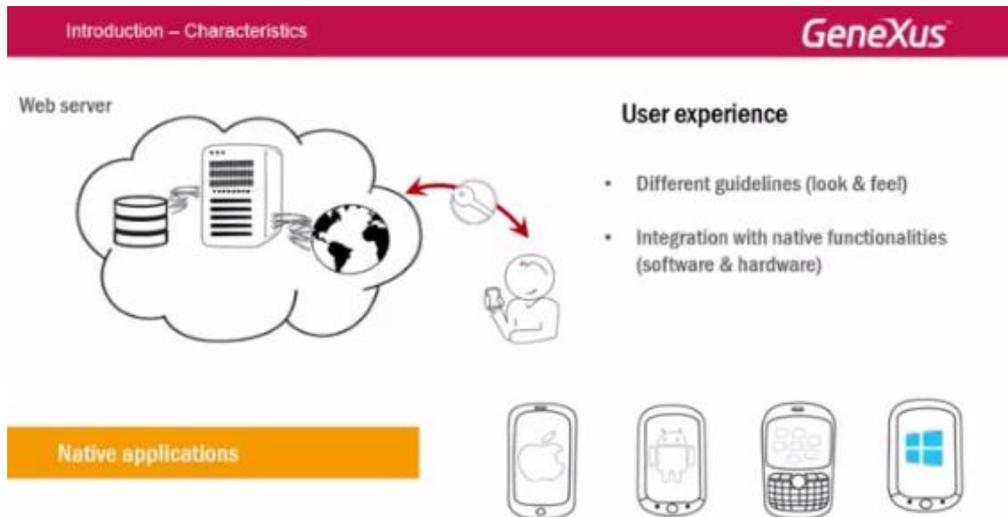


El usuario va a exigir que toda aplicación para su dispositivo tenga un Look & feel similar al resto de sus aplicaciones nativas.



Como por ejemplo su aplicación de contacto, su calendario, la forma de hacer back en el propio dispositivo, es decir **que cumpla las guidelines de su plataforma.**

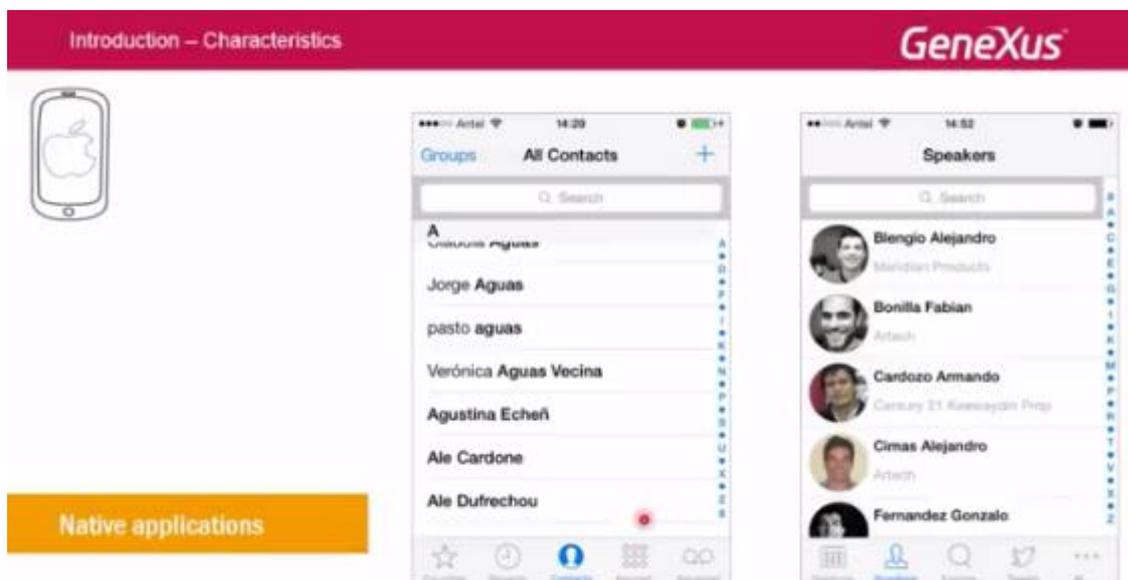
Además, va a exigir que se integre con las demás funcionalidades del dispositivo..



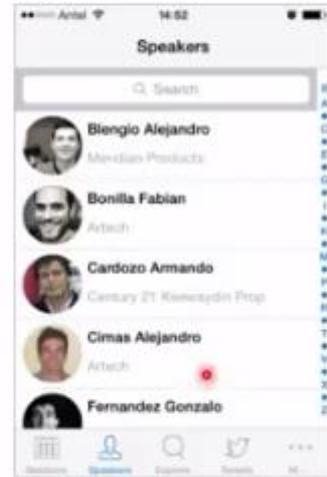
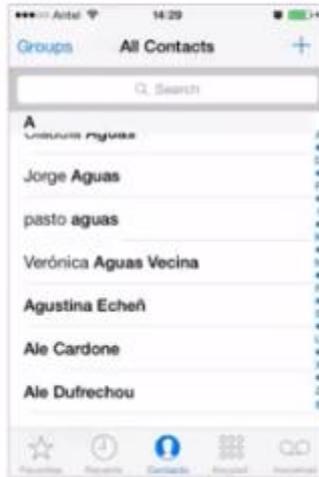
de software: como el calendario o la agenda de contactos

y de hardware: como la cámara de fotos, las llamadas telefónicas o el GPS.

Aquí vemos como se ve por ejemplo para un iphone 7 la aplicación nativa de los contactos:

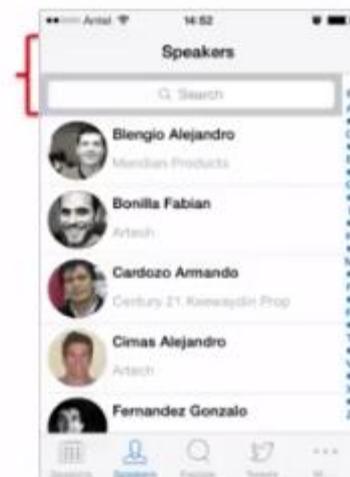
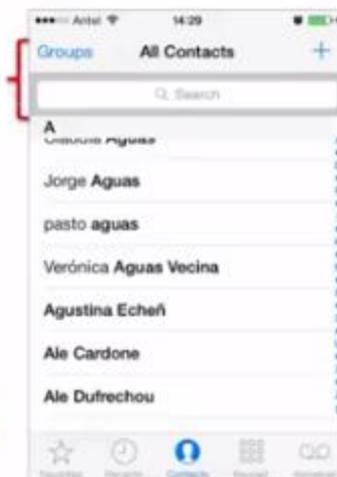


y cómo se ve la aplicación desarrollada con GeneXus “EvenDay”:



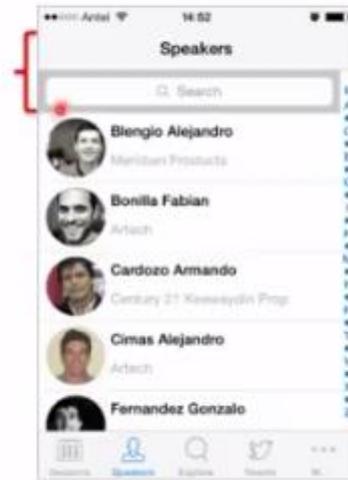
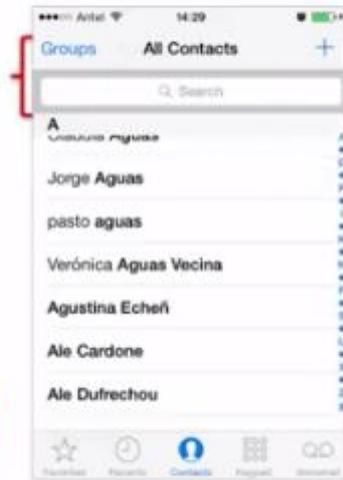
Native applications

Independientemente de las definiciones de diseño, colores y demás, si observamos bien tienen un determinado patrón común de apariencia. Una barra superior con algunas acciones



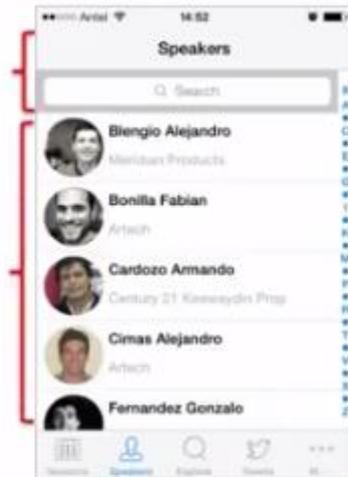
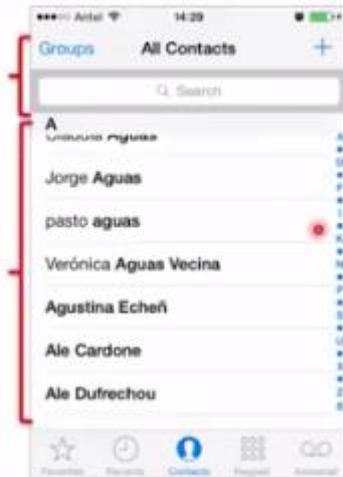
Native applications

Una opción de search



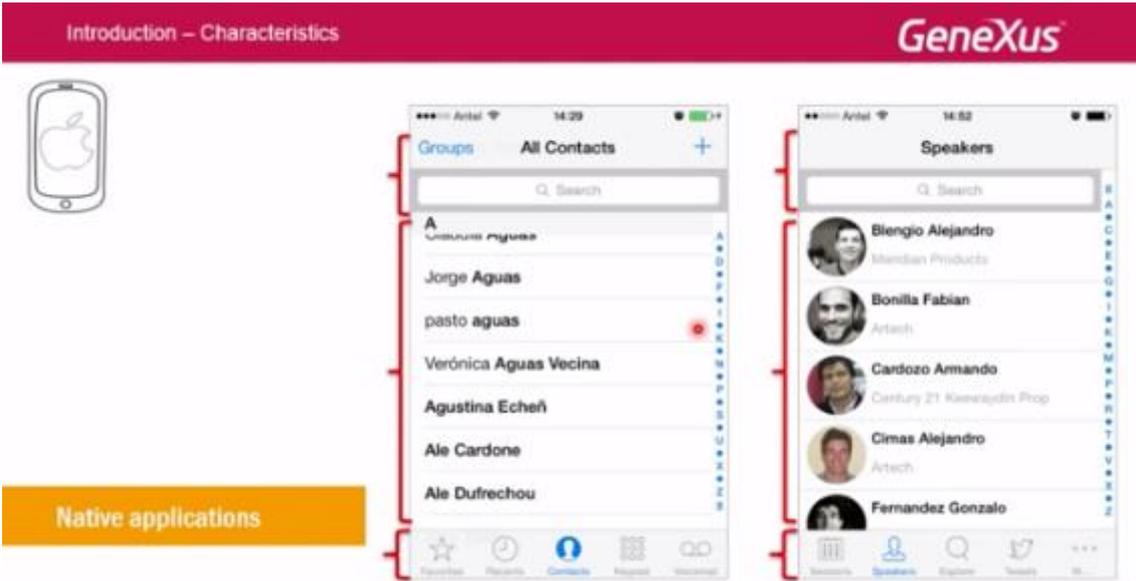
Native applications

La información en el centro:

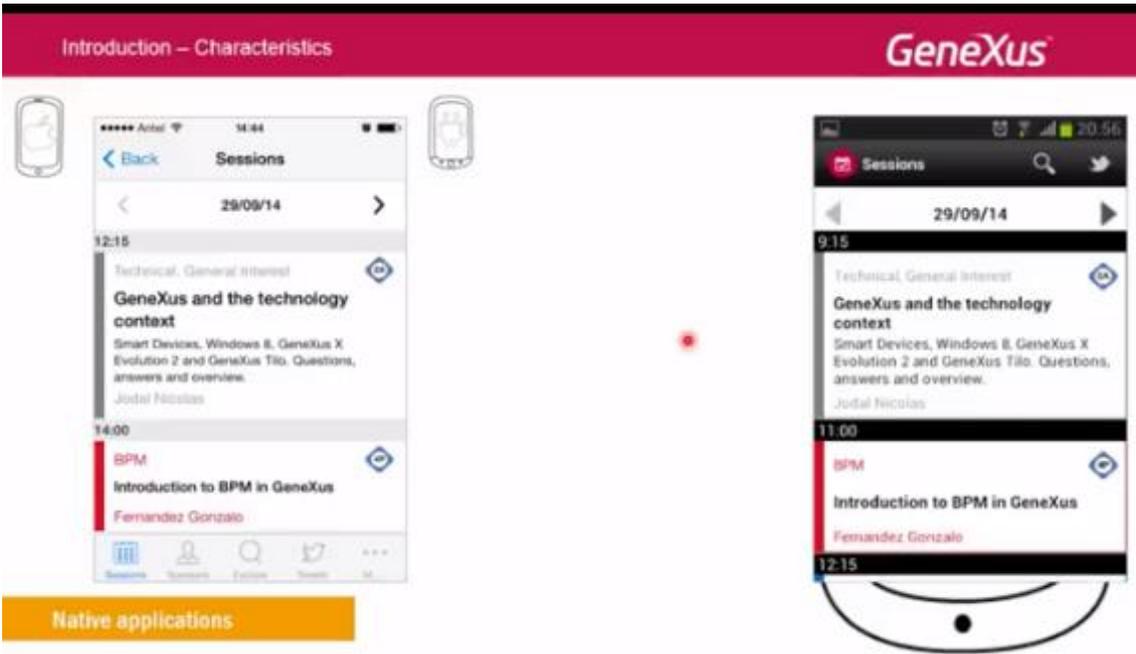


Native applications

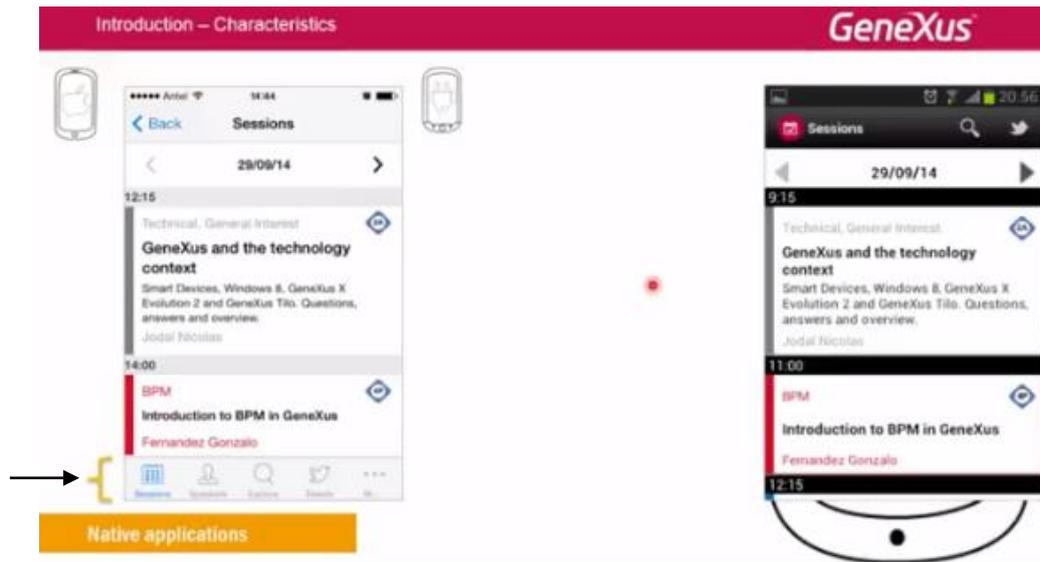
Y luego un menú en forma de tabs



para cambiar de opción al visualizar.

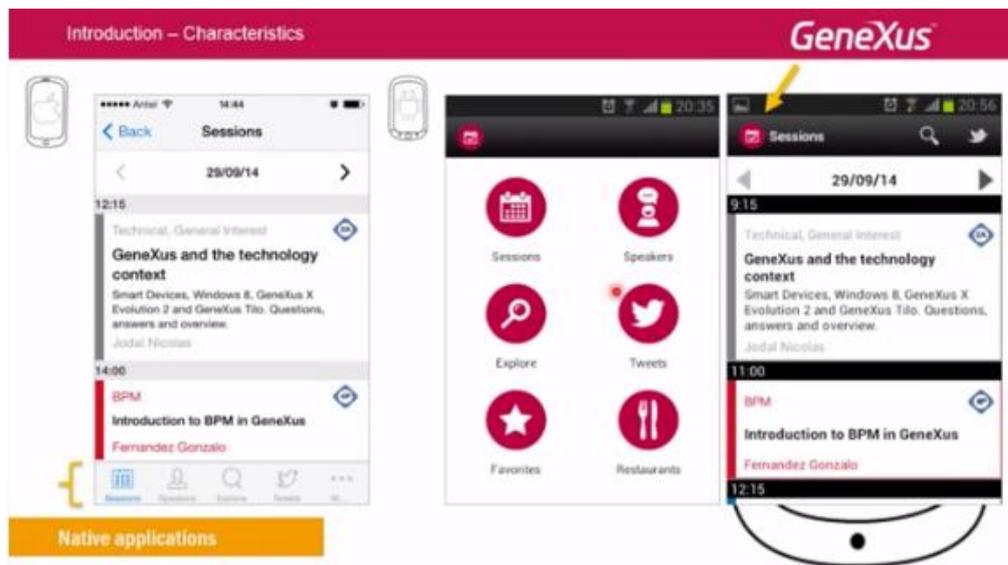


Sin embargo cuando cambiamos de dispositivo, si bien la aplicación es esencialmente la misma (muestra una lista de conferencias en este caso) tiene sutiles pero importantes diferencias estéticas y funcionales. Por ejemplo...en el caso de iOS, tengo el menú permanentemente visible:

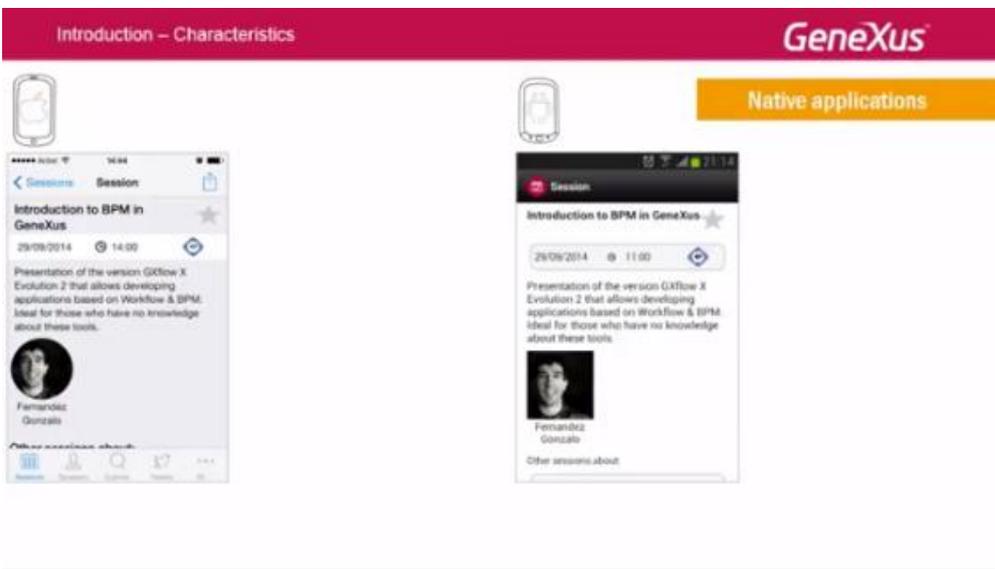
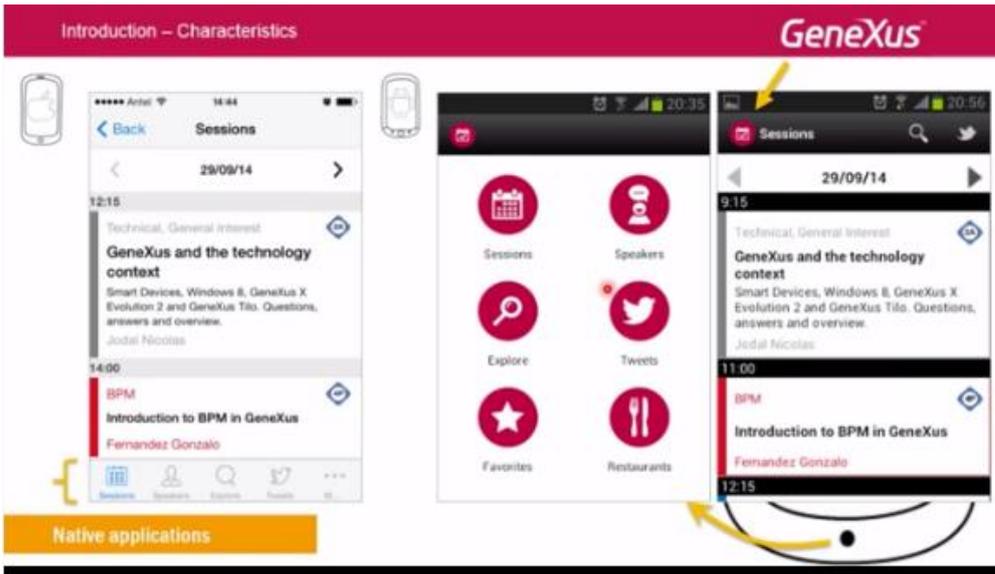


porque sus guidelines así lo indican. Mientras que en Android no.

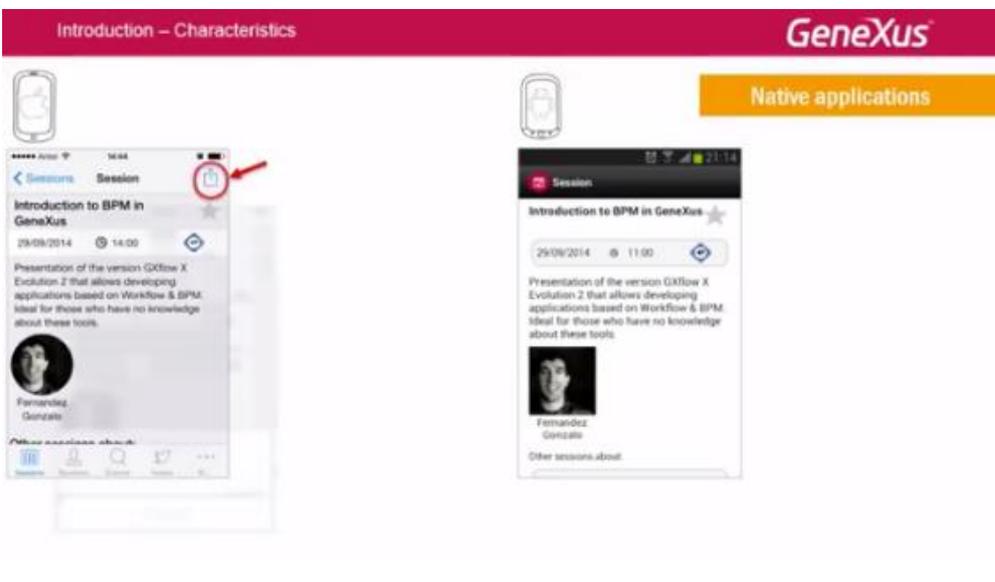
Para volver al menú se hace tap sobre el icono de la aplicación



o si la pantalla que se está visualizando fue directamente invocada desde el menú con el botón de back del propio dispositivo, se vuelve.



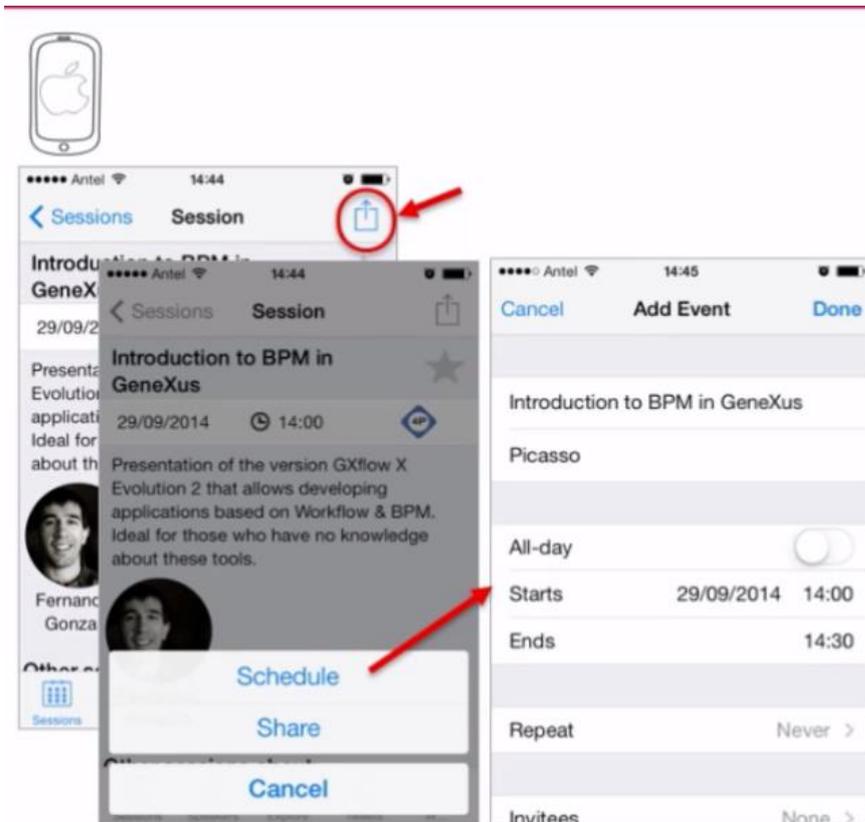
Aquí podemos ver cómo la aplicación para iphone, una vez posicionado sobre una conferencia, nos permite a través de un icono especial



ver las acciones que podemos realizar sobre esa conferencia

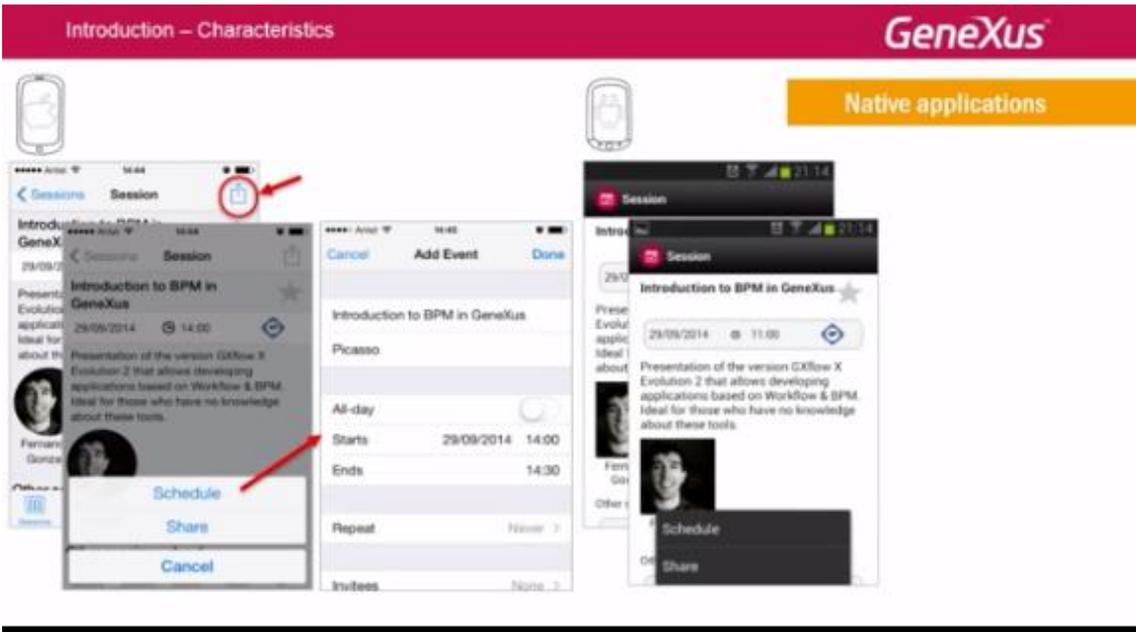


como agendarla en el calendario del dispositivo

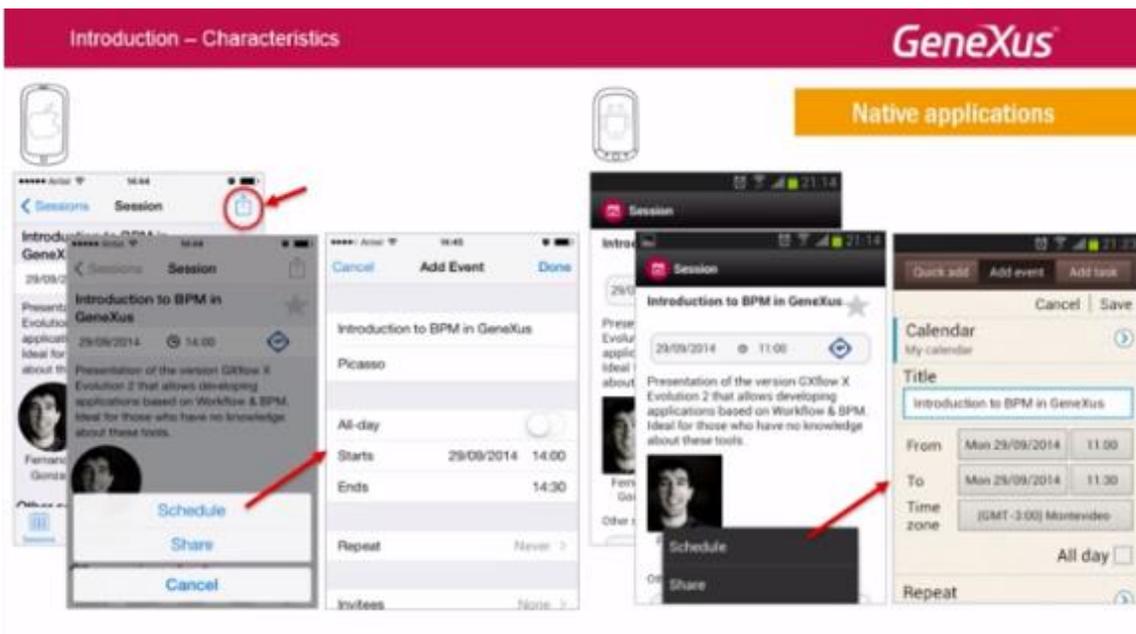


lo que abrirá el programa nativo del dispositivo o compartirlo.

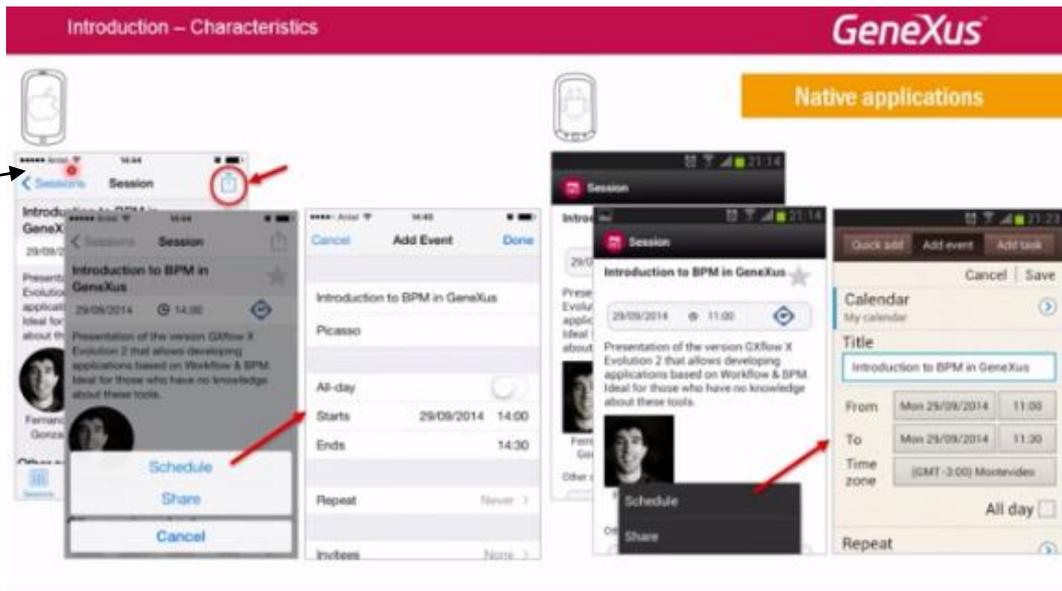
En cambio, la aplicación para un teléfono Android, nos permite acceder a las acciones a través de un botón especial en el propio dispositivo.



De igual manera cuando elijamos la opción Schedule nos abrirá el calendario del dispositivo Android.



Asimismo vemos cómo en iOS se incorpora un icono en la barra superior para poder volver al llamador



En cambio, en Android, se cuenta con el botón back del propio dispositivo.

En resumen, **es un requerimiento para las aplicaciones para Smart Devices, el que sean nativas.**

En el siguiente video nos introduciremos en el modelo conceptual de estas aplicaciones.

GeneXus

The Conceptual Model of mobile applications for Smart Devices

Veremos los cuatro tipos de pantallas que se necesitan para desarrollar una aplicación para Smart Devices, y los objetos que las implementan, y sus relaciones.

Los invitamos a verlo.

