

Características das aplicações móveis

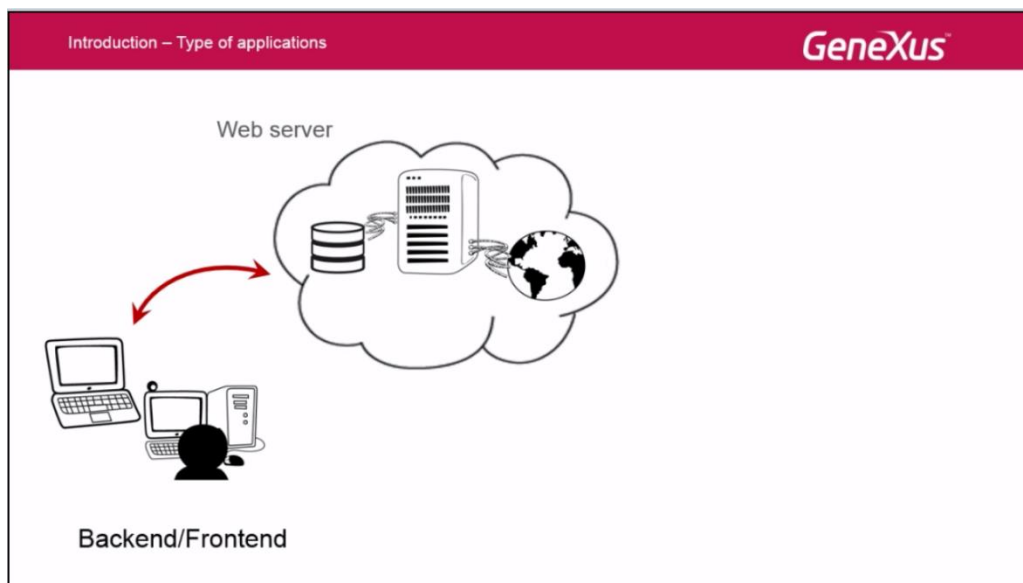


Characteristics of mobile applications for Smart Devices

Cecilia Fernández | GeneXus Training

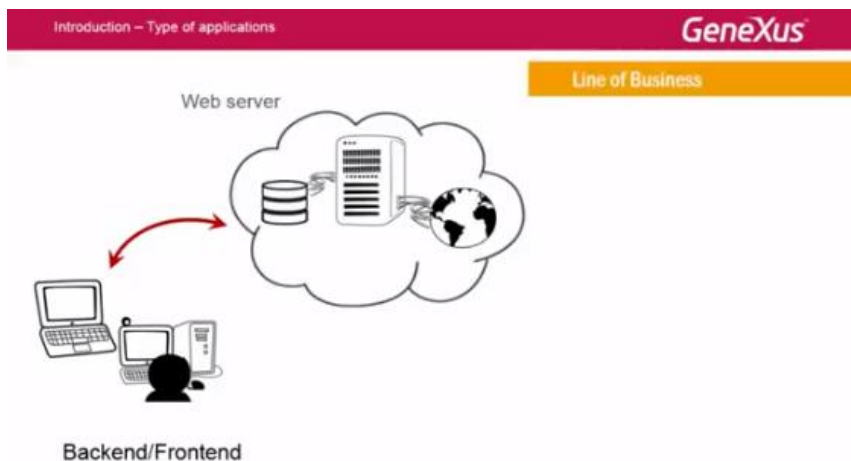
No vídeo de introdução vimos a aplicação onde queremos.

Agora vamos abordar as principais características das aplicações para Smart Devices

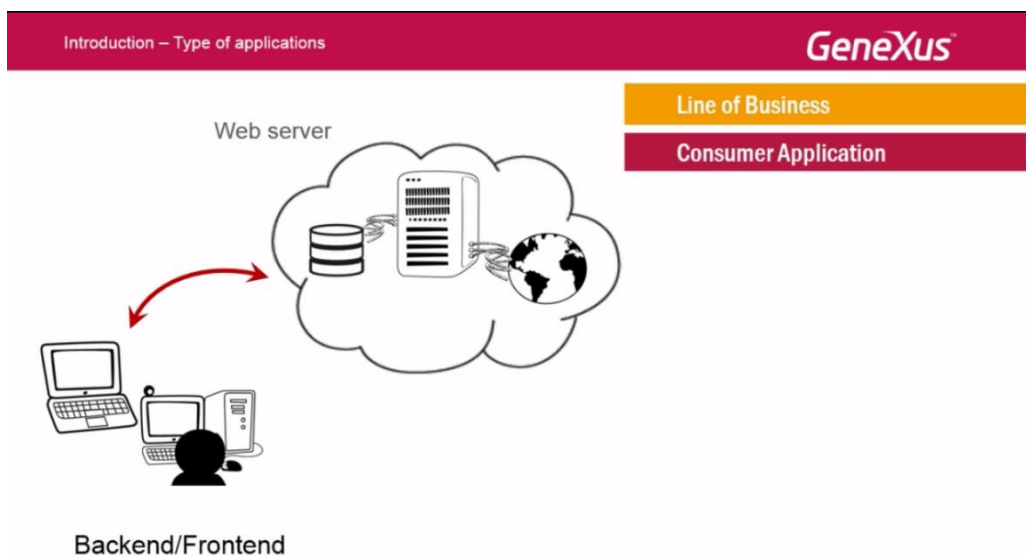


Toda aplicação para Smart Devices que for desenvolvida terá necessariamente uma parte rodando em um servidor web.

Não somente em caso de uma aplicação Line of Business



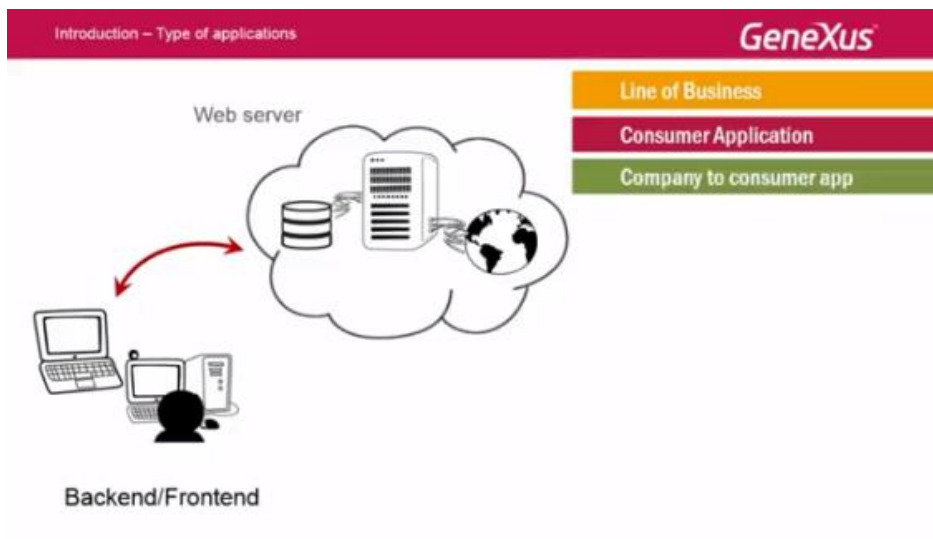
ou seja, uma aplicação para Smart Device, que é só uma parte da aplicação empresarial. Mas também no Consumer Application



ou seja, aplicações enviadas para os mercados de distintas tecnologias, e baixados pelos usuários tanto na forma paga como gratuita.

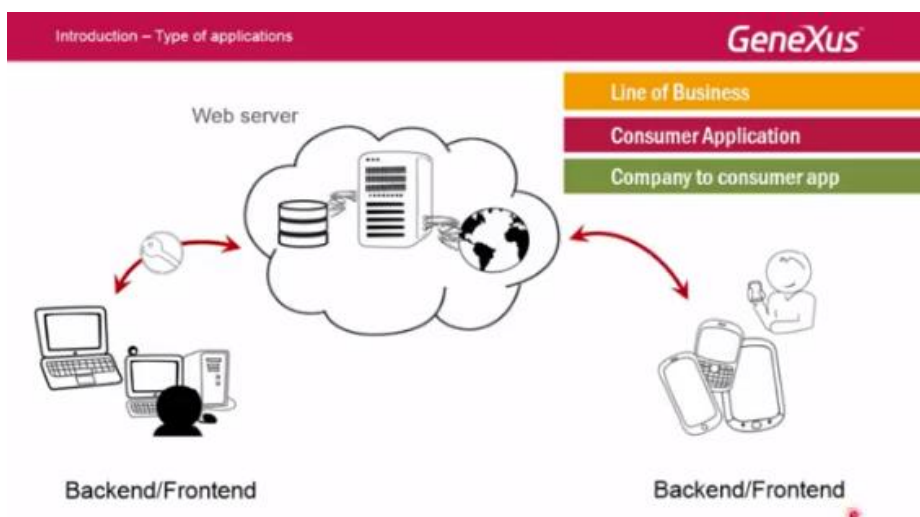
Aqui o foco está na pessoa.

Entre estes dois encontram-se a Company to consumer applications:

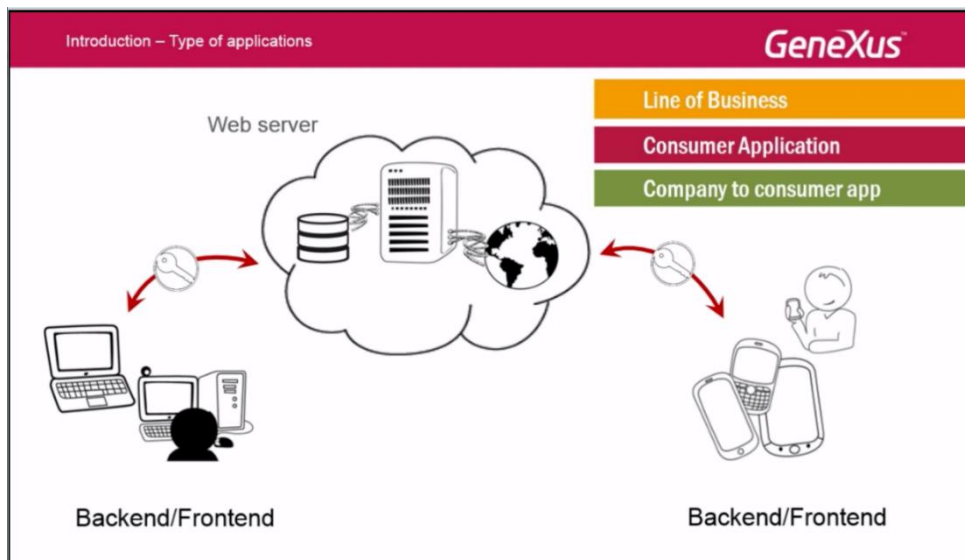


aplicações desenvolvidas pela empresa para seus clientes e para ampliar sua base de usuários.

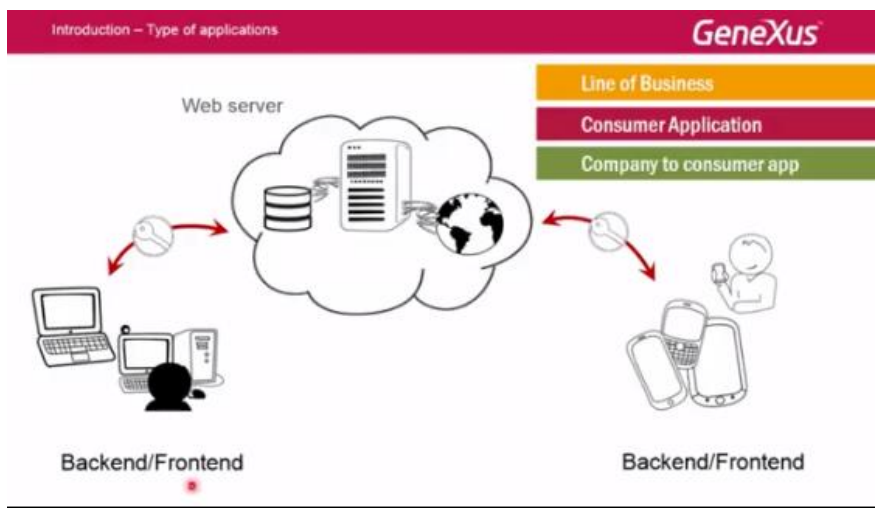
Nossa aplicação EvenDay pode ser considerada híbrida, desde que tenha uma parte cujo foco está desde os assistentes do evento em questão; “Frontend”



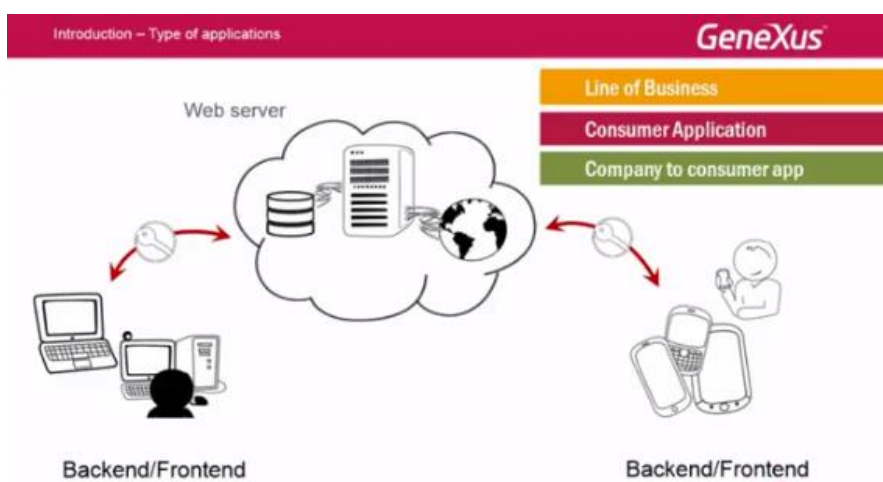
e outra parte que será utilizada pelos organizadores do evento para poder modificar dados das bases de forma móvel, no “Backend”



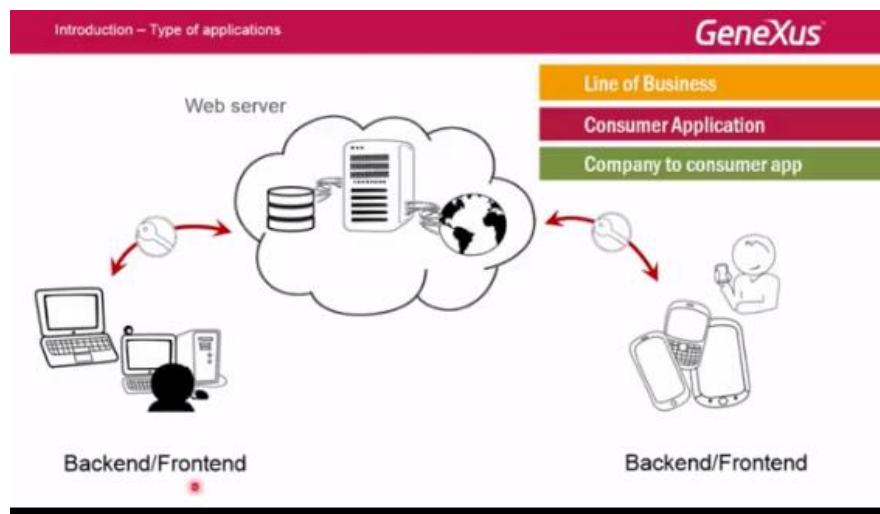
Definitivamente, assim como queremos publicar as conferências, oradores e demais através de um sistema web



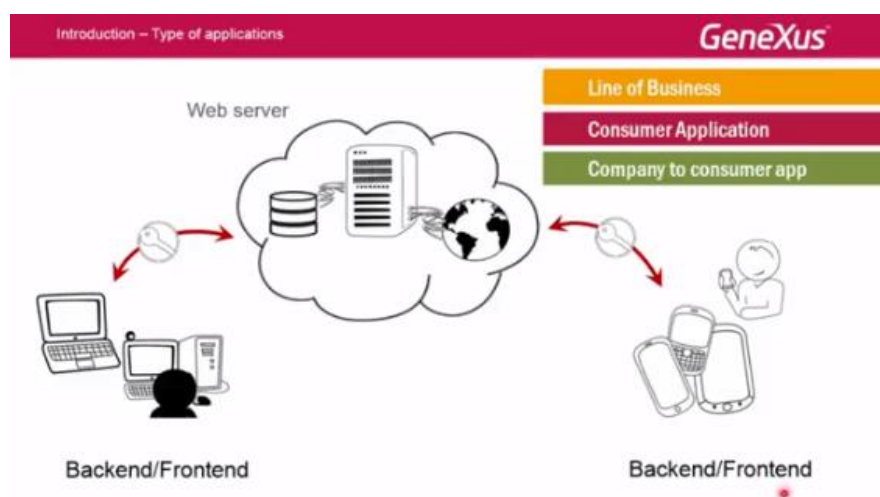
também queremos fornecer essas funcionalidades e mais através de uma aplicação móvel:



E da mesma maneira, assim como necessitamos de um sistema backend web para que os usuários autorizados possa administrar a informação do sistema:

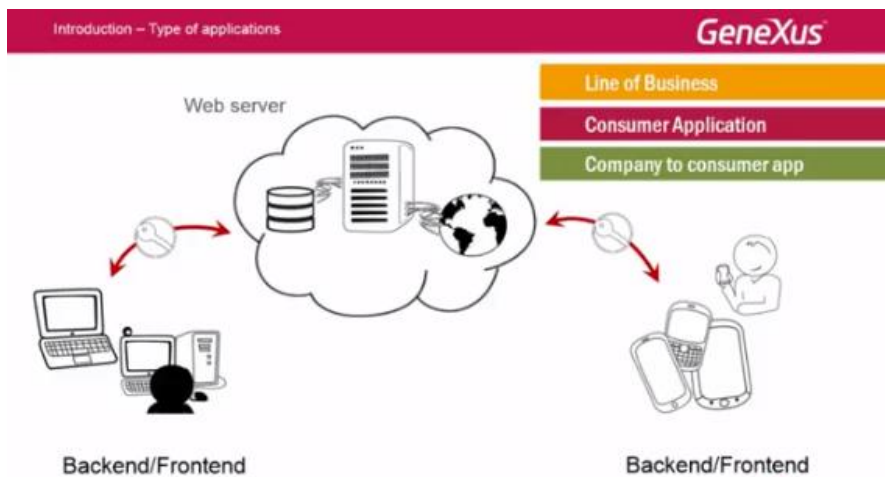


podemos necessitar também que toda parte dessas manipulações de dados, possa ser realizada por esses mesmos usuários autorizados na aplicação para o Smart Devices.

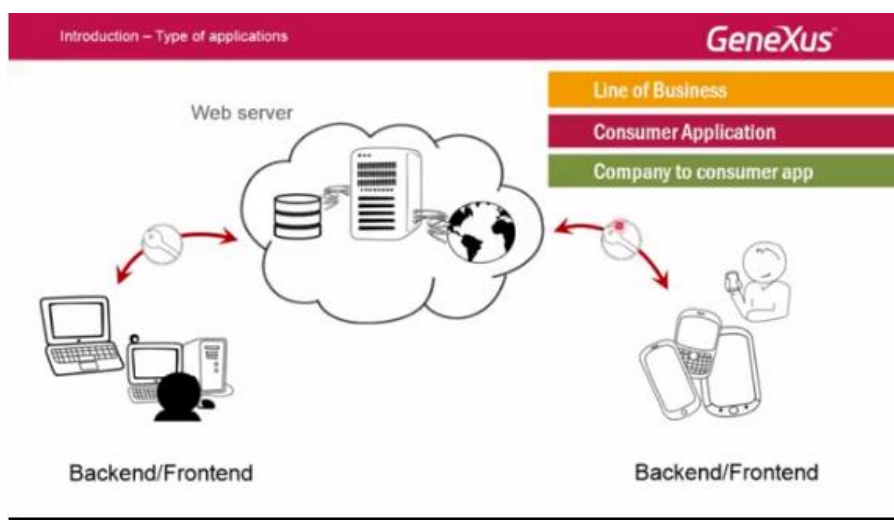


Deste modo, podemos fazer mudanças nos dados da base de dados centralizada, a partir do mesmo local do evento, um minuto antes de uma conferência ter início.

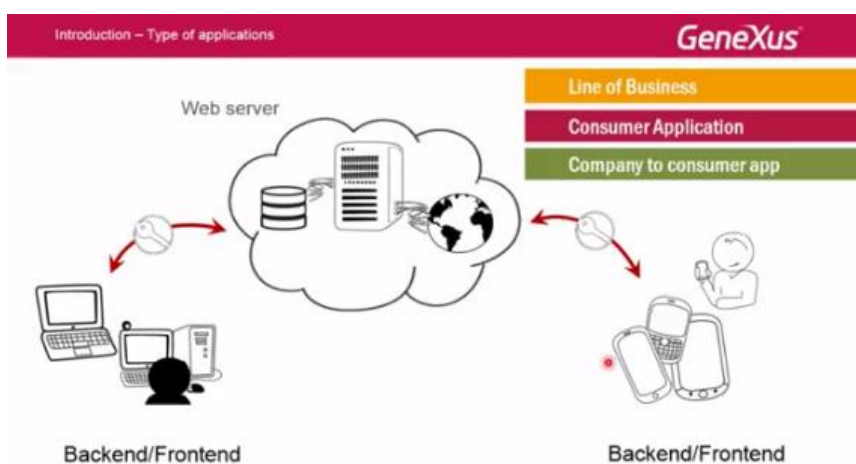
Não haverá quase diferenças funcionais entre a implementação do backend e a do frontend.



A diferença será, dada para os usuários autorizados

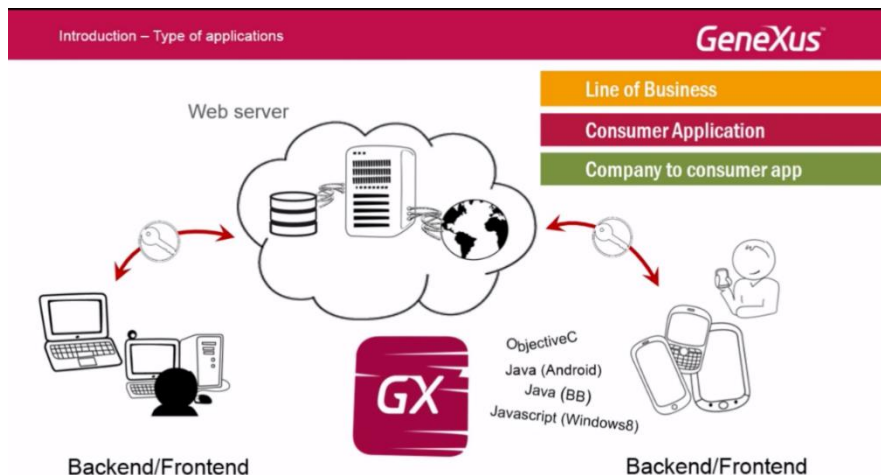


para o qual temos segurança, permitirá as operações de Crud (Create, update, delete)



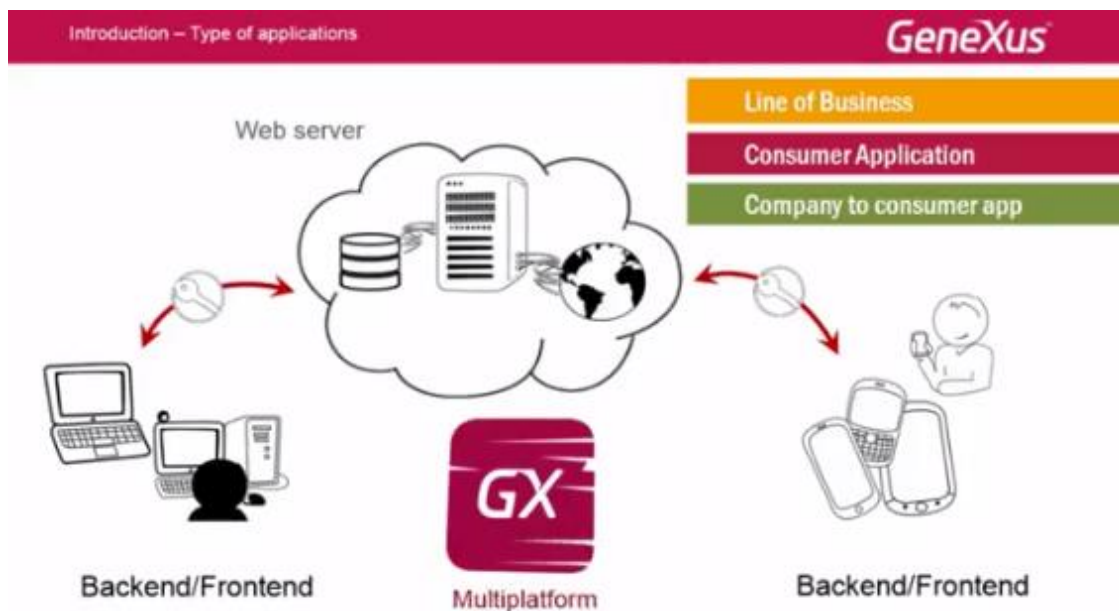
sobre alguns dados através de telas que para os usuários não autorizados não estão disponíveis.

Assim como a parte web desenvolvida no GeneXus, e este logo, implementado em algumas das línguas disponíveis; Ruby, Java ou .NET, o que significa que o GeneXus é múltiplaforma, a parte móvel de desenvolvimento analogamente

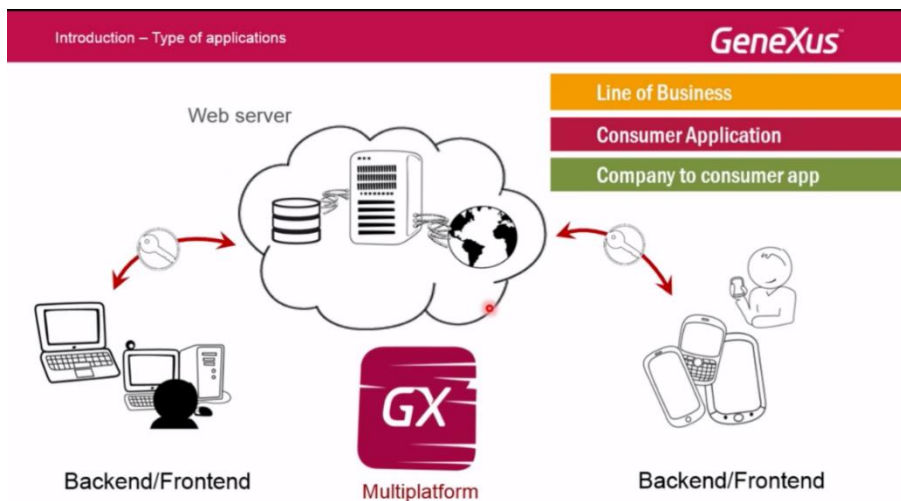


seguindo a lógica do GeneXus que logo será implementado na linguagem de programação da plataforma escolhida e seguindo os padrões de interface e comportamento dessa plataforma.

As plataformas suportadas são as mais variadas: IOS, Android, Blackberry e Window8. Portanto, o GeneXus sendo multiplataforma também continua desenvolvendo aplicações para dispositivos inteligentes.

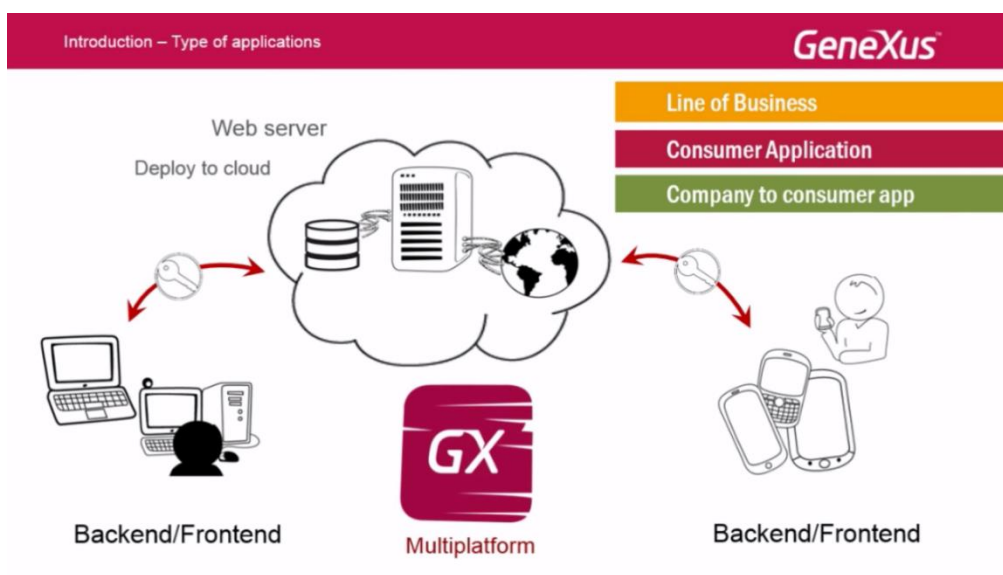


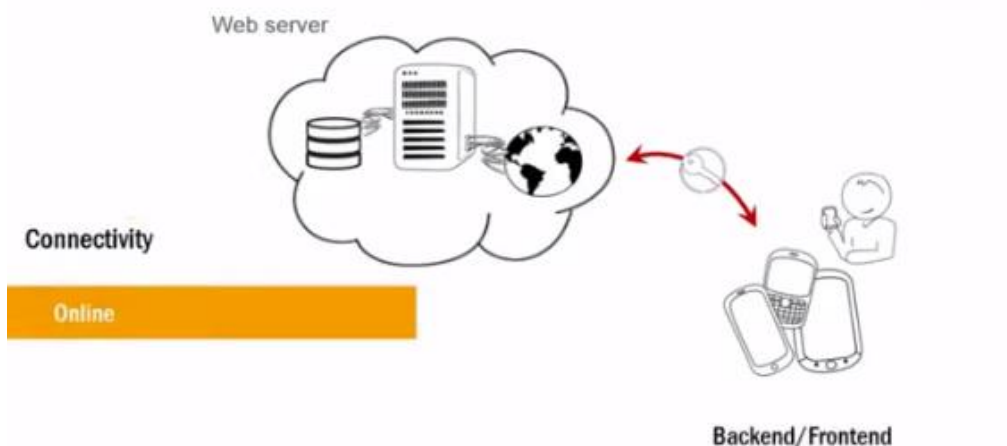
Para simplificarmos a tarefa de prototipagem, o GeneXus oferece a possibilidade de desenvolver a parte web na nuvem.



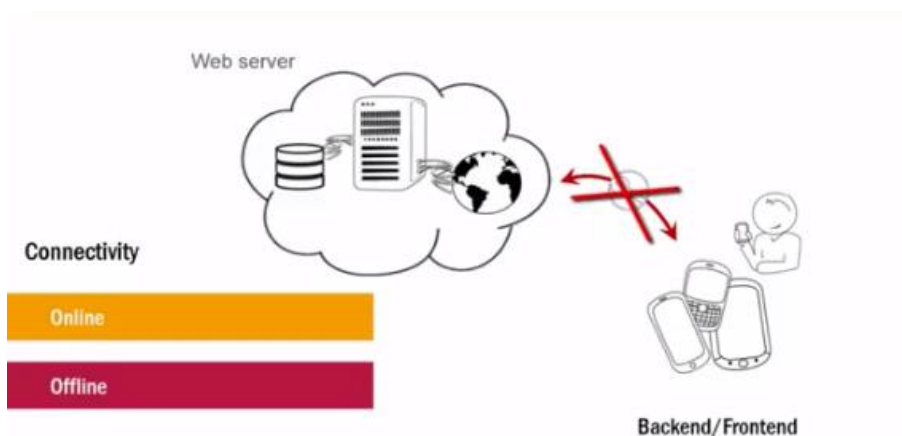
de modo que sempre que tenhamos acesso à internet poderíamos testar a aplicação de qualquer lugar e todo o software; programas e base de dados, estarão hospedados sem que tenhamos que nos preocupar com a infraestrutura.

Isso se faz através da propriedade de **Deploy to Cloud** para gerador web





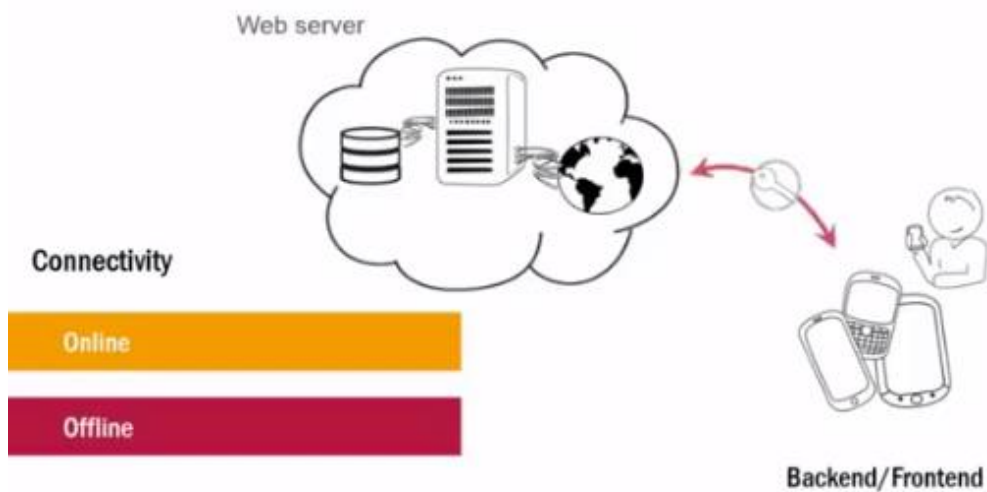
Um cenário bem importante para os dispositivos inteligentes, para alguns tipos de aplicações, é permitir que a aplicação ou parte dela esteja executando quando encontra desconectada da internet.



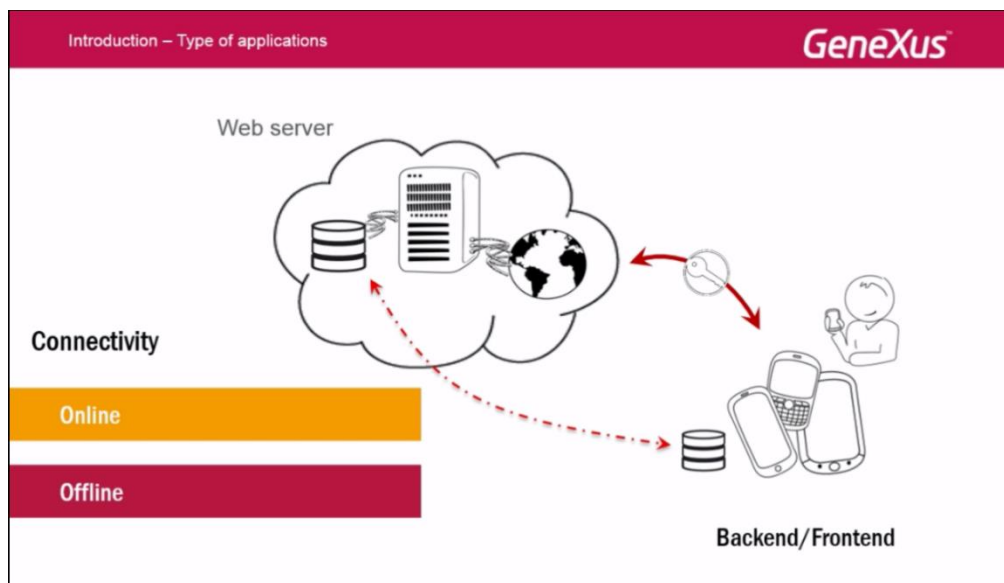
Pense por exemplo, quando se está trabalhando em zona rural ou quando um vendedor tem que fazer ordens de compra na rua, em diferentes locais.

No caso da nossa aplicação vamos querer que o usuário possa ver toda a agenda de conferência e toda a informação relacionada inclusive quando há perda de conexão.

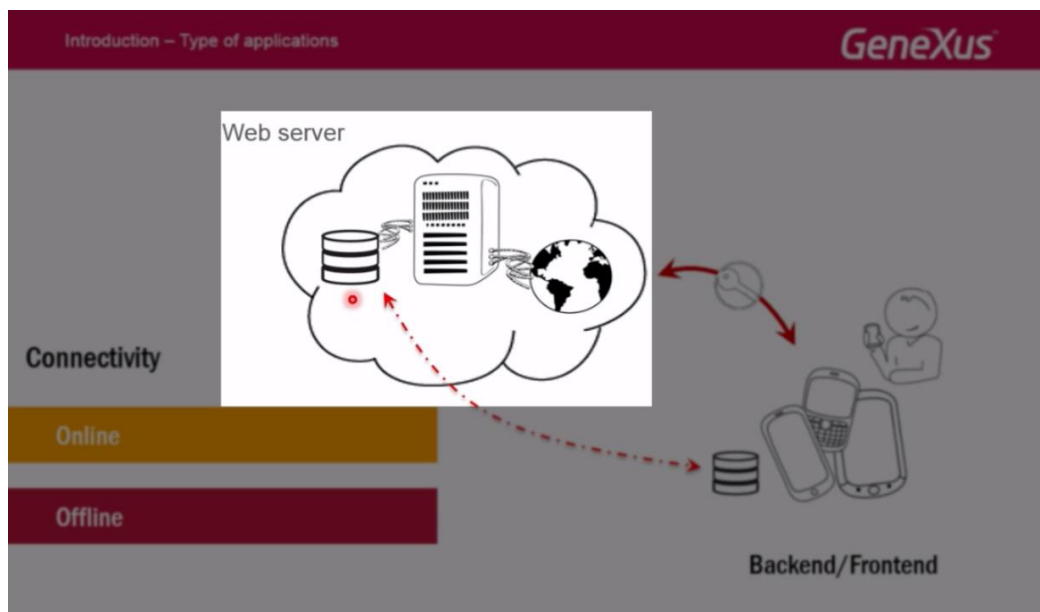
Logo quando está se reestabelece, automaticamente a aplicação atualiza seus dados,



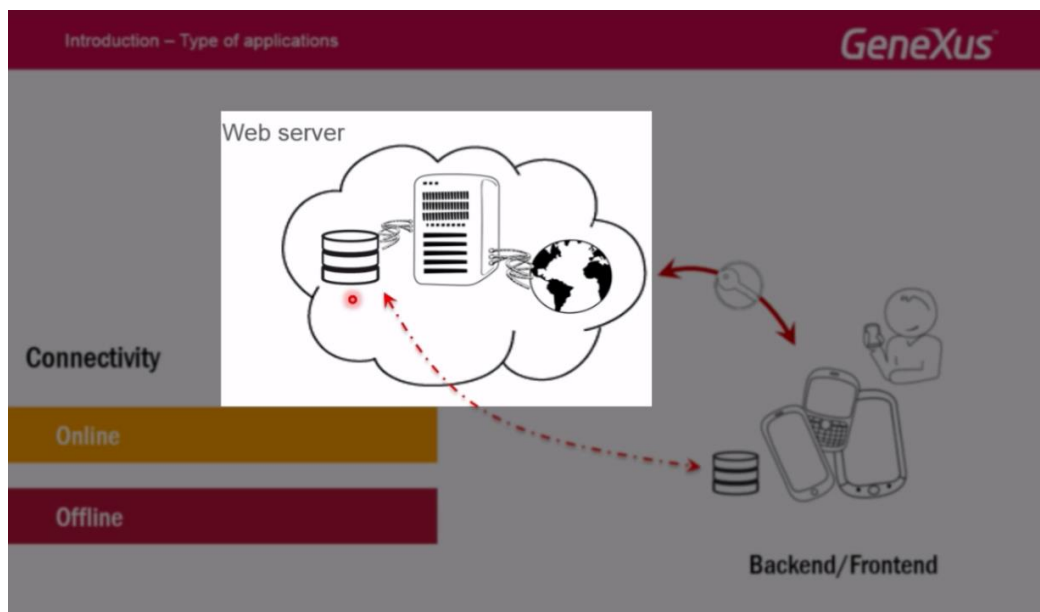
estará em uma base local no dispositivo sincronizado com o server, recebendo a informação destes e também enviando informação como por exemplo, quando o usuário marcar algumas conferências como favoritas; estando desconectado e essa informação queremos que envie ao servidor no momento de recuperar a conexão.



No entanto, será tarefa que requer necessariamente o acesso ao servidor web



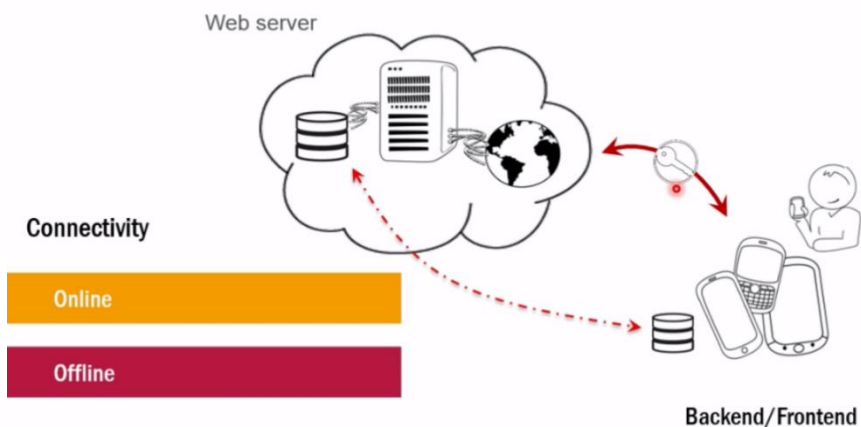
seja por sua sensibilidade, tarefas que devem ser validadas na base de dados centralizada...



como pelo rápido mudança de dados.

Estas tarefas deveriam executar **online**

Em nosso caso o login deverá ser com conexão



e o panel que mostra os tweets é desejável que também seja.

Portanto, podemos escolher quais objetos da aplicação podem ser executados off-line e quais objetos não.



O usuário vai exigir que toda aplicação para seu dispositivo tenha um Look & feel similar ao resto de suas aplicações nativas.

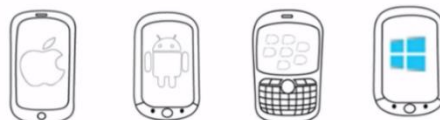
Web server



User experience

- Different guidelines (look & feel)

Native applications



Como por exemplo, sua aplicação de contato, calendário, a forma de fazer back no próprio dispositivo, ou seja, **que cumpra as guidelines de sua plataforma.**

Adiante vai exigir que seja integrado com as demais funcionalidades do dispositivo

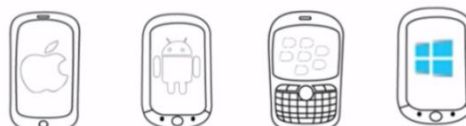
Web server



User experience

- Different guidelines (look & feel)

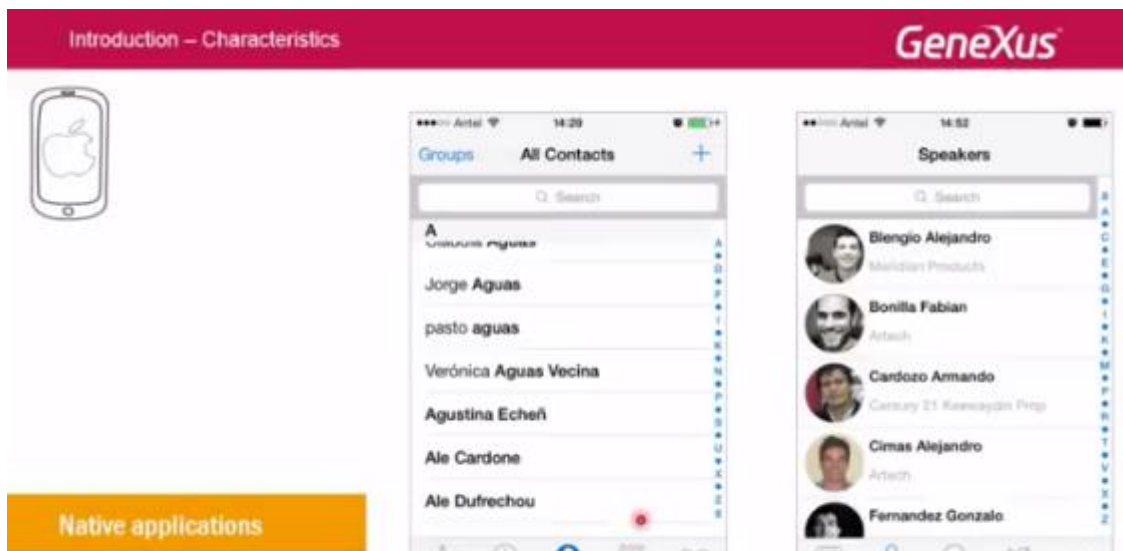
Native applications



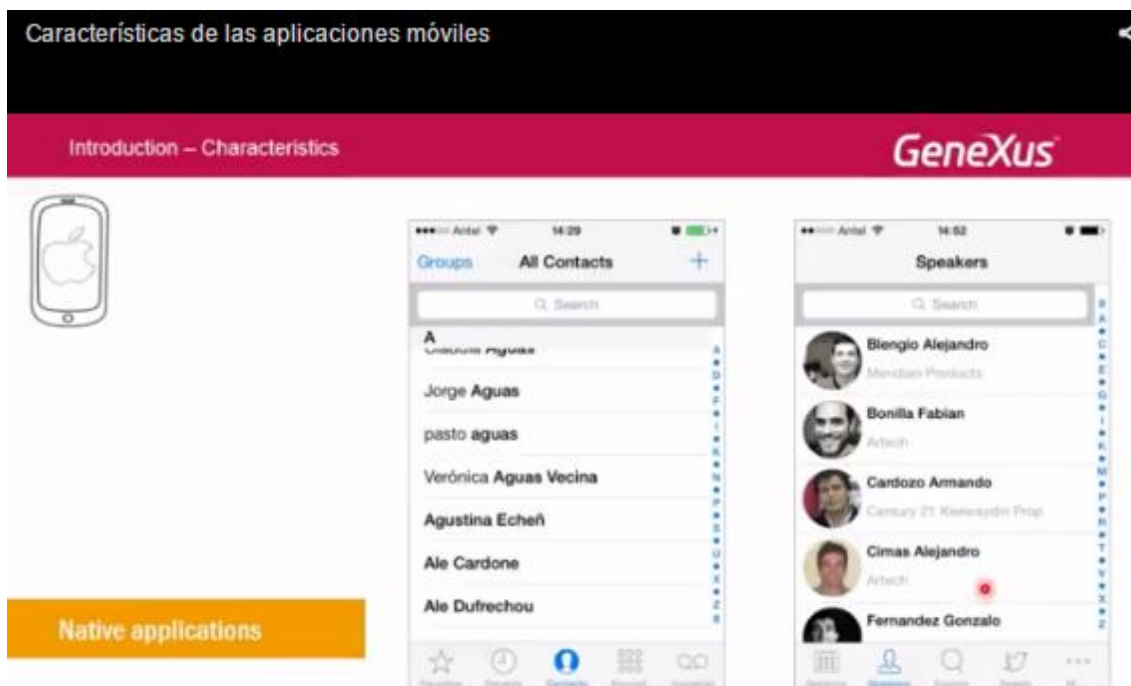
de software, como o calendário ou a agenda de contatos

e de hardware como a galeria de fotos, as chamadas telefônicas e o GPS.

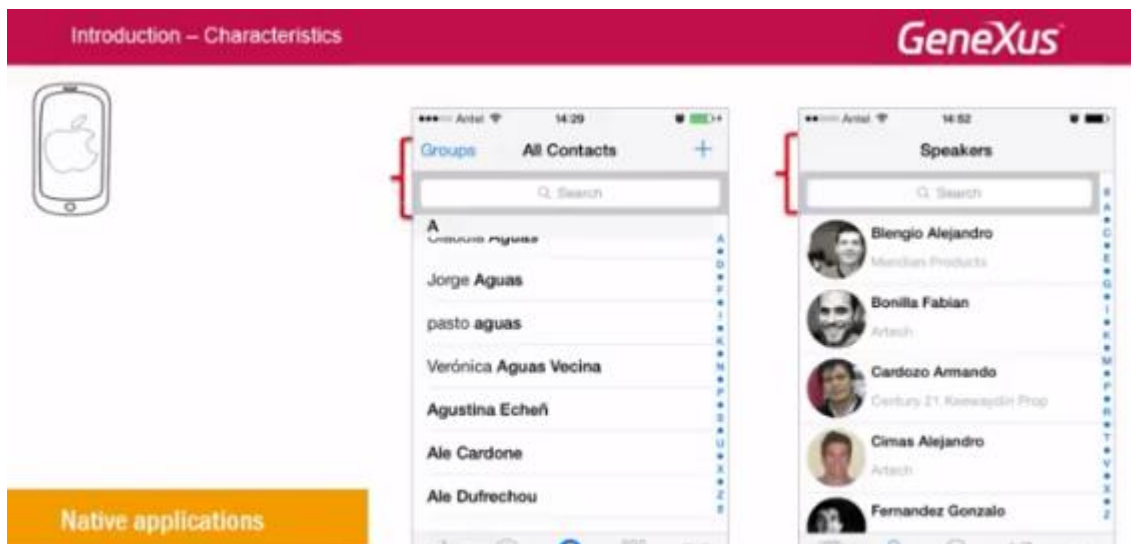
Aqui vemos como se vê por exemplo, para um iphone 7 a aplicação nativa dos contatos:



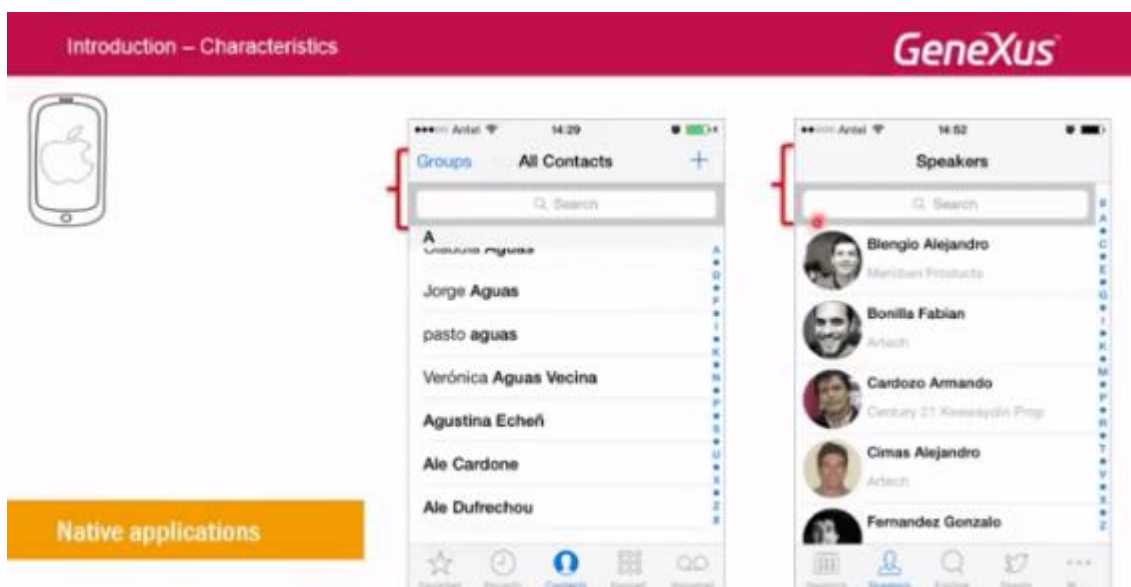
e como se vê pela aplicação desenvolvida com o GeneXus “EvenDay”.



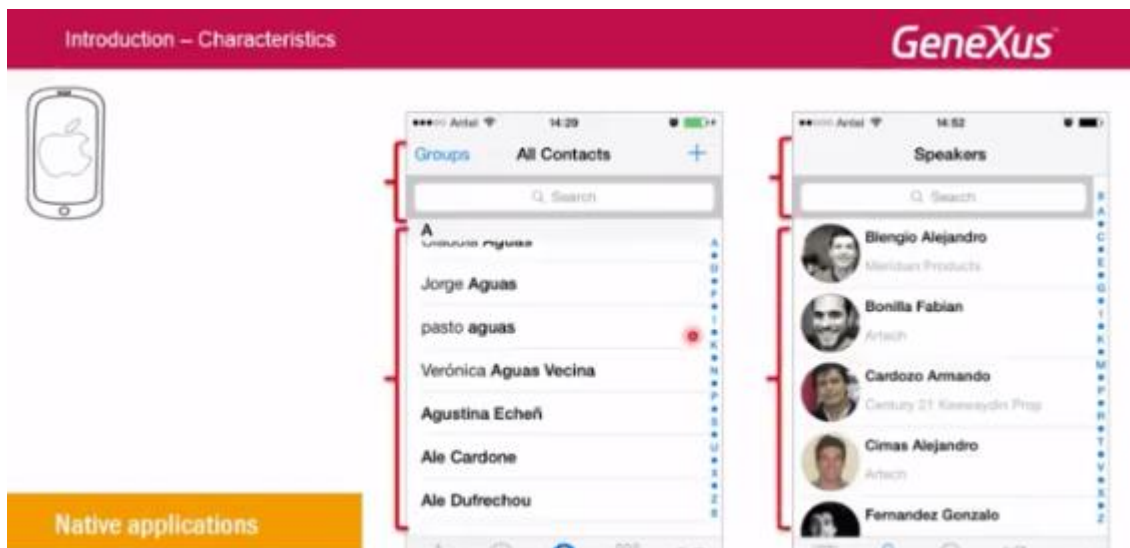
Independentemente das definições do desenho, cor e outros, observamos bem, que temos um determinado padrão comum de aparência. Uma barra superior com algumas ações



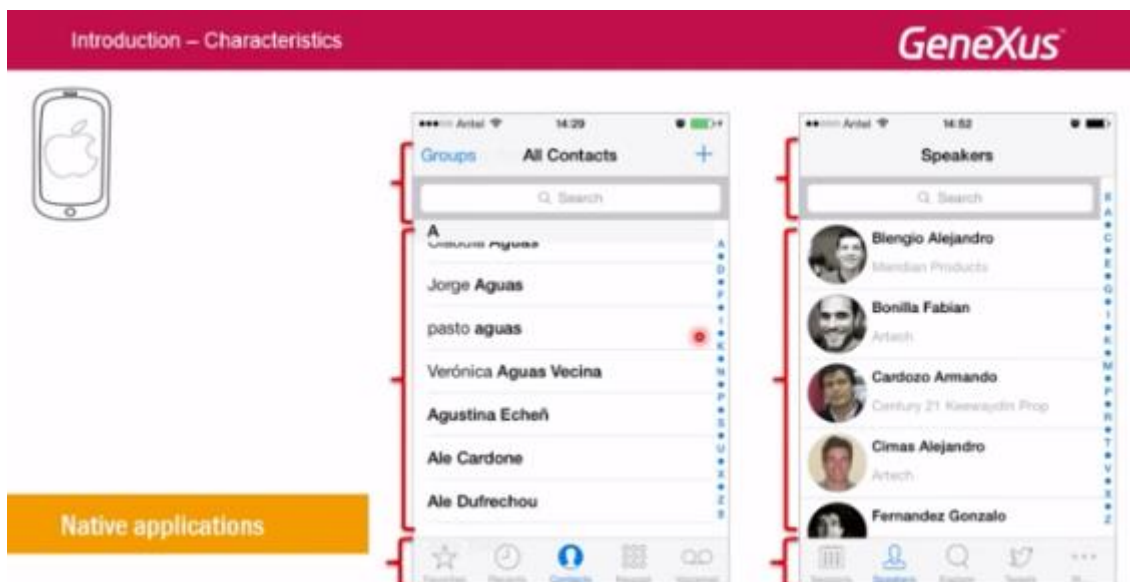
Uma opção de search



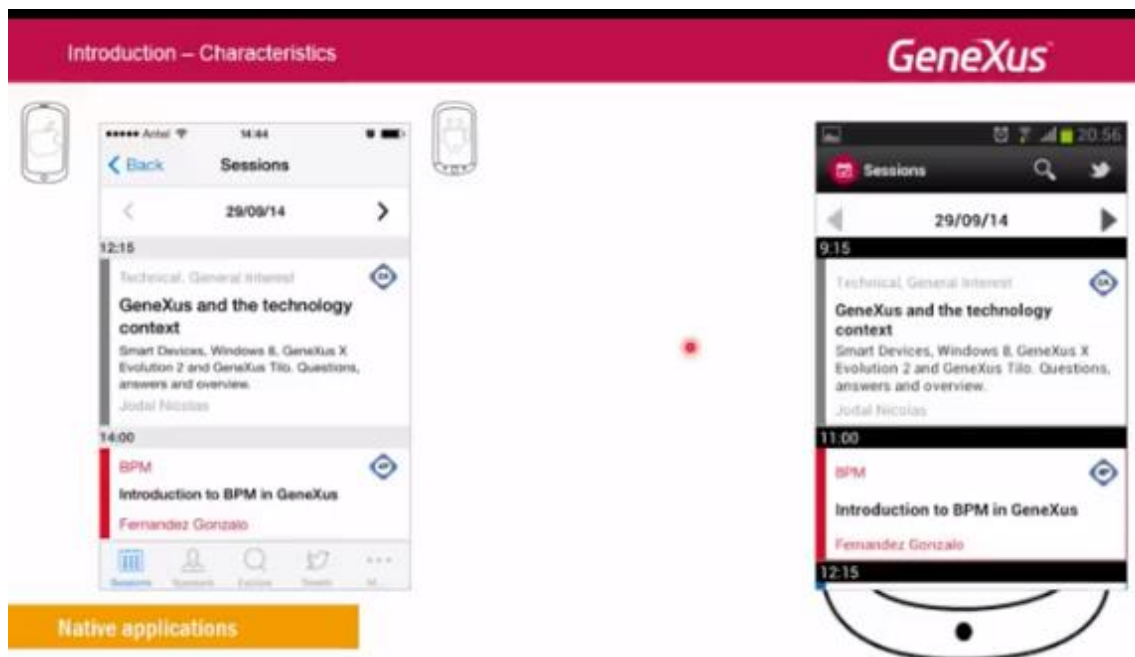
A informação no centro:



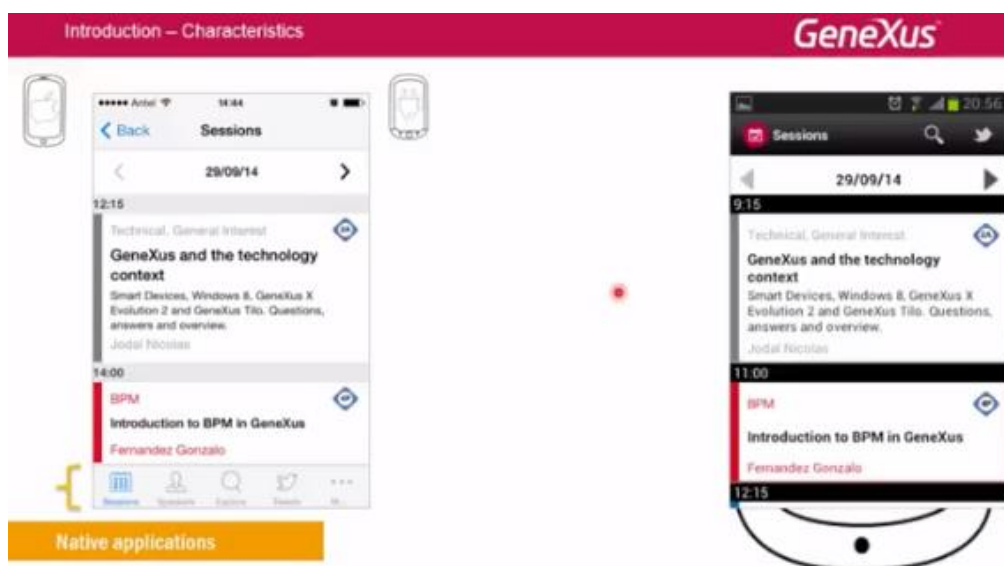
E logo um menu em forma de tabs



para mudar de opção ao visualizar.

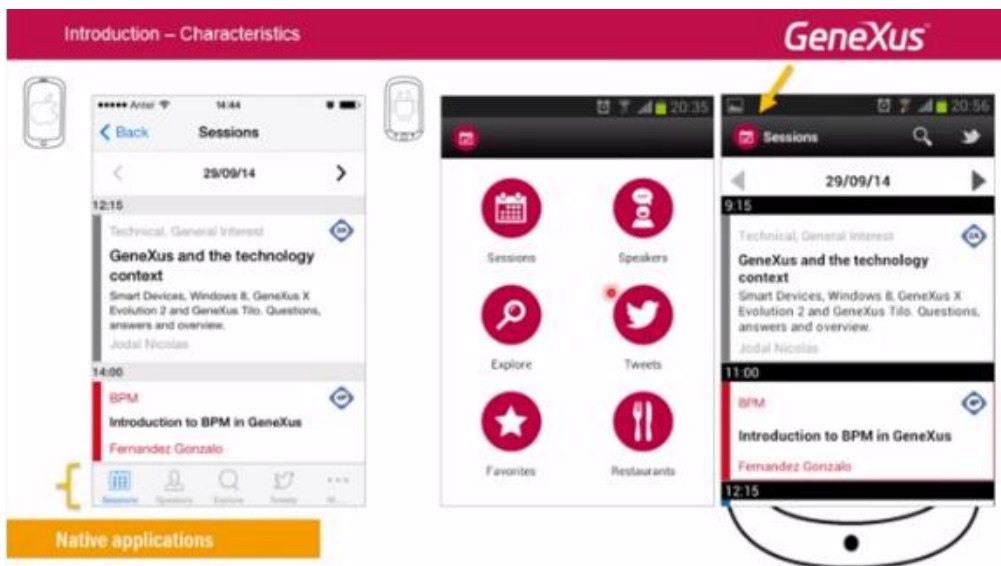


No entanto quando mudamos de dispositivo, se bem que a aplicação é essencialmente a mesma (mostra uma lista de conferências neste caso) tem sutis mas importantes diferenças estéticas e funcionais. Por exemplo...no caso de IOS tem um menu permanentemente visível :

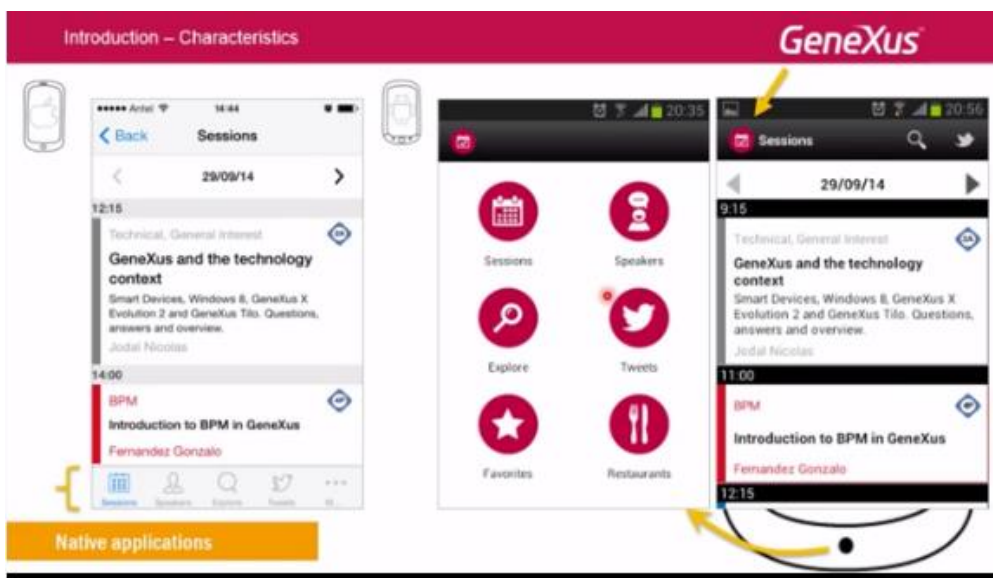


porque suas guidelines assim o indicam. Enquanto no Android não.

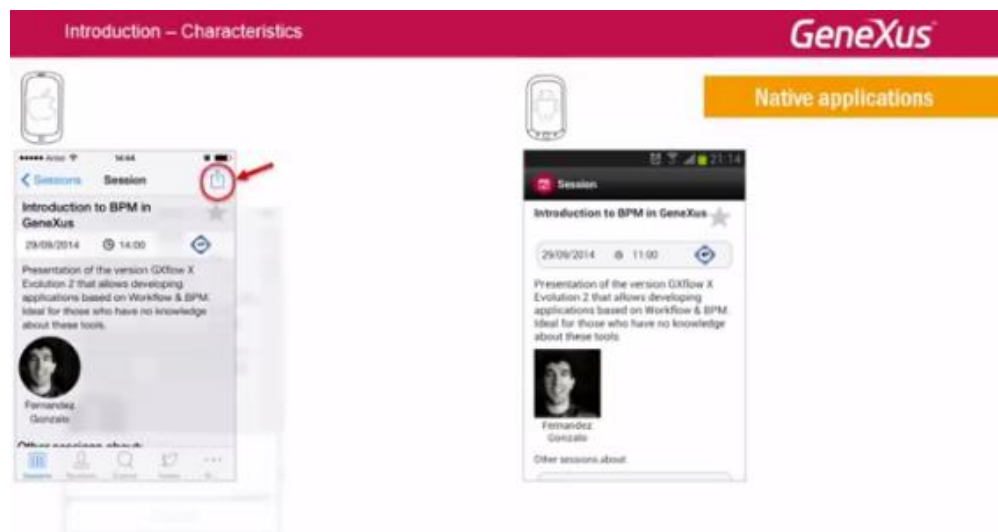
Para retornar ao menu dá um tap sobre o ícone da aplicação



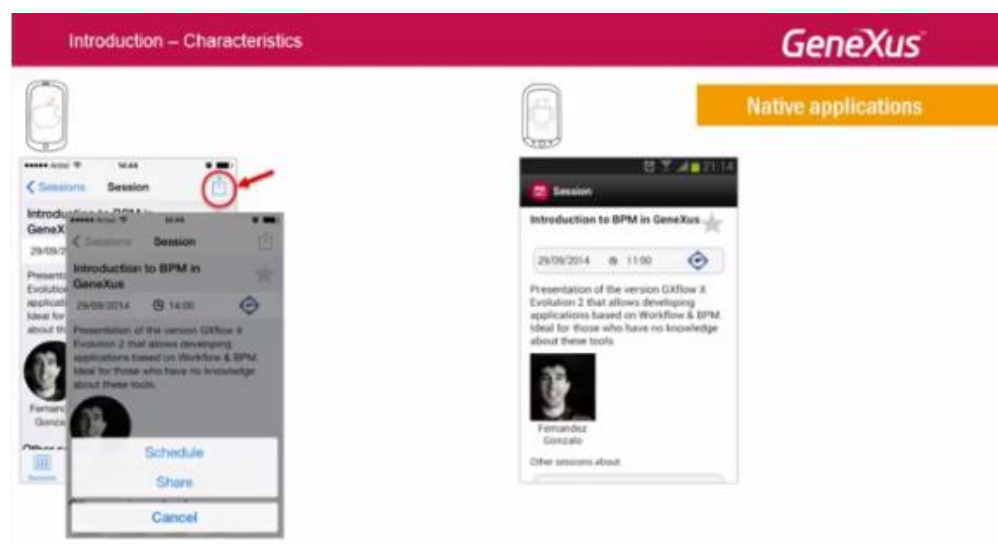
Ou seja, a tela que está visualizando foi diretamente invocada desde o menu com o botão back do próprio dispositivo retorna.



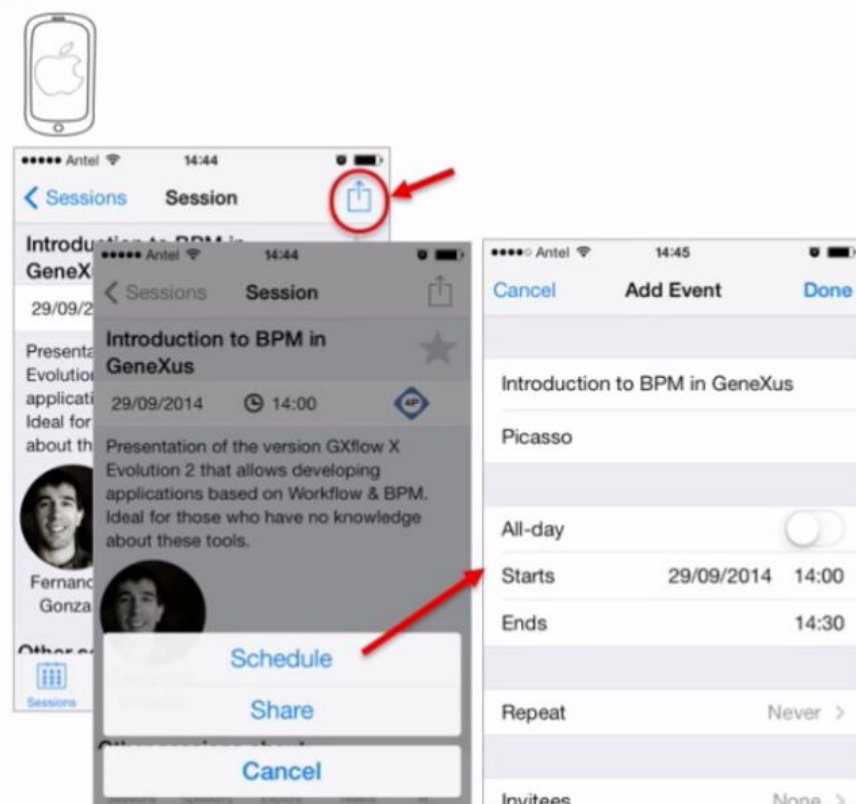
Aqui podemos ver como a aplicação para iphone uma vez posicionado sobre uma conferência permite-nos através de um ícone especial



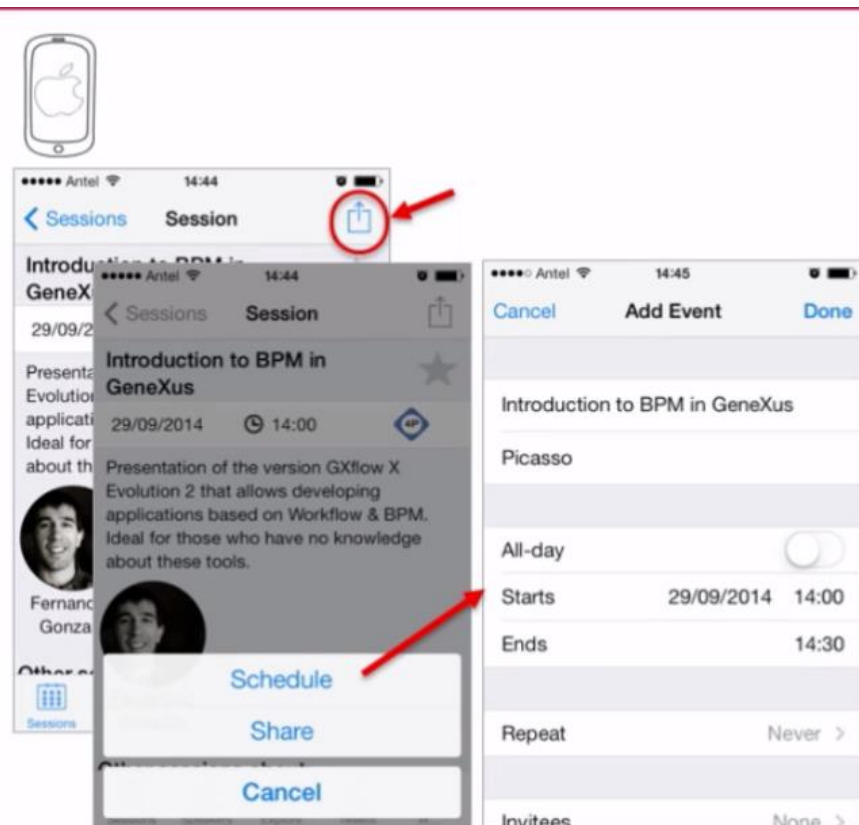
ver as ações que podemos realizar sobre essa conferência.



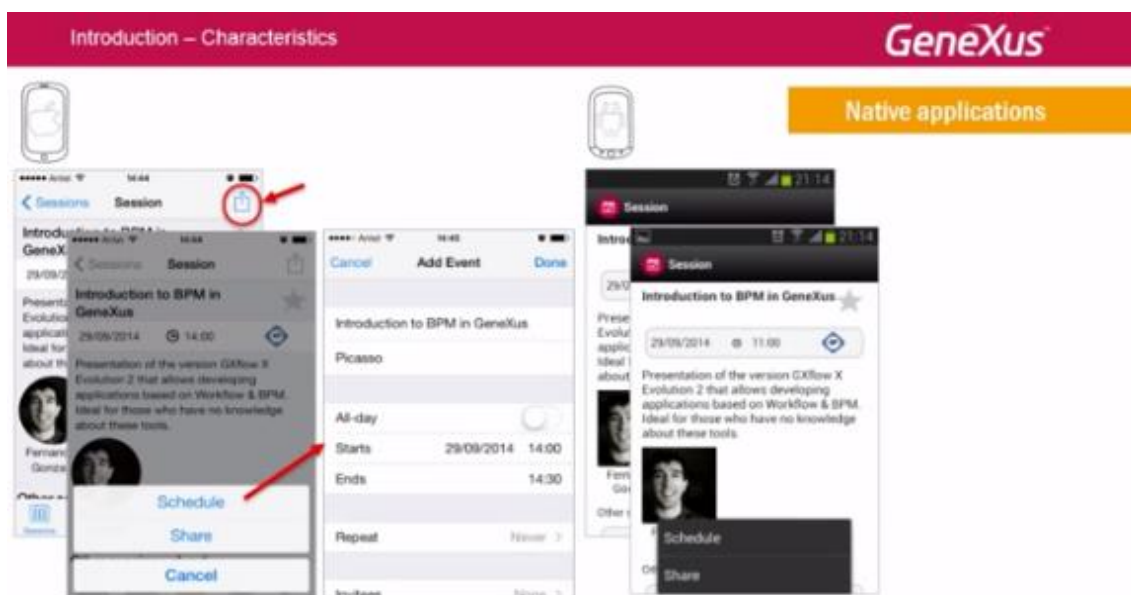
como agendar no calendário do dispositivo



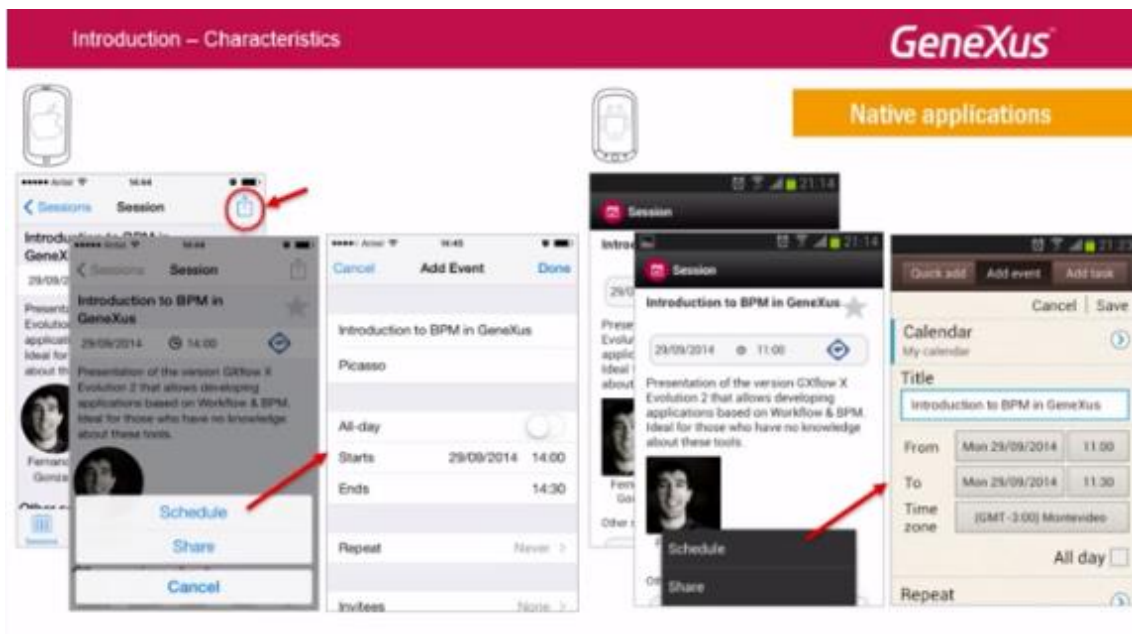
como agendar no calendário do dispositivo o que irá abrir no programa nativo do dispositivo ou compartilhá-lo.



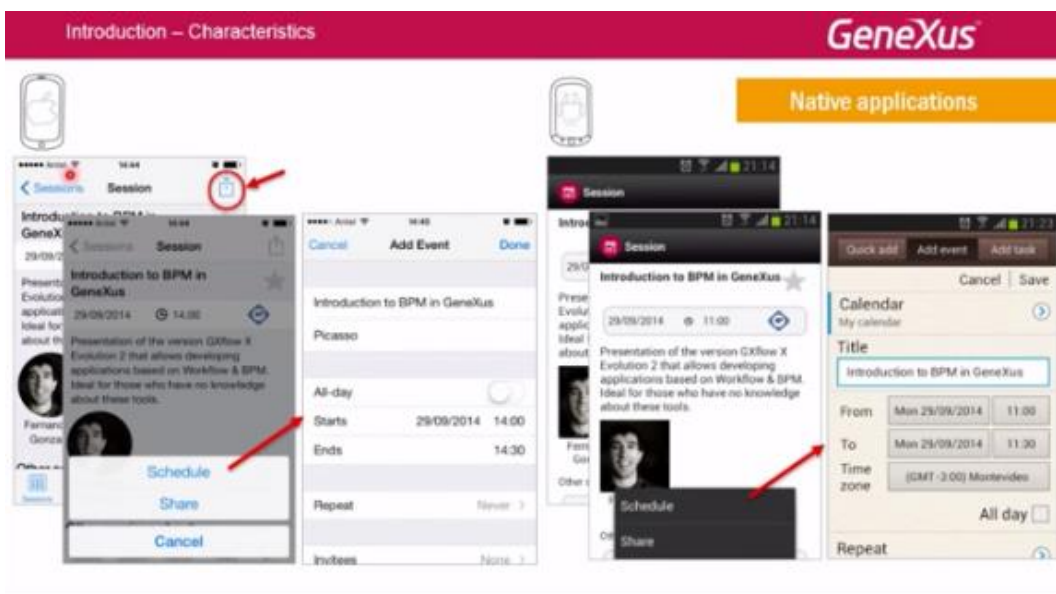
Na mudança do telefone Android permite-nos acessar as ações através de um botão especial no próprio dispositivo.



Da mesma maneira quando escolhemos a ação Schedule irá abrir o calendário do dispositivo Android.



Assim mesmo vemos como no iOS se incorpora um ícone na barra superior para poder retornar o chamador



Ao contrario no Android há o botão back do próprio dispositivo

Em resumo, **é um requisito para as aplicações para Smart Devices que são nativas**

No vídeo seguinte introduziremos no modelo conceitual dessas aplicações.

The Conceptual Model of mobile applications for Smart Devices

Veremos os quatro tipos de telas que necessitam desenvolver uma aplicação para Smart Devices, e os objetos que as implementam, e suas relações.

Convidamos a vê-lo.

