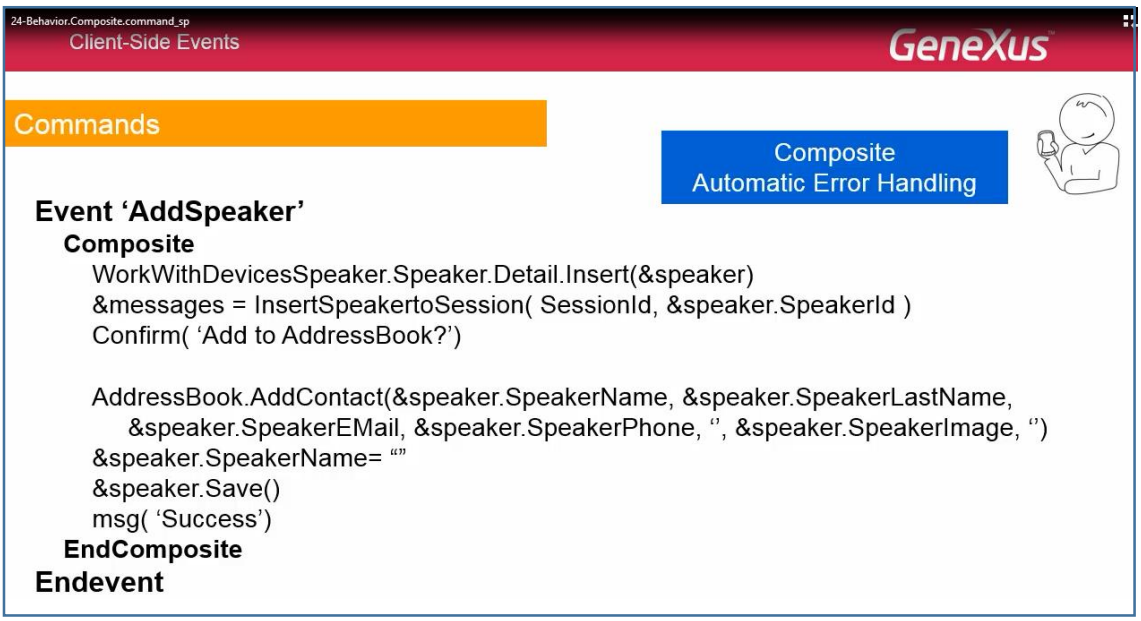
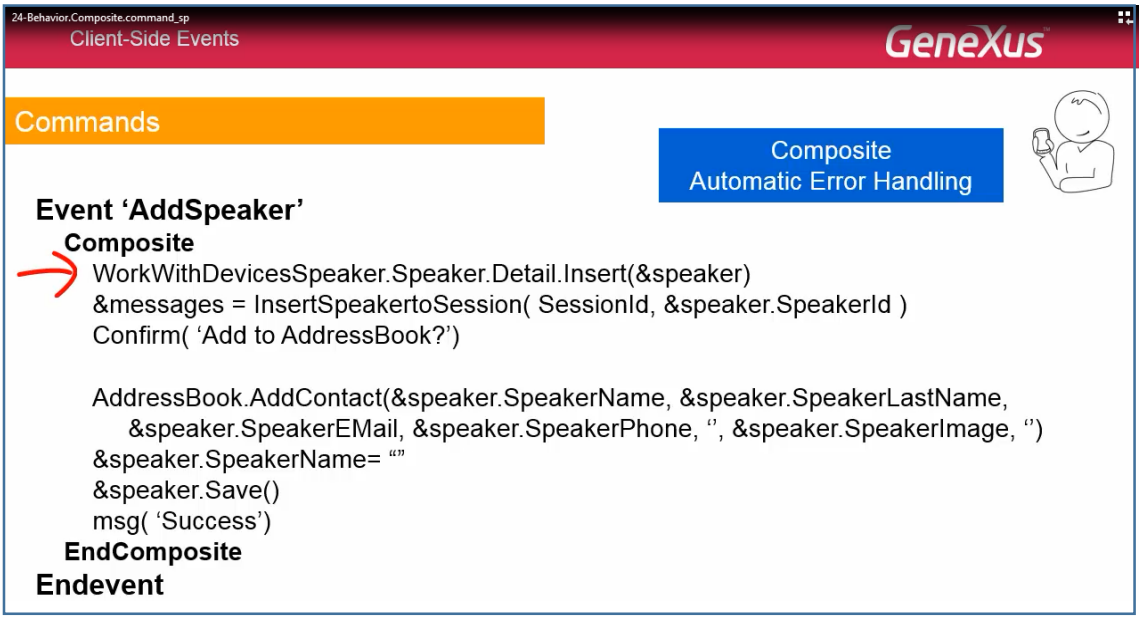
Behavior: Composite Command

Estudiaremos en detalle el comando Composite, específico de aplicaciones para Smart Devices.

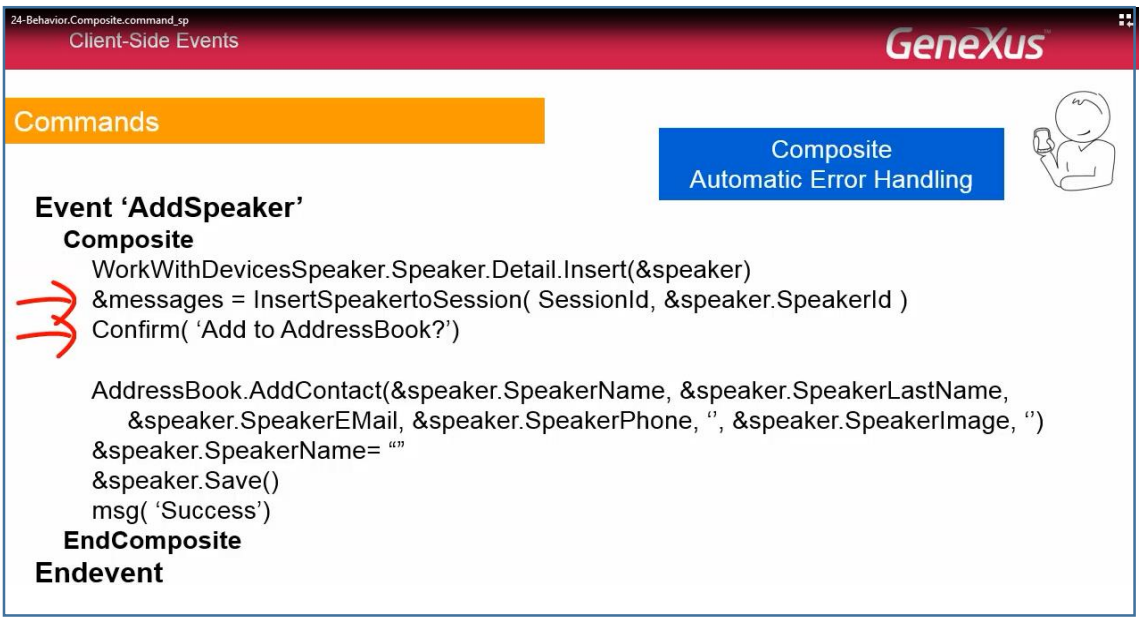
Lo veremos a través de un ejemplo, donde codificamos un evento del cliente, que requerirá utilizar este comando.



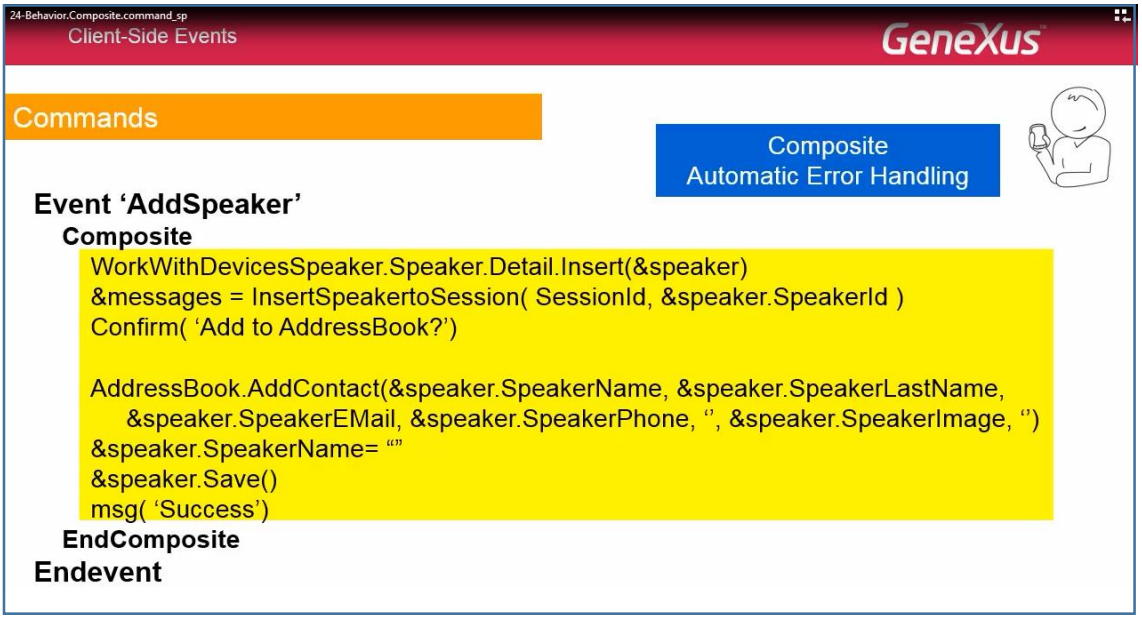
Ya habíamos dicho que toda vez que dentro de un evento del cliente, en una aplicación móvil, se requieren dos o más invocaciones del tipo que sean.. a objetos con interfaz:



a procedimientos o Apis:

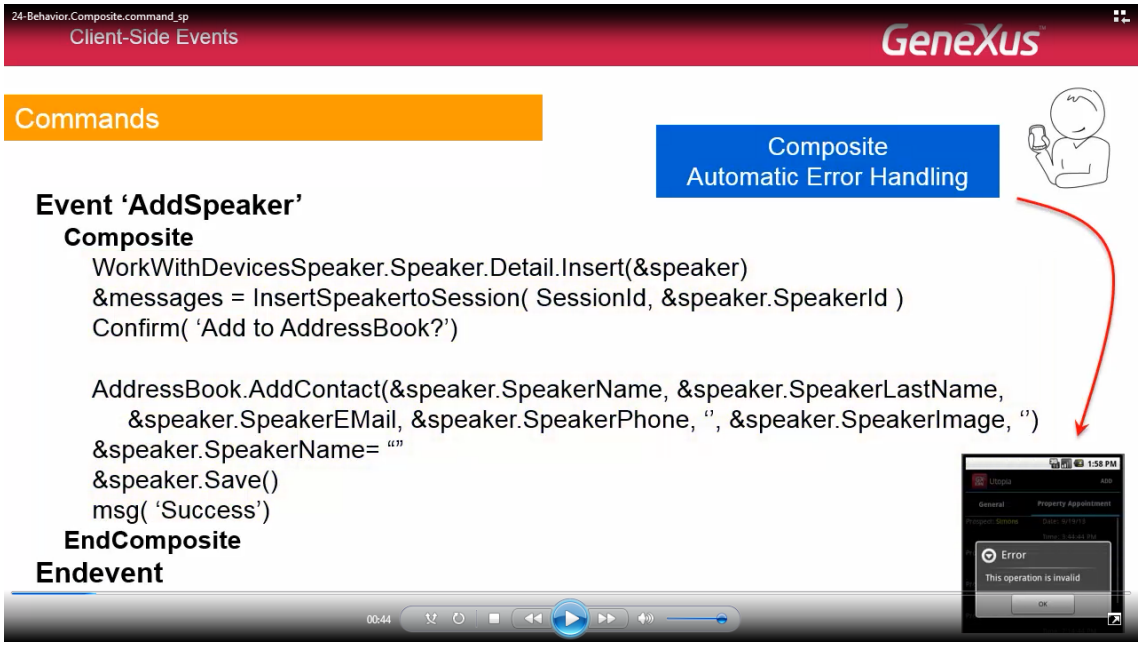


es necesario colocar el código completo del evento



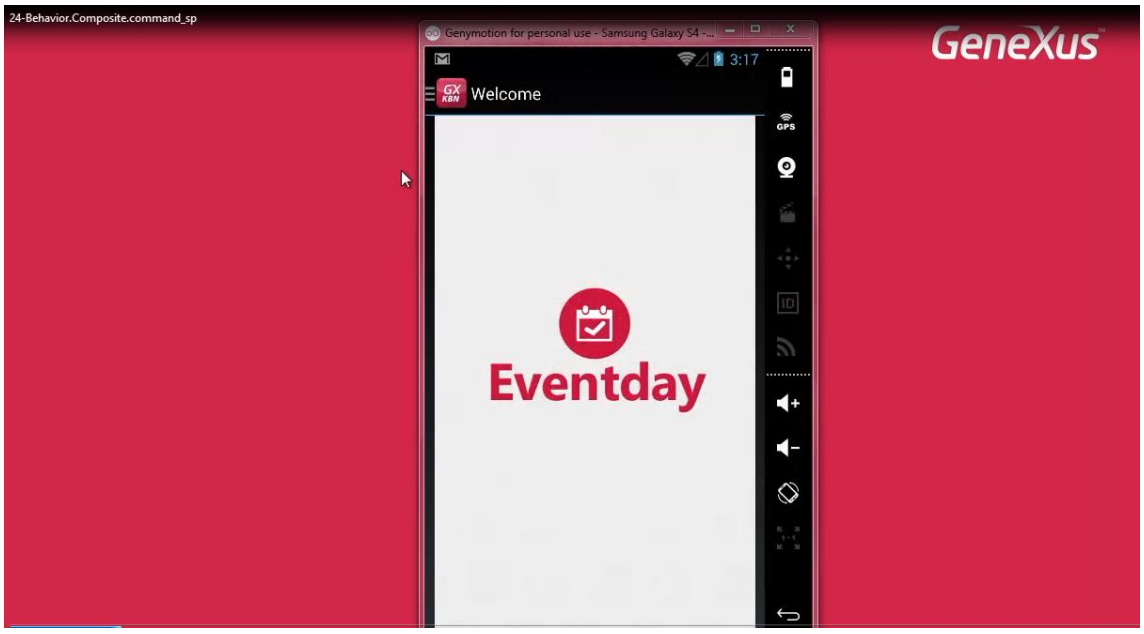
dentro del comando: Composite

La razón es que este comando brindará al código que contiene, el manejo de errores automático.

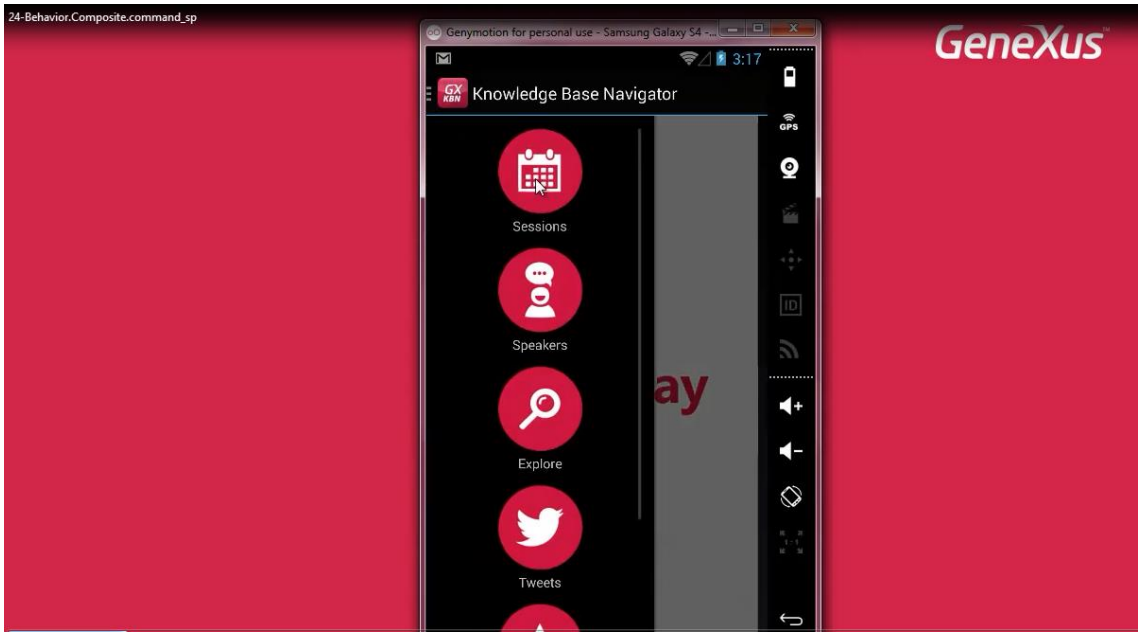


La ejecución será interrumpida ante un error, y se desplegarán los mensajes automáticamente al usuario, sin que el desarrollador deba hacer nada.

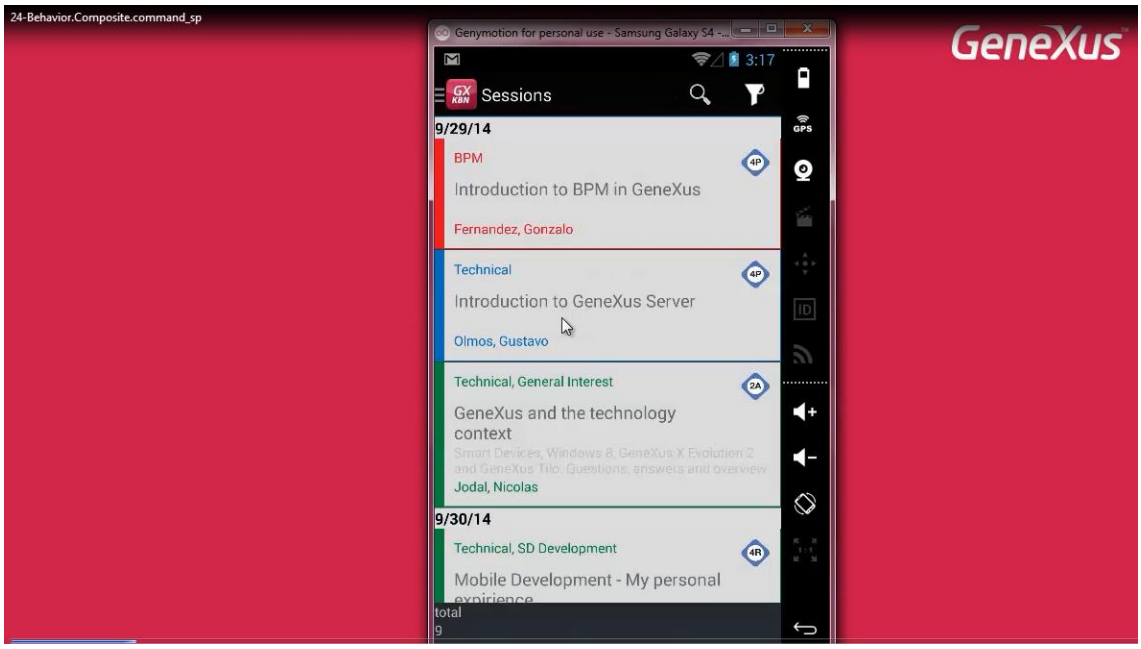
Comencemos con el ejemplo.



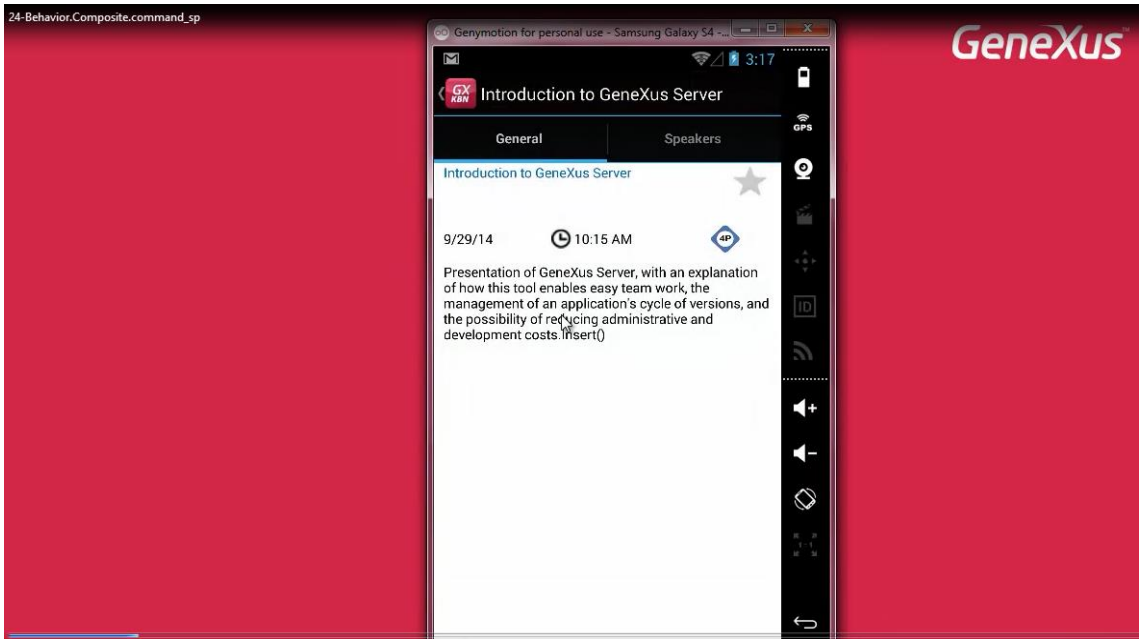
Observemos que si vamos al List de Sessions



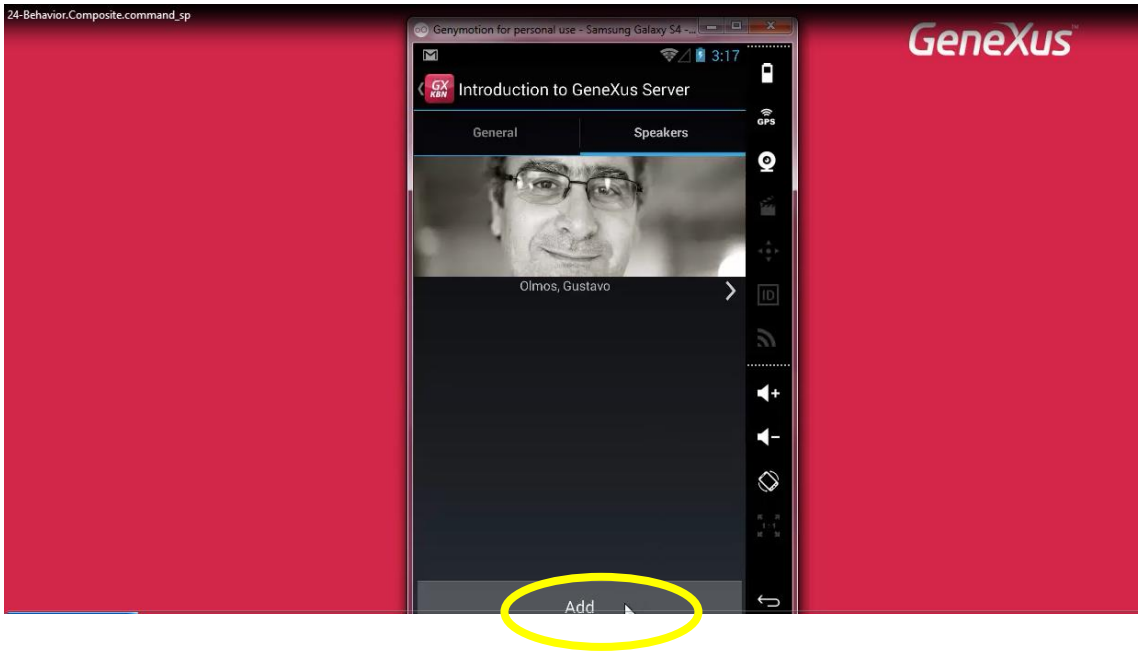
y elegimos una Session



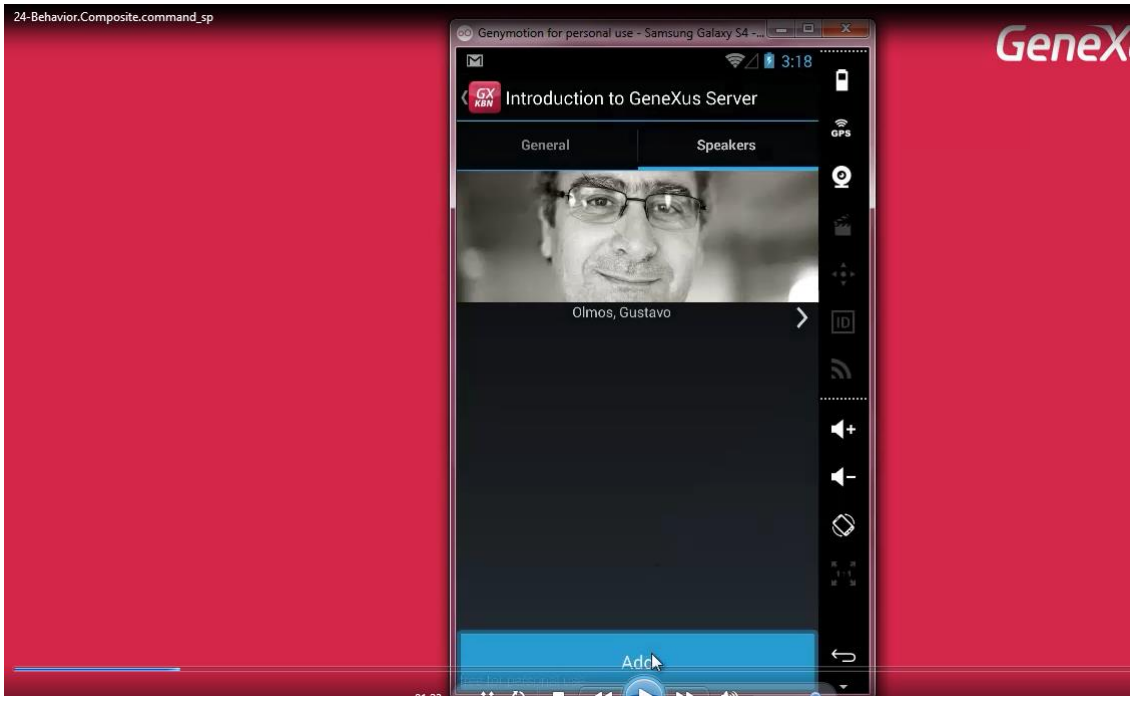
para ver su información

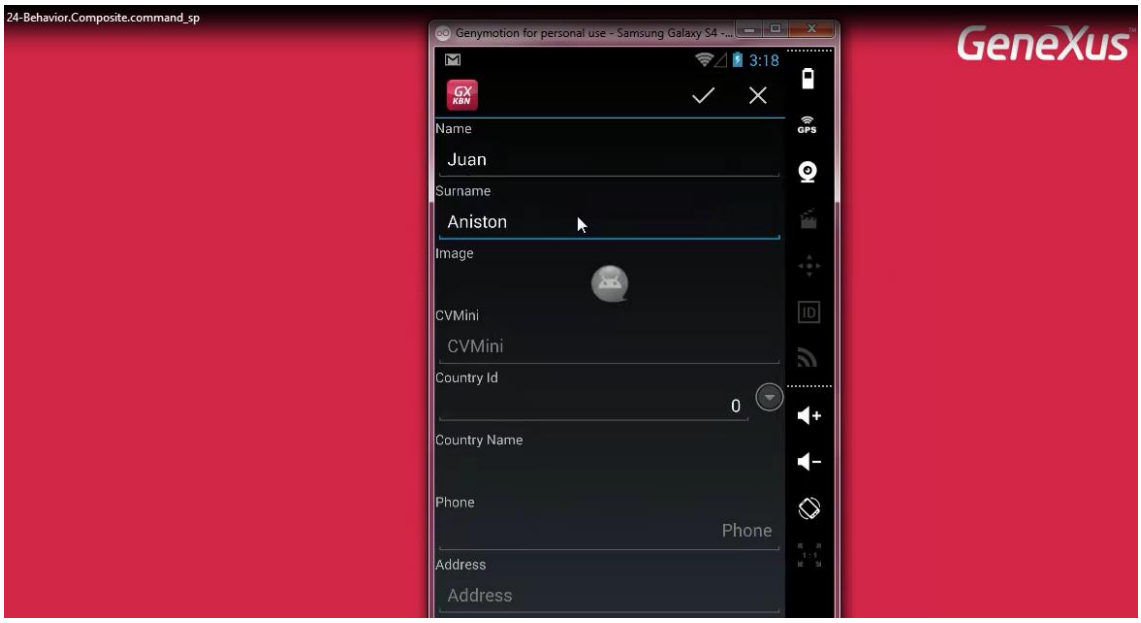


en el TAB de Speakers, hemos agregado un botón Add:

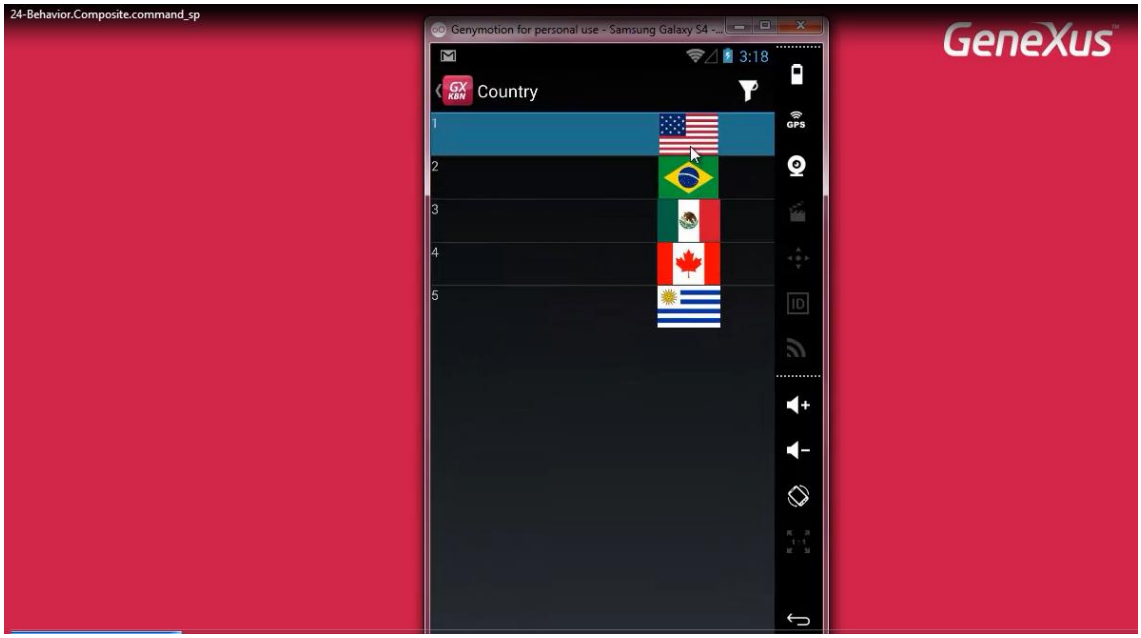


Este botón Add, nos permitirá insertar un nuevo orador en el sistema, asociarlo a esta conferencia… y además, agregarlo a la libreta de direcciones del dispositivo. Veámoslo.

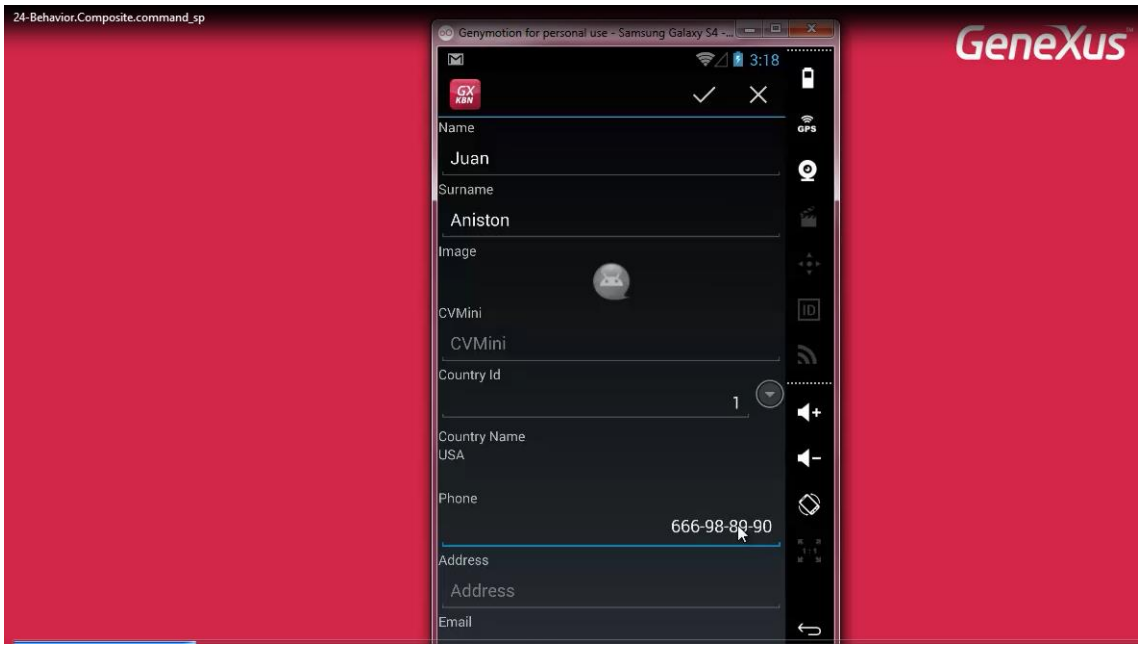


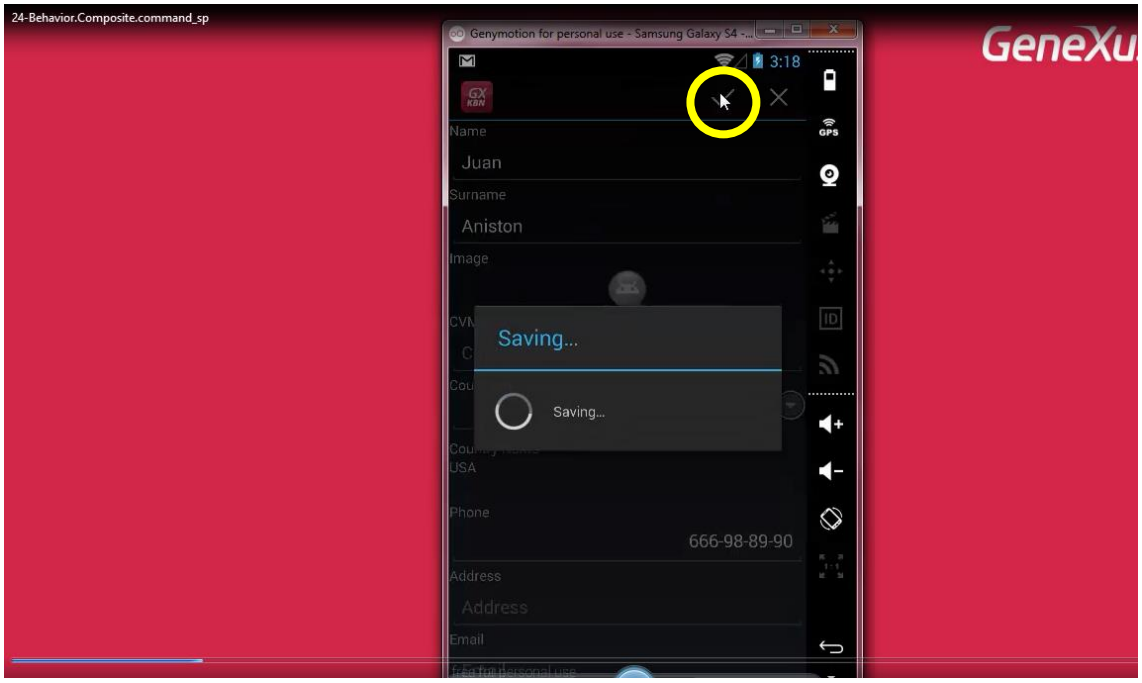


Vamos a asignarle simplemente un país..

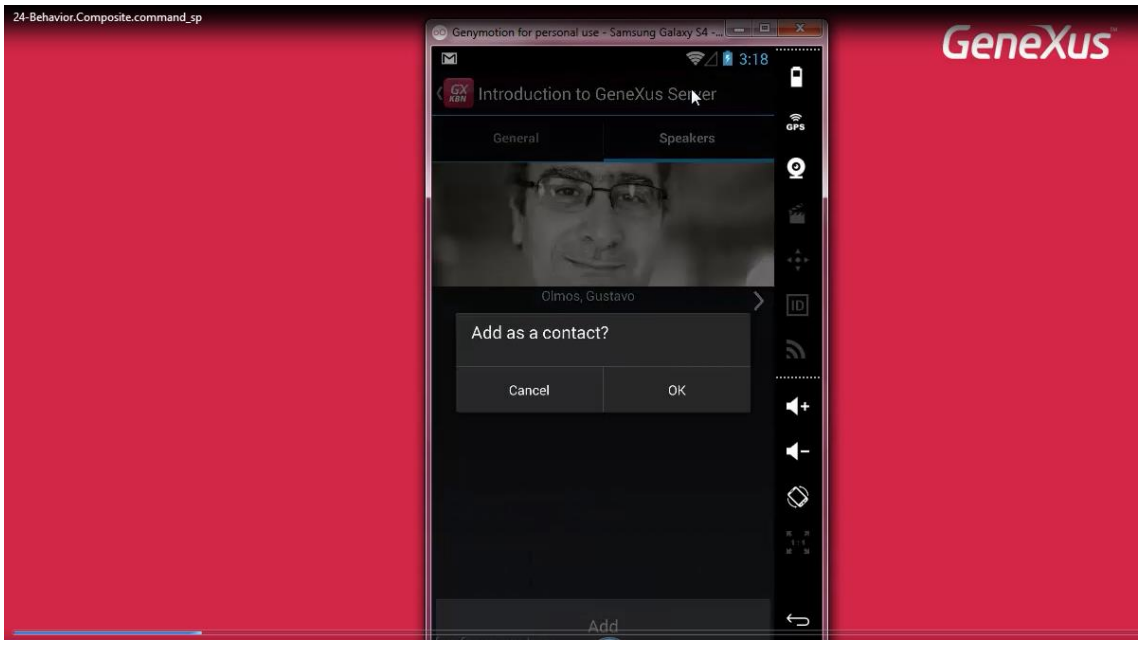


y un teléfono

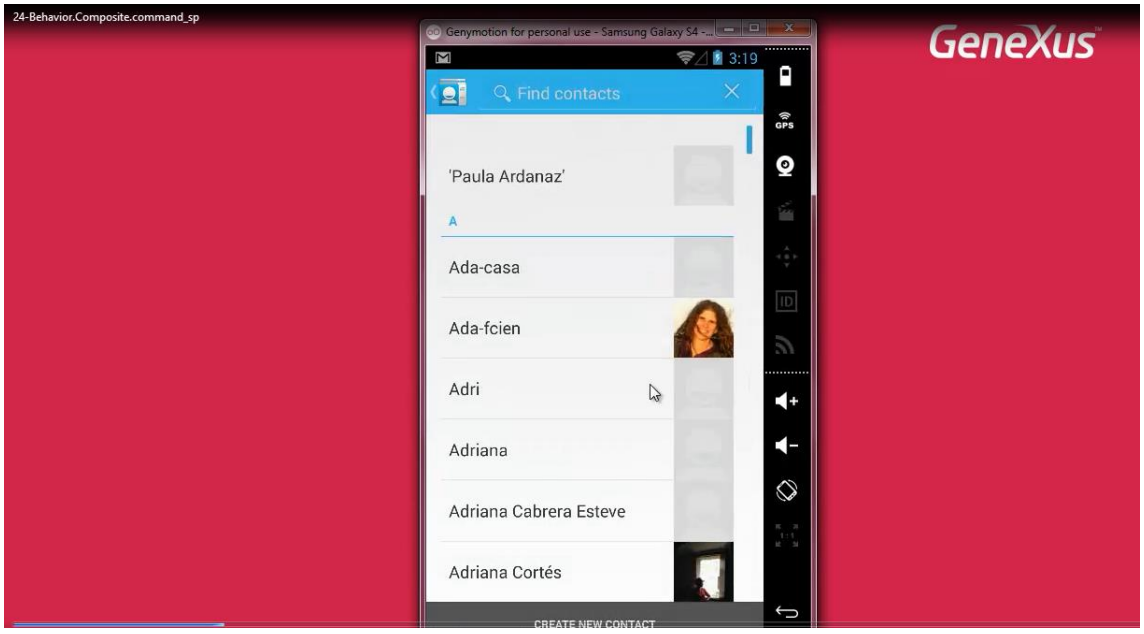




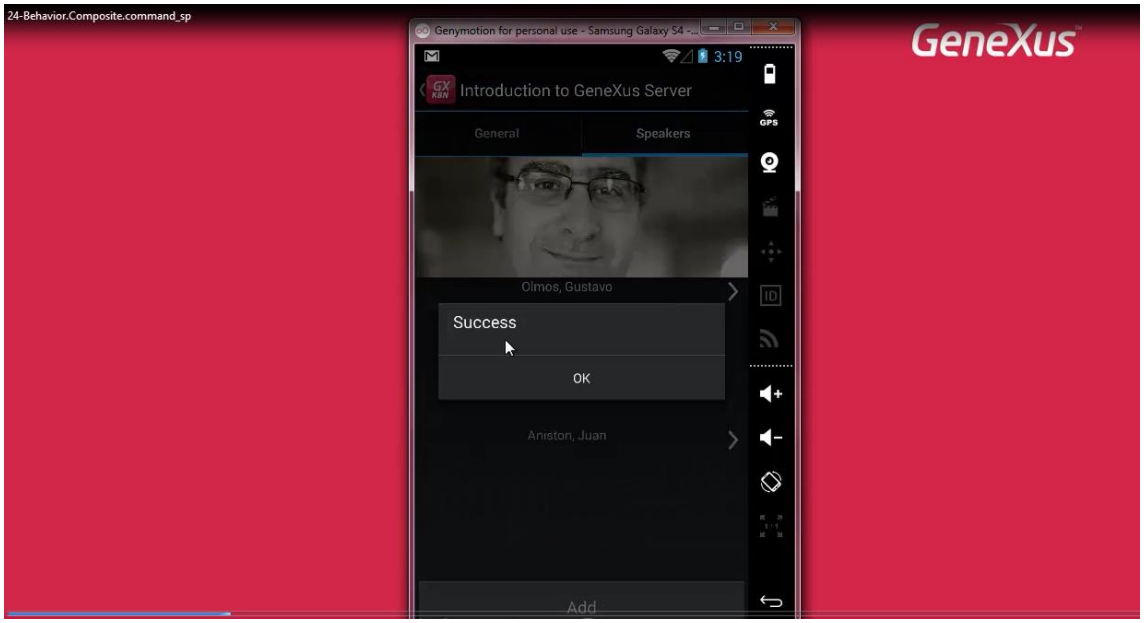
Grabamos… y en ese acto hizo 2 cosas: insertó el speaker en la base de datos… y además asoció ese speaker a la conferencia… y nos está preguntando si queremos agregarlo como contacto:



Decimos que sí… y está abriendo la libreta de direcciones del dispositivo



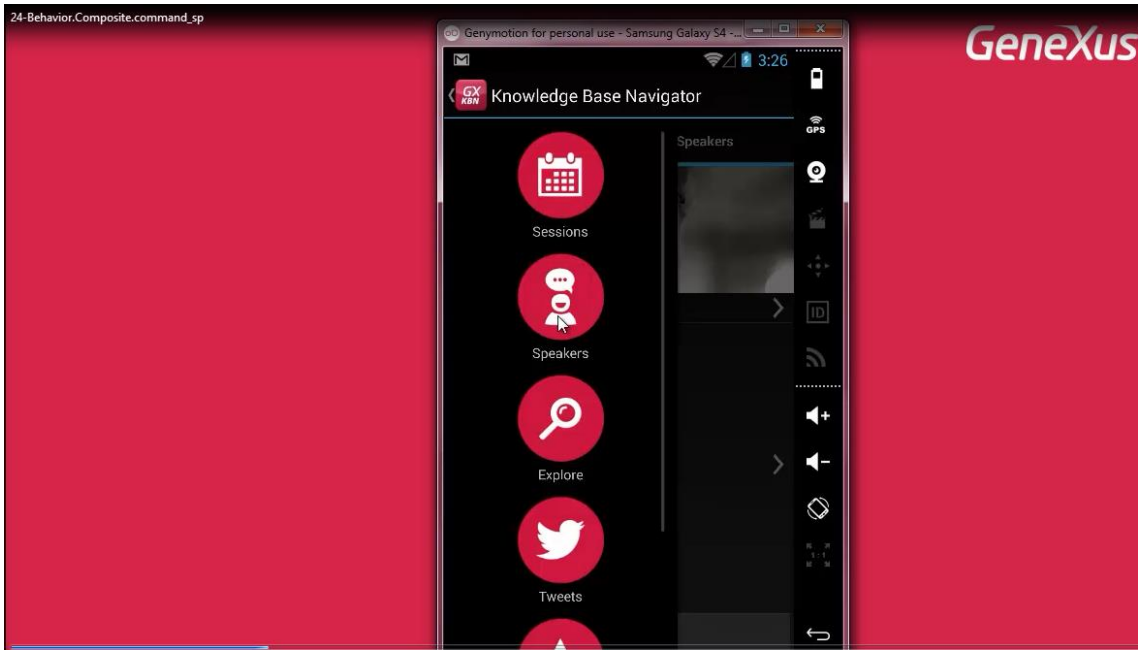
para crear un nuevo contacto con la información que le pasamos



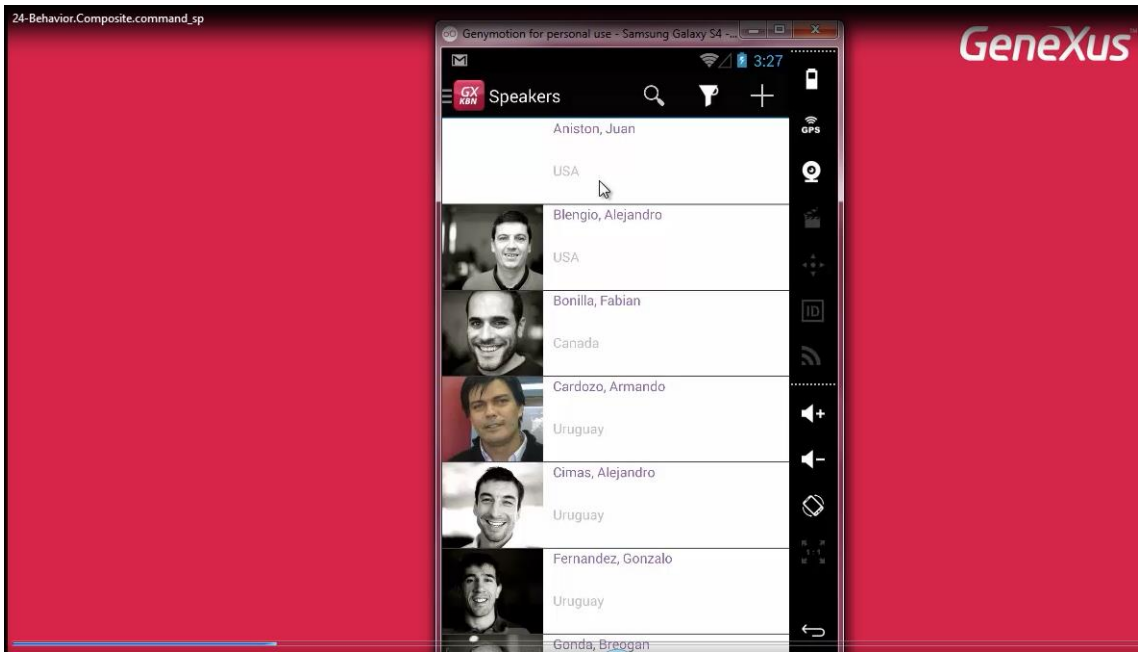
Nos está desplegando este mensaje: Success
Damos: OK
Y vemos cómo quedó agregado.

No está saliendo la foto puesto que no la hemos ingresado.

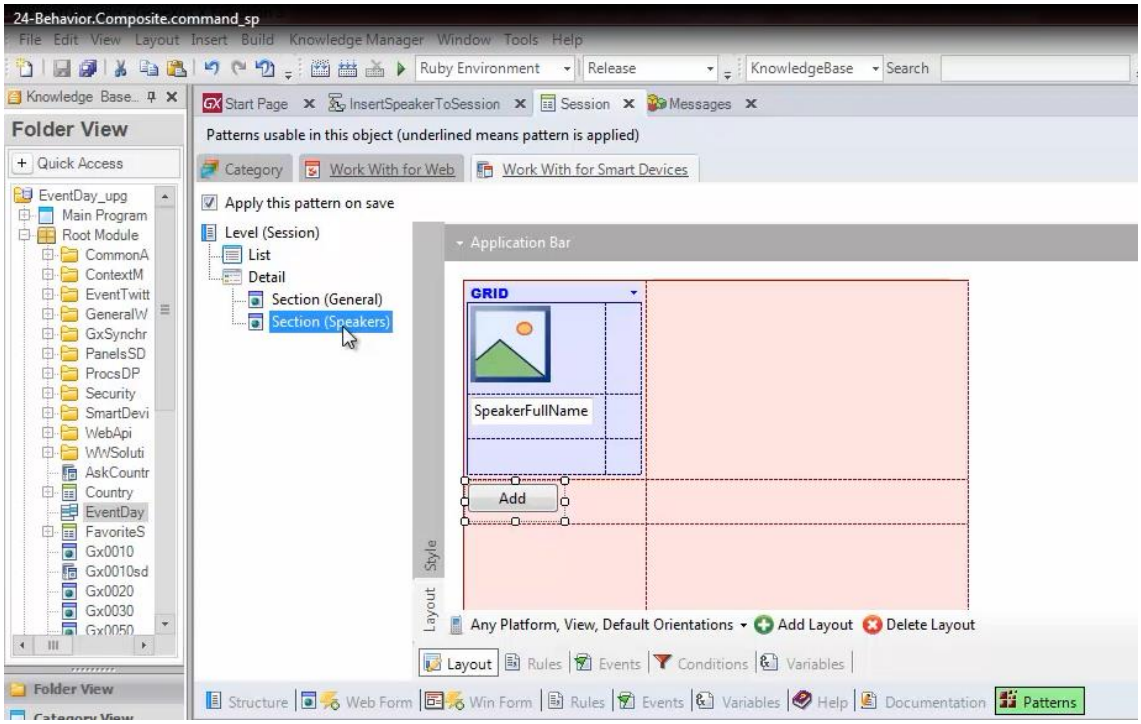
Si ahora vamos al List de Speakers



tenemos que encontrar a ese orador listado… y aquí está:

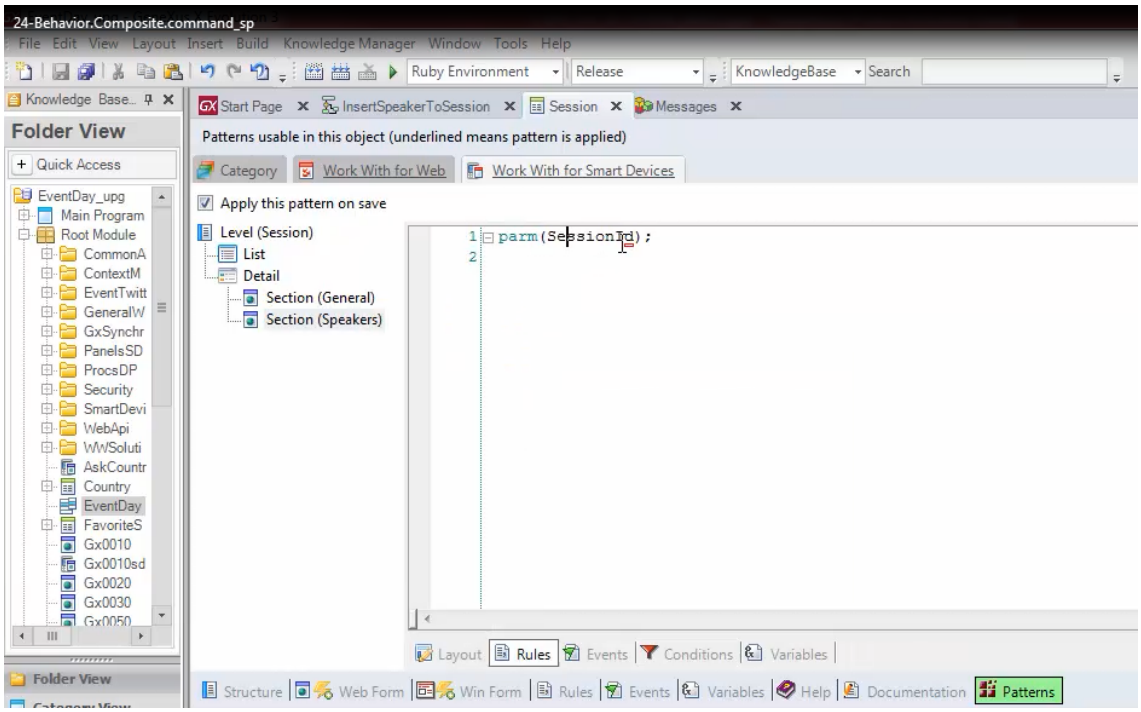


Vayamos a GeneXus, a ver cómo se implementó este evento.



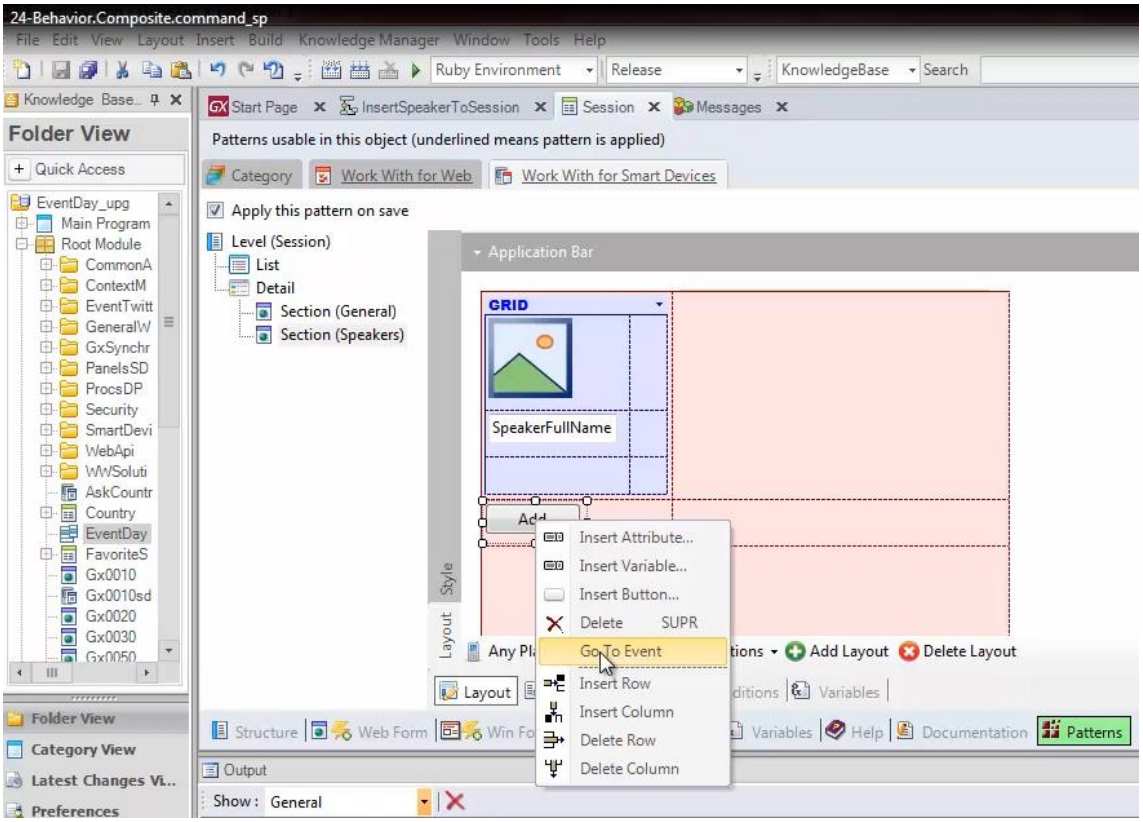
Estamos en la Section Speakers del Work With de Sessions.

Por tanto, si vamos a las rules

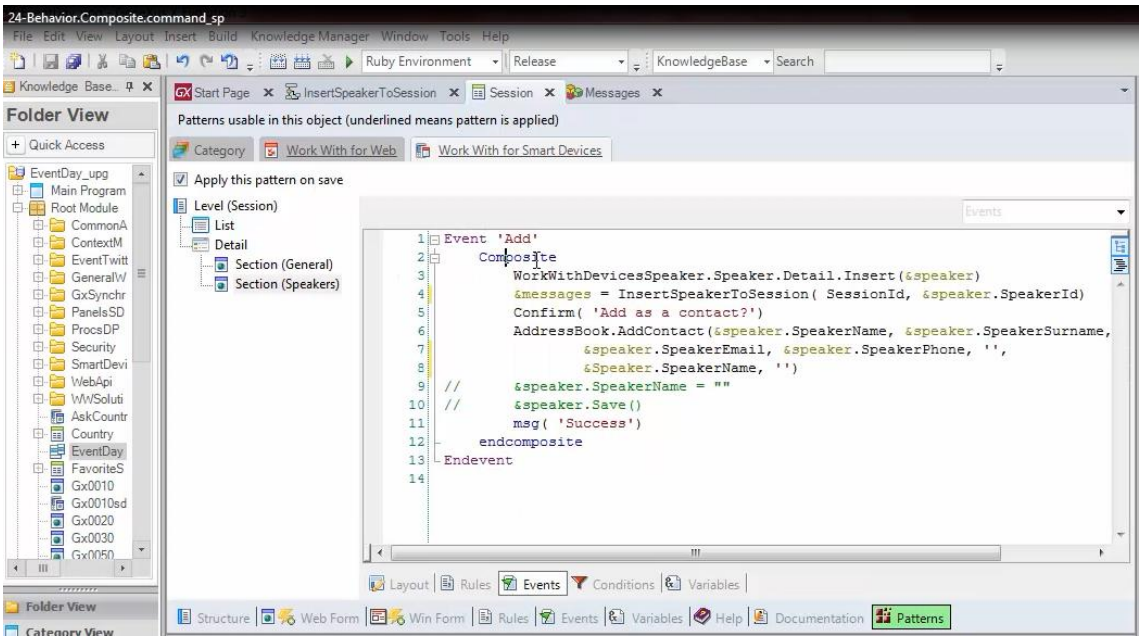


vemos que tenemos instanciada la sesión en la que nos encontramos posicionados.

Si vamos a ver el código correspondiente al botón Add

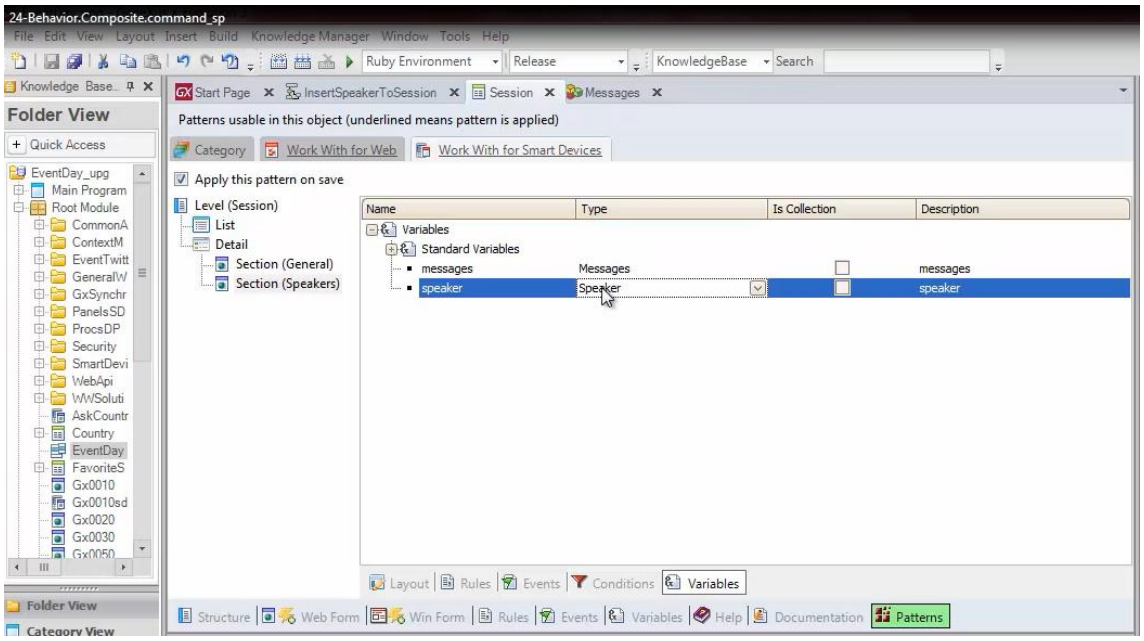


vemos que está iniciado con el comando Composite



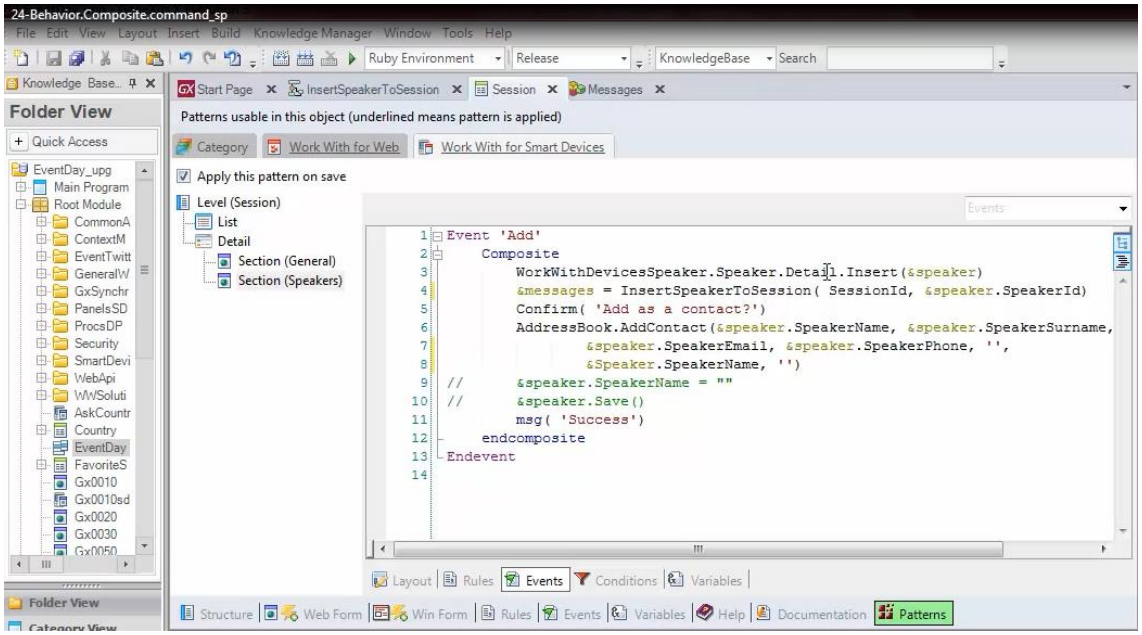
que agrupa esta serie de invocaciones, la primera de las cuales es al WorkWithDevices de Speaker … llamando al nodo Detail en modo Insert, para insertar entonces un nuevo orador al sistema.

Recordemos que el método Insert, permitía 2 posibilidades: invocar sin parámetros (puesto que el usuario iba a ingresar toda la información correspondiente al orador.. o ingresar con 1 parámetro, que tenía que ser una variable del tipo de datos: el business component correspondiente)



De esta forma, recordemos..

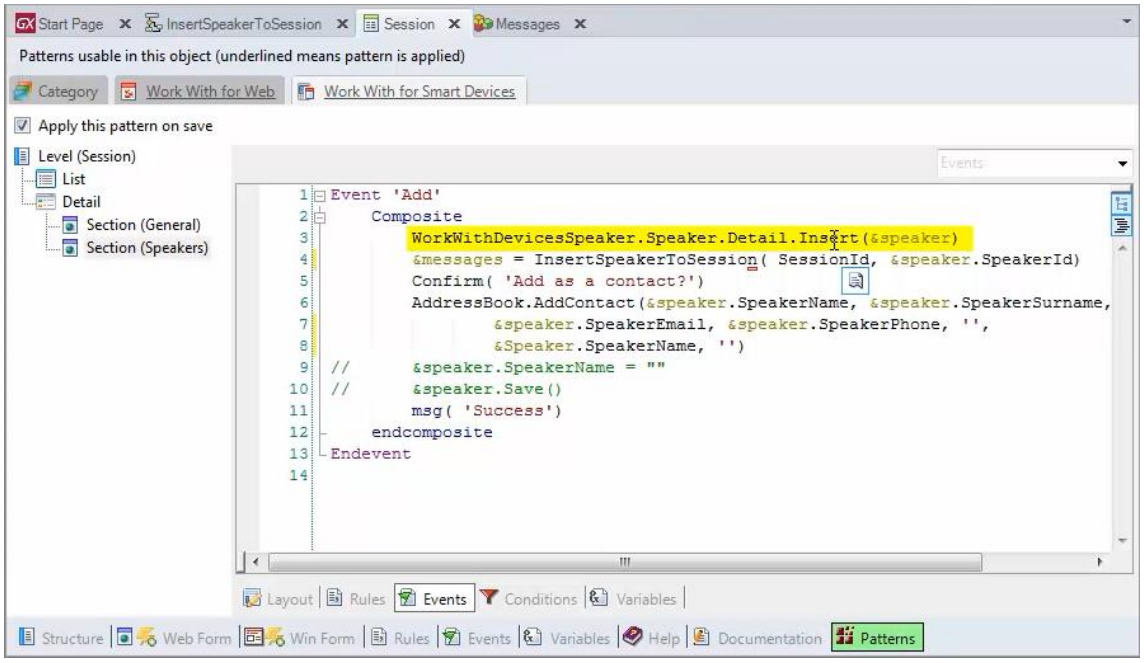
nos permitía inicializar esa pantalla de Detail con algunos valores.. y además, cuando la ejecución volviera a la siguiente cláusula



Iba a venir con la variable cargada con los valores ingresados por el usuario.

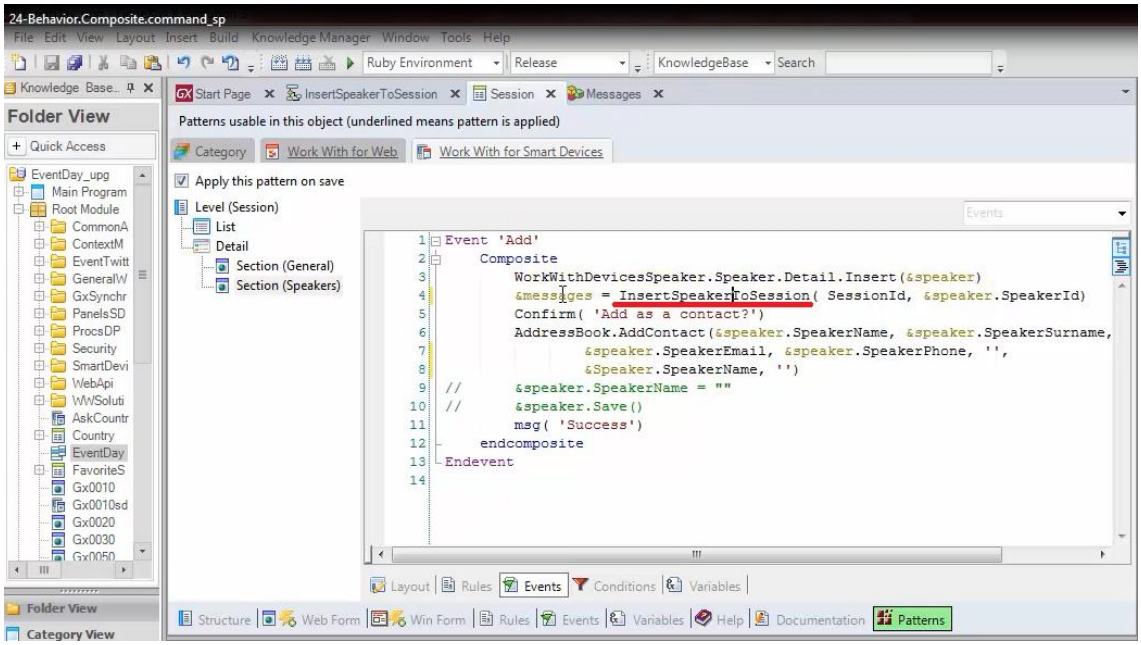
De esta manera podemos utilizar por ejemplo, el SpeakerId que fue asignado para ese orador.

Vemos que aquí estamos llamando a un procedimiento, al que le estamos pasando el identificador de sesión.. y ese orador recién insertado. ¿Para qué? Para poder insertarlo como orador de la sesión, puesto que aquí



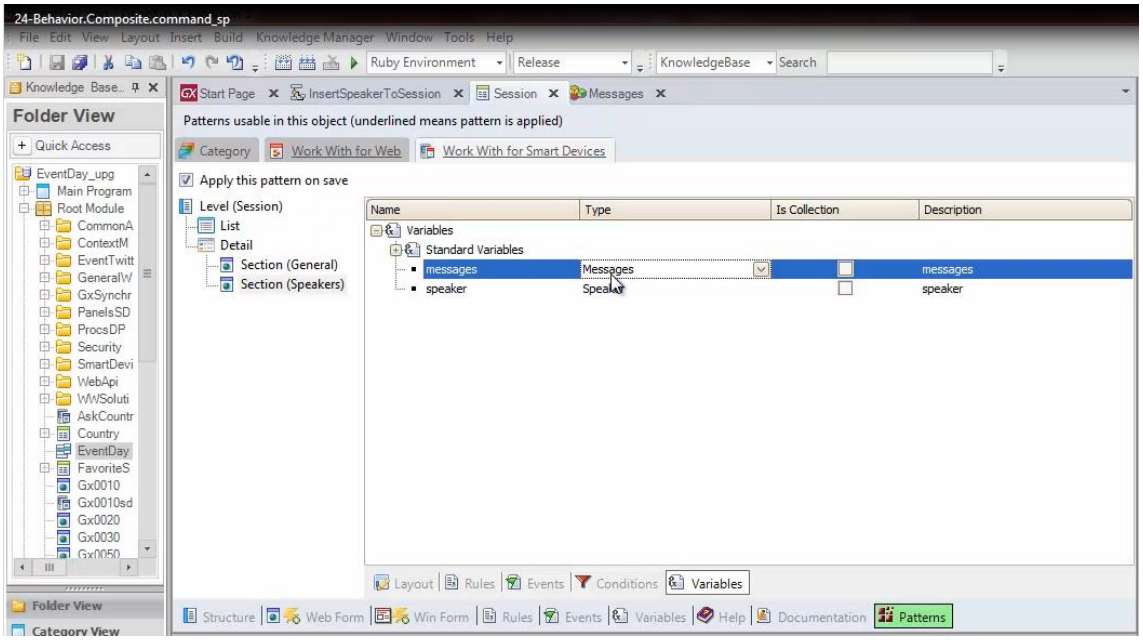
lo único que se ha hecho es insertar el orador en la tabla de oradores.

Si vamos a ver este procedimiento

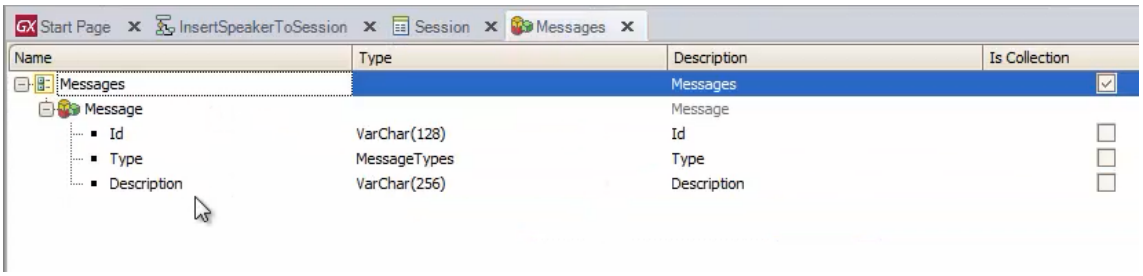


vemos que está devolviendo en la variable &messages

que es del tipo de datos: Messages predefinido



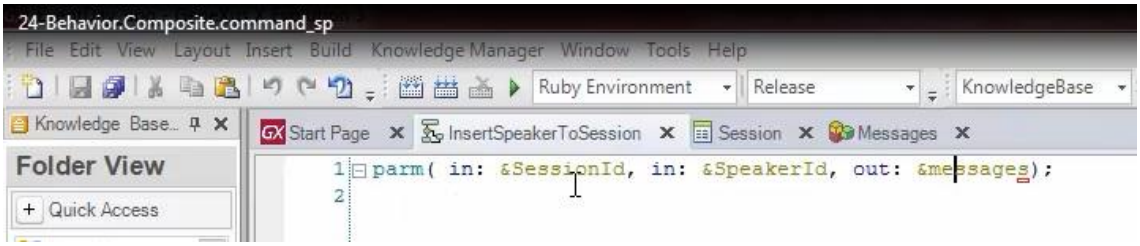
… esta colección:



que recordemos: era utilizada por los business components, para poder cargar todos los mensajes de advertencia y error, que se produjeran como consecuencia de utilizar el business component.

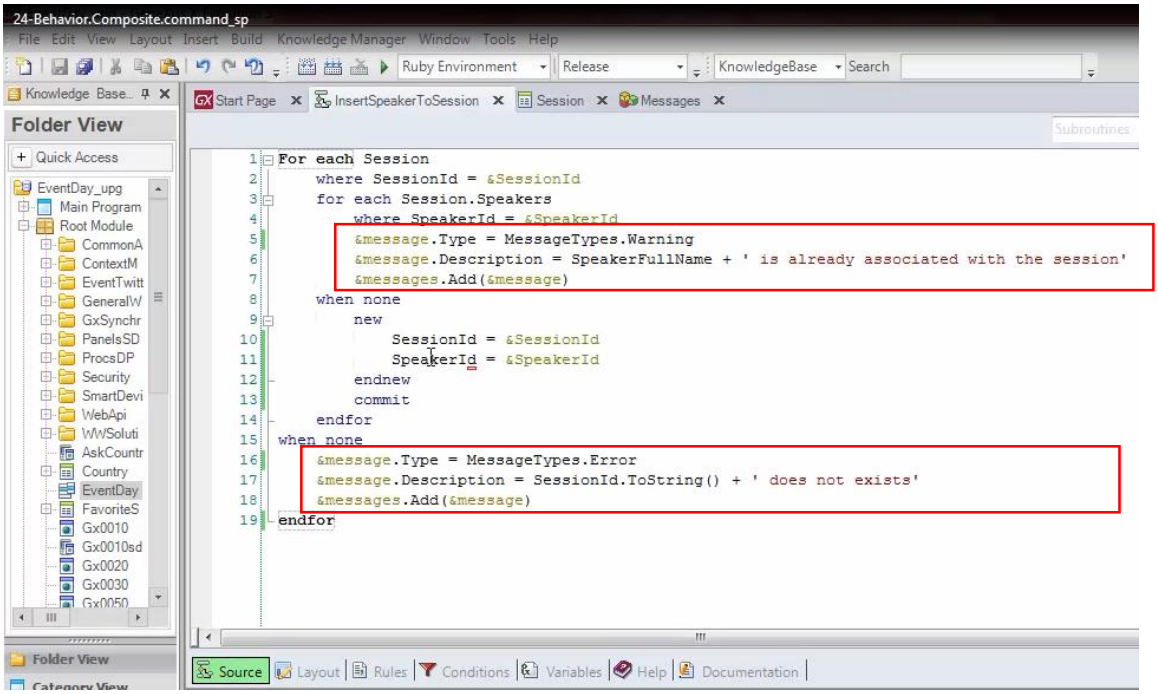
Decíamos entonces, el procedimiento va a devolver esa variable cargada.

Si vamos al procedimiento… primero empecemos por las reglas..



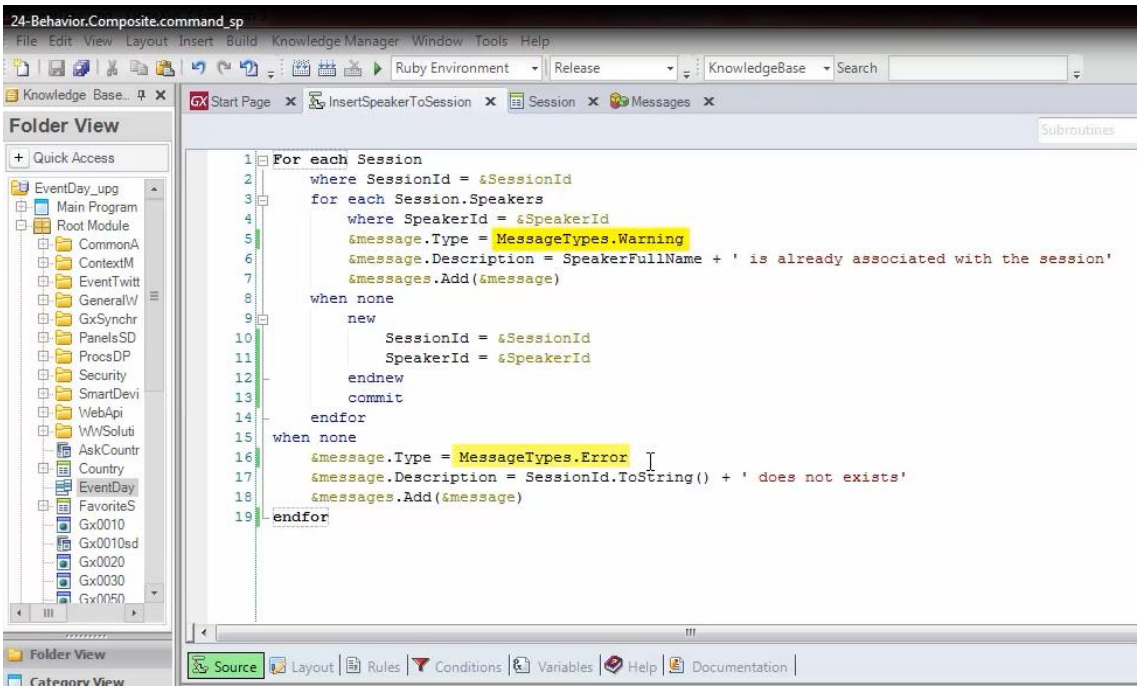
Vemos que está recibiendo en estas 2 variables… y devolviendo esa variable &messages, colección..

Vamos a su source… y antes de analizarlo

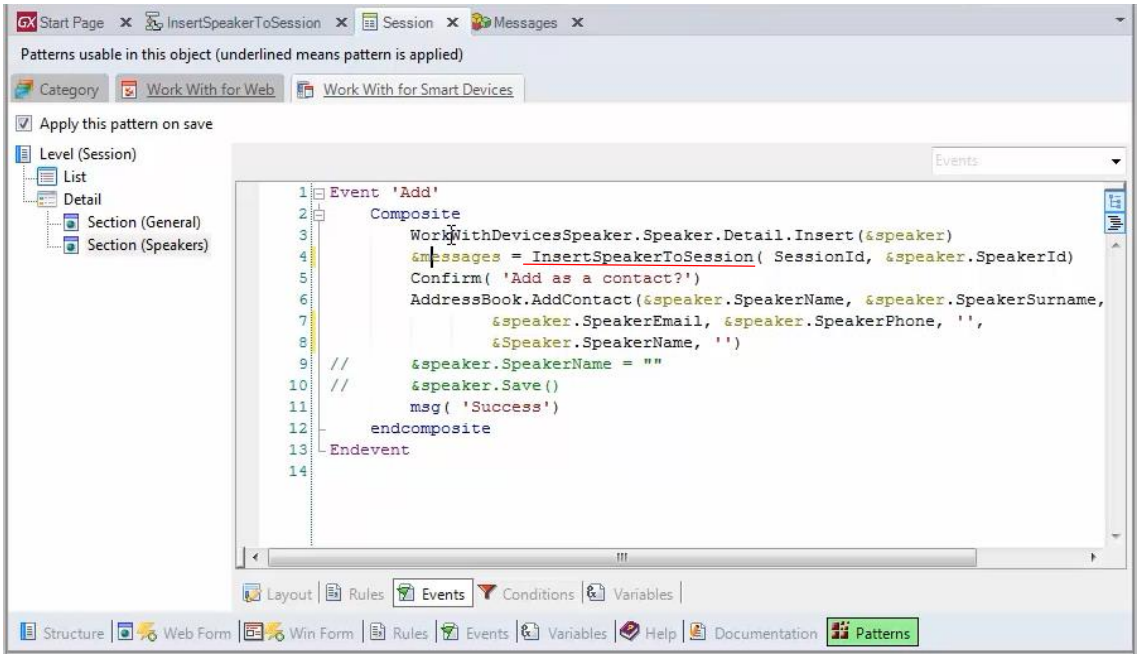


Veamos que estamos ingresando aquí un elemento a esa colección… y aquí otro.

Este de tipo Warning.. este de tipo Error:



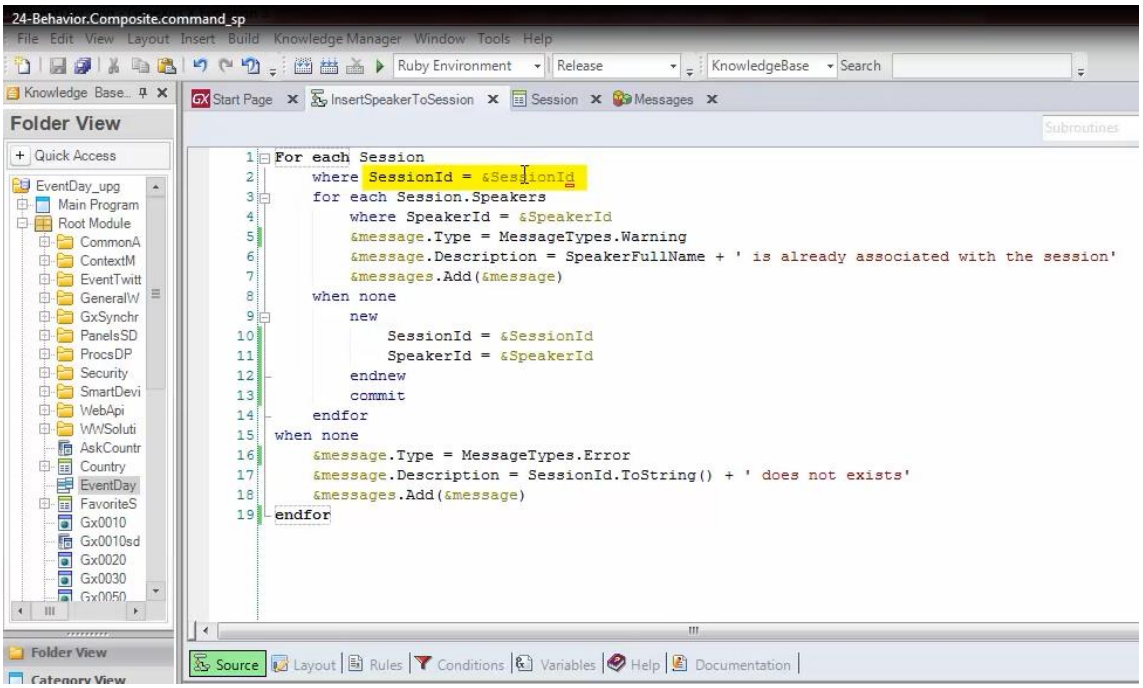
Al devolver esta colección de mensajes



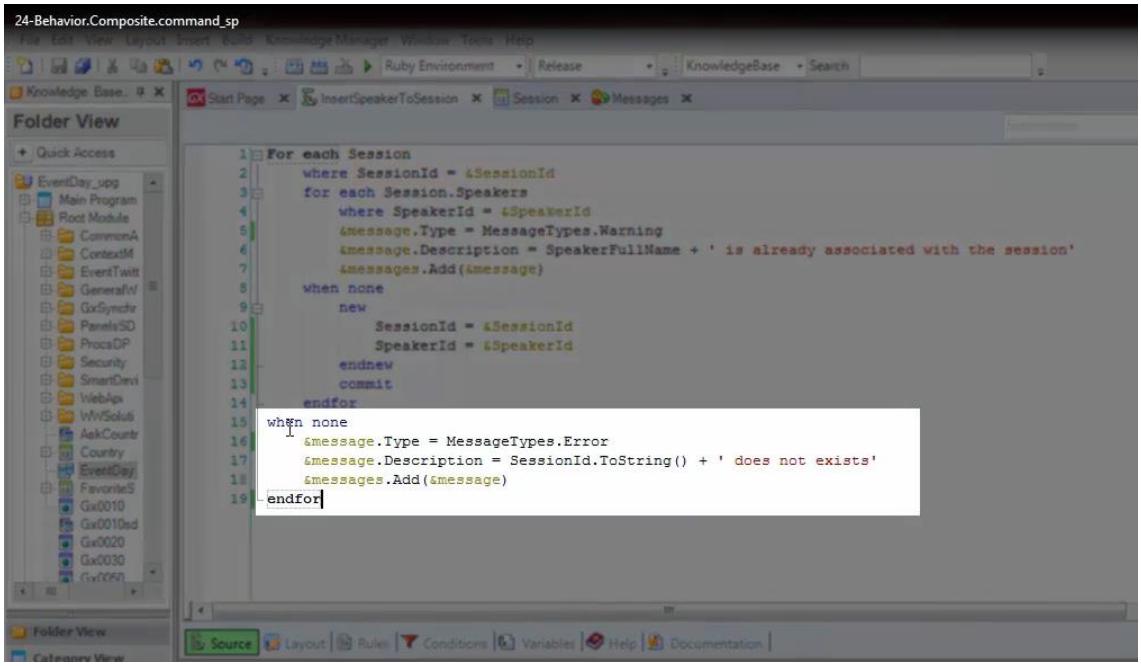
y estar este procedimiento dentro del comando Composite, esos mensajes se van a manejar automáticamente y desplegarse automáticamente al usuario en la pantalla.

En nuestro caso el procedimiento, va a devolver vacía esta colección. Veamos por qué..

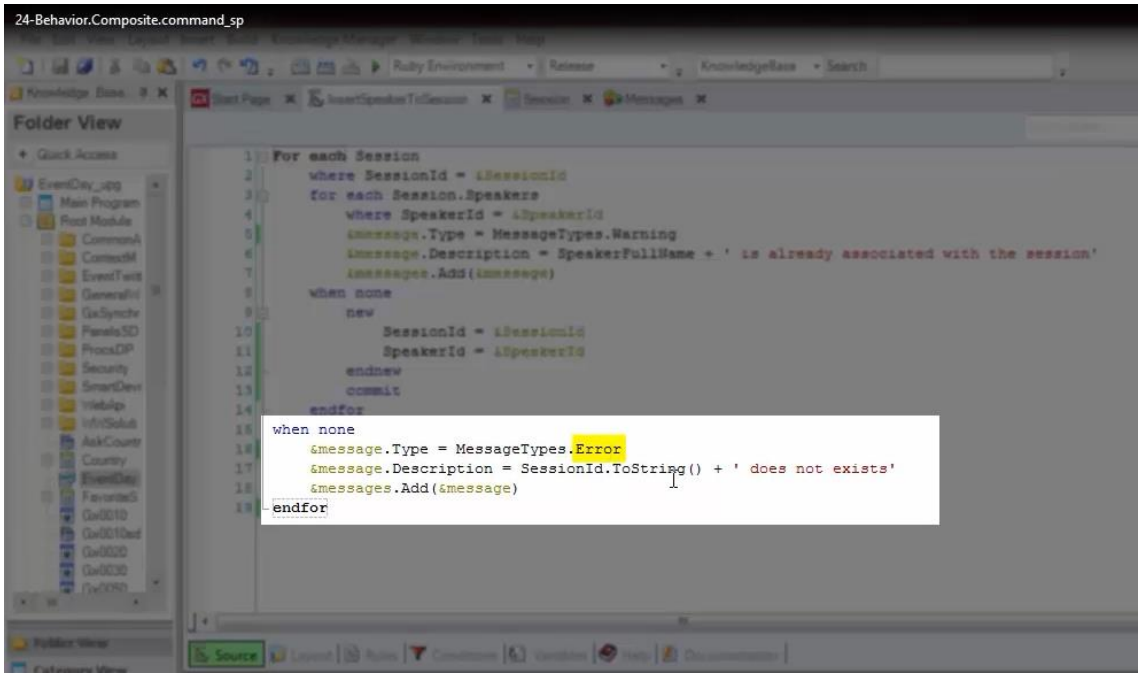
El procedimiento está recorriendo las Sessions filtrando por la que recibió por parámetro



Si no existe esa sesión… entonces viene por el when none…

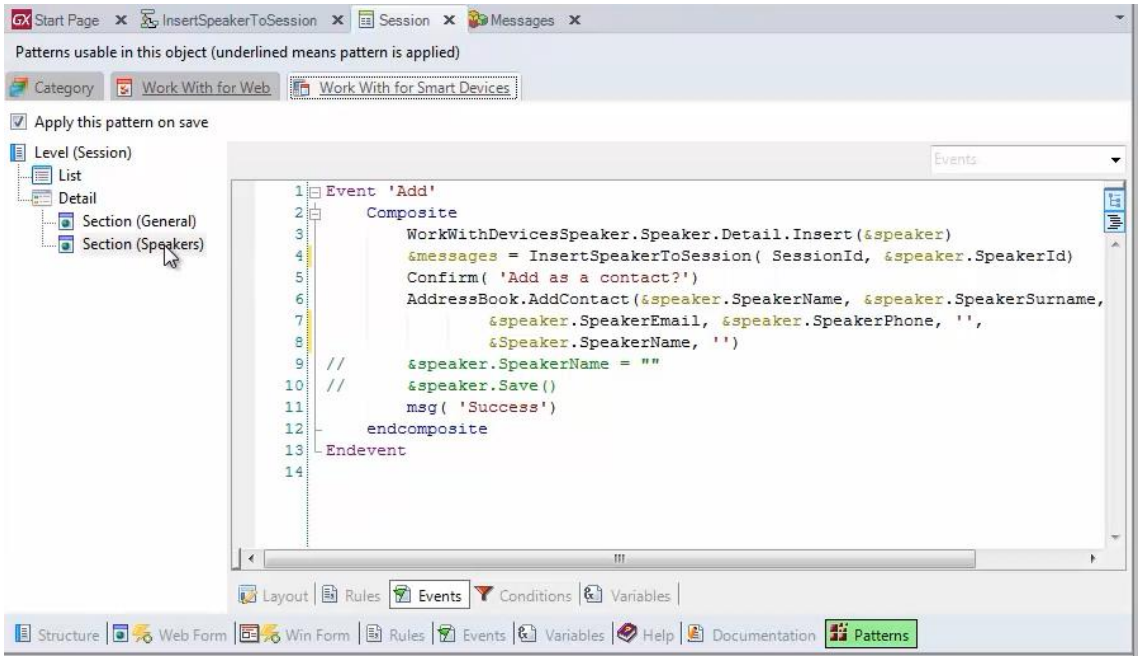


y ahí inserta un elemento en esa colección, de tipo Error..



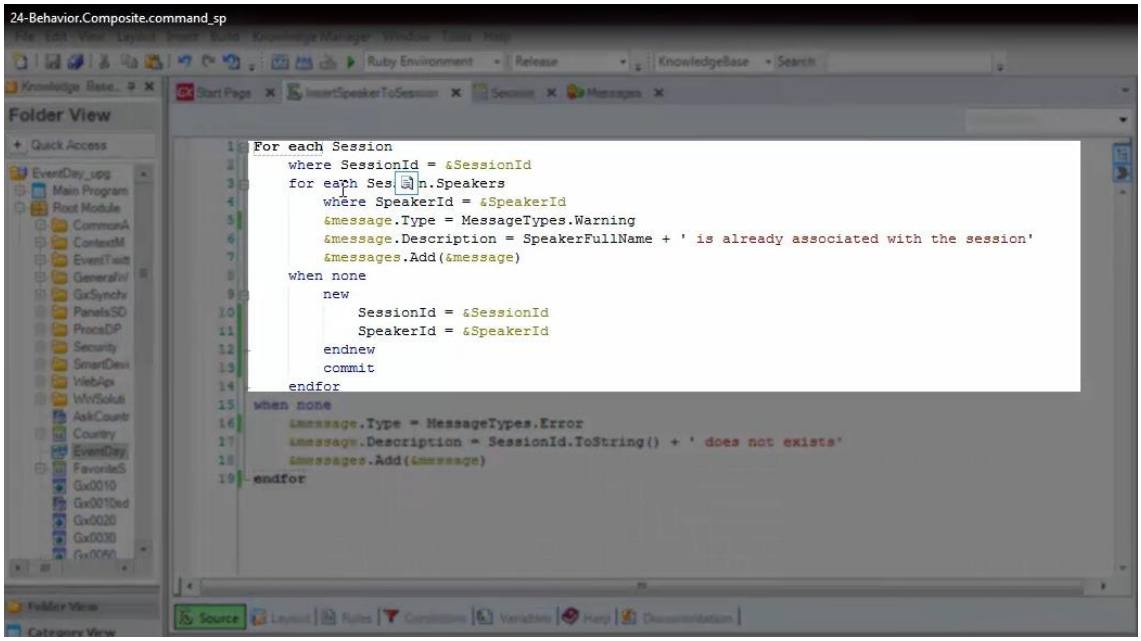
que está indicando que esa sesión no existe.

No va a ser nuestro caso, puesto que la sesión estamos seguros que existe, porque estamos a nivel del Detail

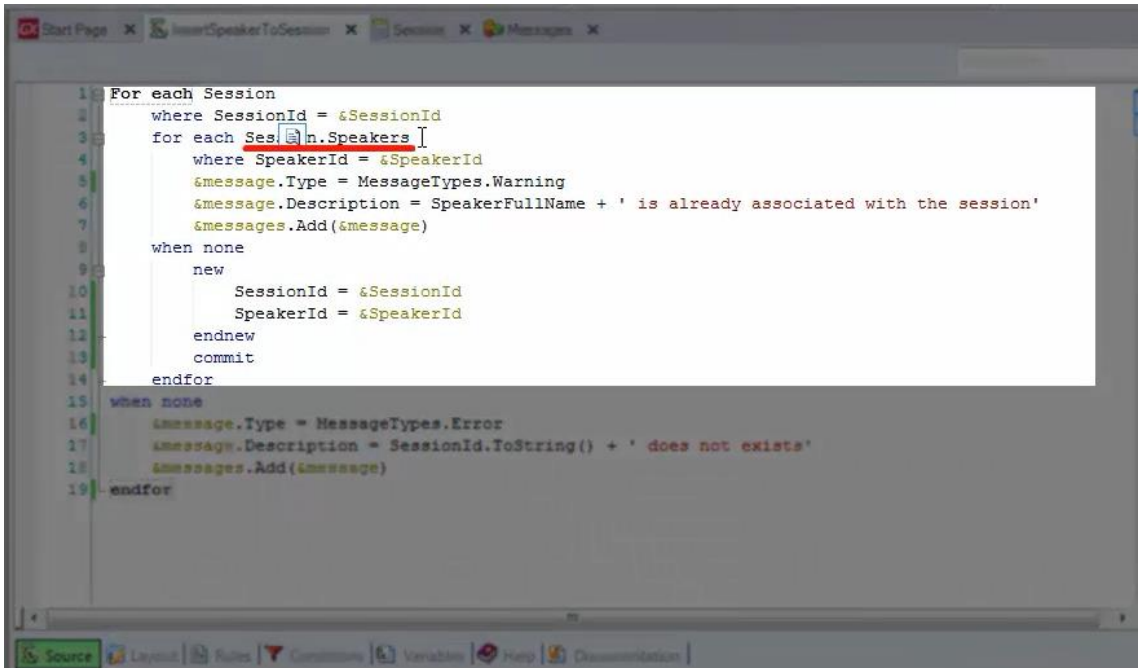


es decir, está instanciada

Si volvemos al procedimiento

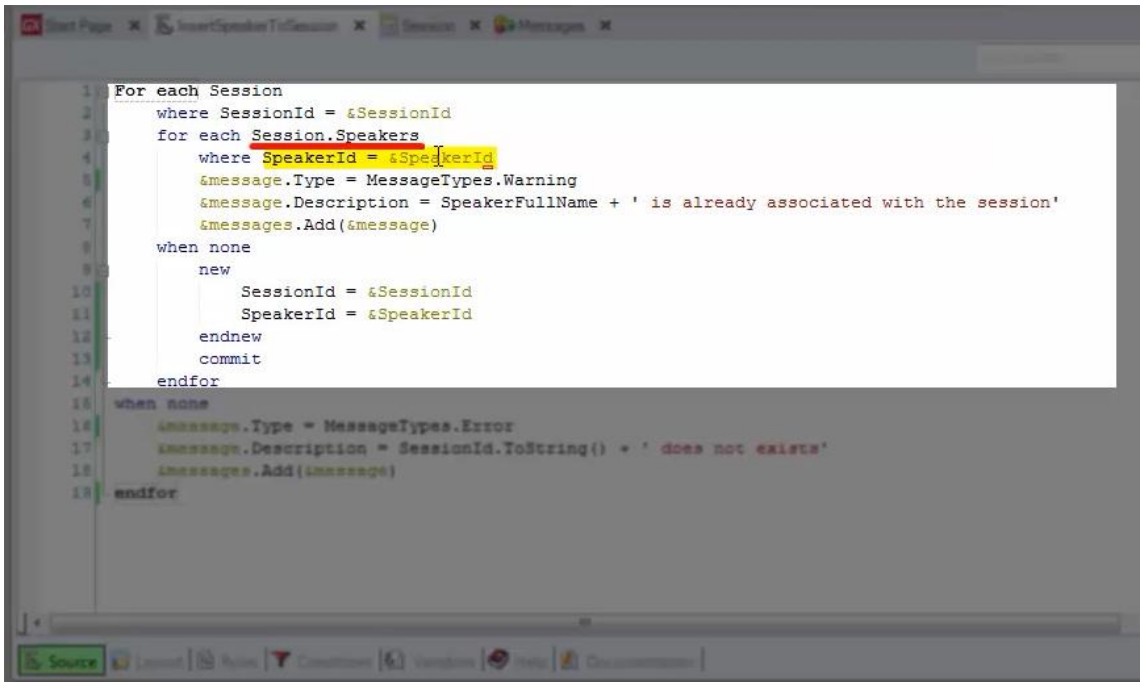


vemos que la sesión existe, entonces lo que hacemos es un For each a ka tabla subordinada de Oradores



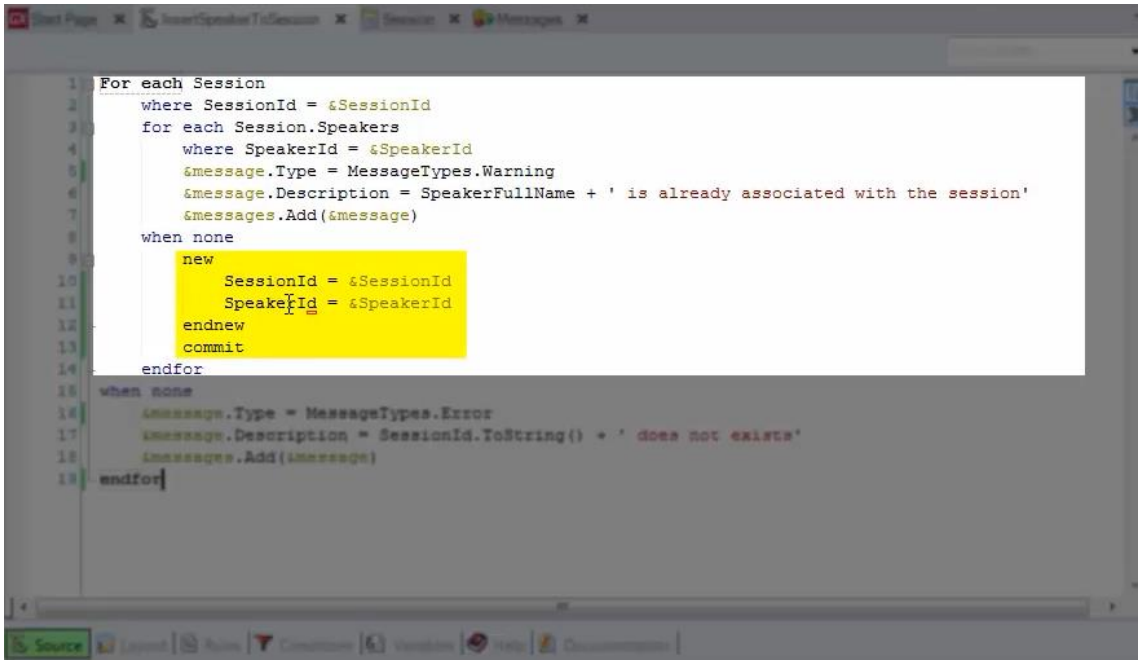
```
For each Session
    where SessionId = &SessionId
    for each Session.Speakers
        where SpeakerId = &SpeakerId
        &message.Type = MessageTypes.Warning
        &message.Description = SpeakerFullName + ' is already associated with the session'
        &messages.Add(&message)
    when none
        new
            SessionId = &SessionId
            SpeakerId = &SpeakerId
        endnew
        commit
    endfor
when none
    &message.Type = MessageTypes.Error
    &message.Description = SessionId.ToString() + ' does not exists'
    &messages.Add(&message)
endfor
```

para ver si ese orador ya está asociado a esa sesión



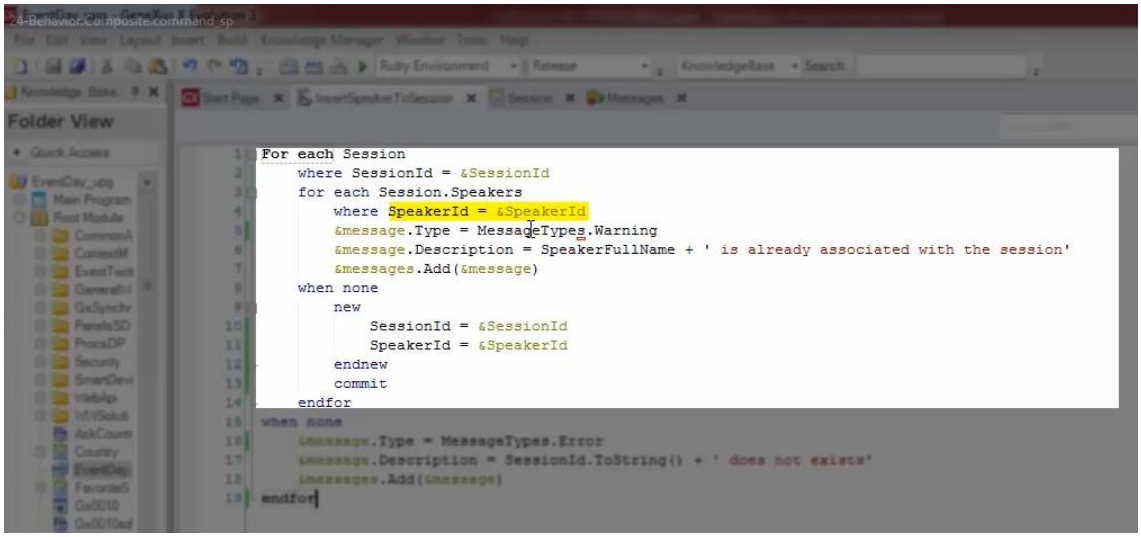
```
For each Session
    where SessionId = &SessionId
    for each Session.Speakers
        where SpeakerId = &SpeakerId
        &message.Type = MessageTypes.Warning
        &message.Description = SpeakerFullName + ' is already associated with the session'
        &messages.Add(&message)
    when none
        new
            SessionId = &SessionId
            SpeakerId = &SpeakerId
        endnew
        commit
    endfor
when none
    &message.Type = MessageTypes.Error
    &message.Description = SessionId.ToString() + ' does not exists'
    &messages.Add(&message)
endfor
```

No va a ser nuestro caso porque recién lo hemos creado al orador, por tanto va a entrar por el when none en nuestro caso… y a dar de alta entonces un nuevo registro que corresponde a esa sesión y a ese orador



```
For each Session
    where SessionId = &SessionId
    for each Session.Speakers
        where SpeakerId = &SpeakerId
        &message.Type = MessageTypes.Warning
        &message.Description = SpeakerFullName + ' is already associated with the session'
        &messages.Add(&message)
    when none
        new
            SessionId = &SessionId
            SpeakerId = &SpeakerId
        endnew
        commit
    endfor
when none
    &message.Type = MessageTypes.Error
    &message.Description = SessionId.ToString() + ' does not exists'
    &messages.Add(&message)
endfor
```

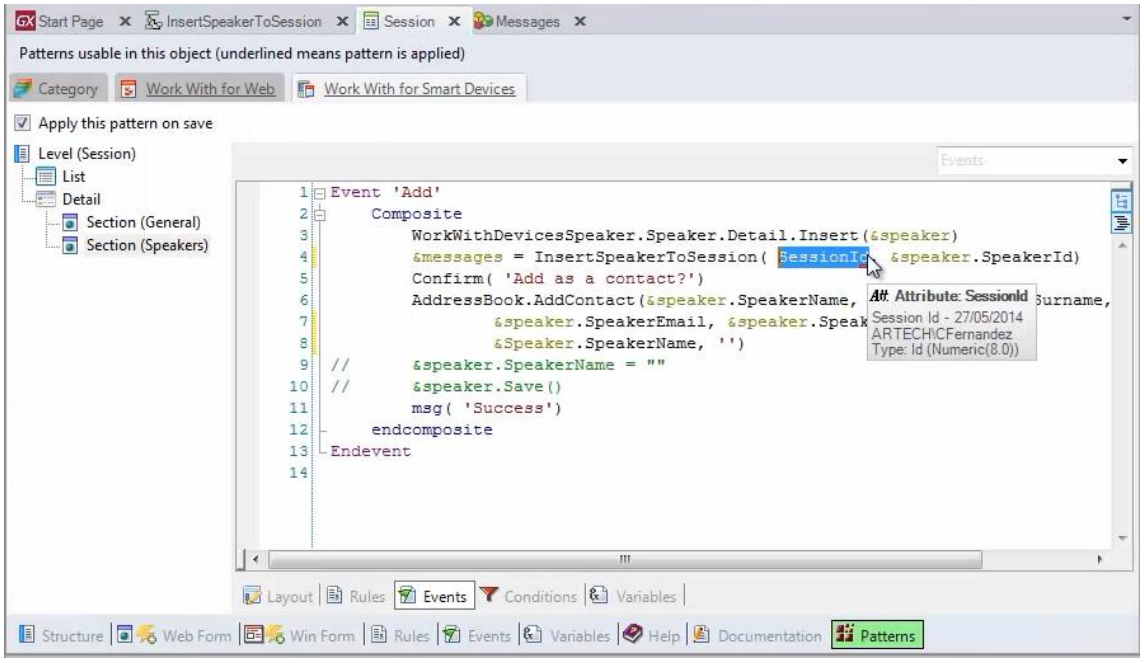
Pero si el orador sí existiera



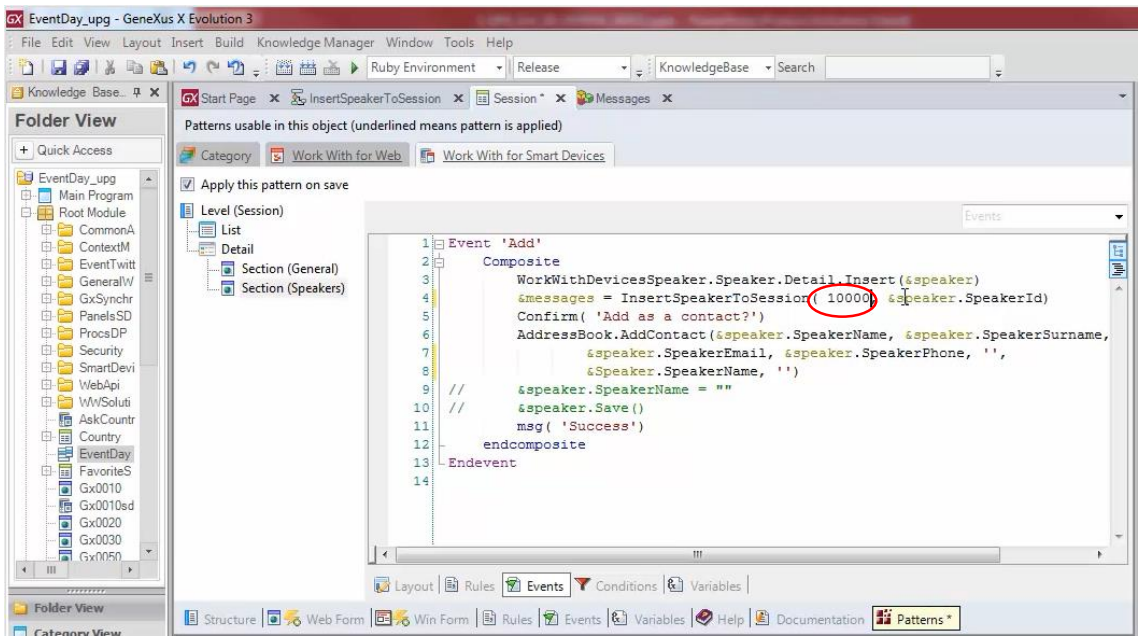
```
For each Session
    where SessionId = &SessionId
    for each Session.Speakers
        where SpeakerId = &SpeakerId
        &message.Type = MessageTypes.Warning
        &message.Description = SpeakerFullName + ' is already associated with the session'
        &messages.Add(&message)
    when none
        new
            SessionId = &SessionId
            SpeakerId = &SpeakerId
        endnew
        commit
    endfor
when none
    &message.Type = MessageTypes.Error
    &message.Description = SessionId.ToString() + ' does not exists'
    &messages.Add(&message)
endfor
```

Entonces vemos que agregamos un elemento a esa colección, de tipo advertencia, que informa sobre la situación.

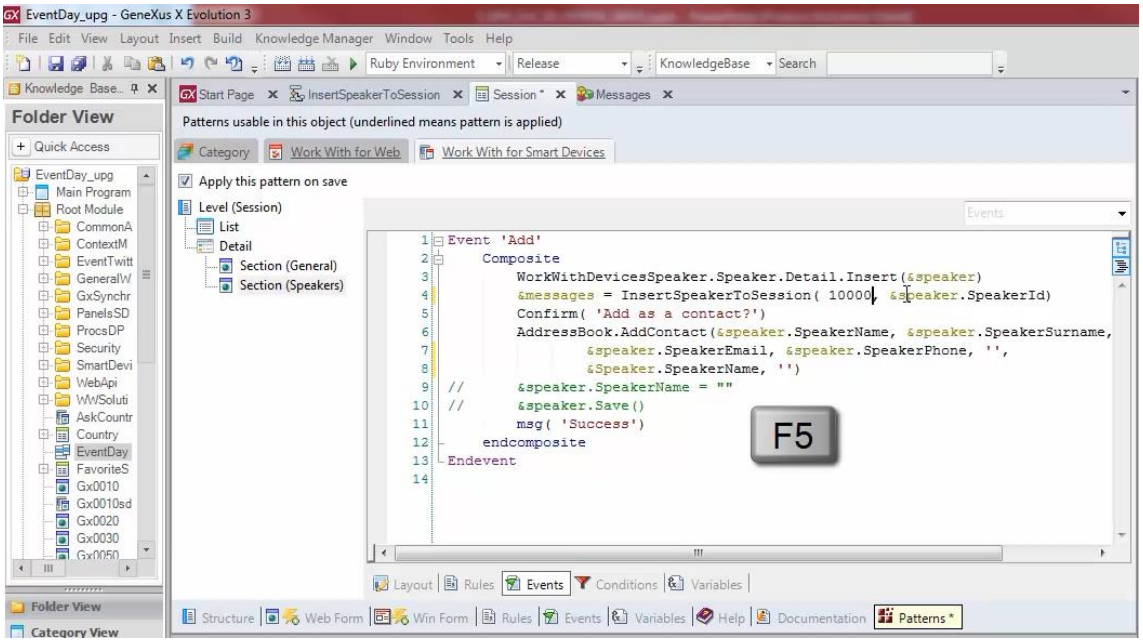
Entonces para probar lo que mencionábamos, podemos modificar provisoriamente los parámetros

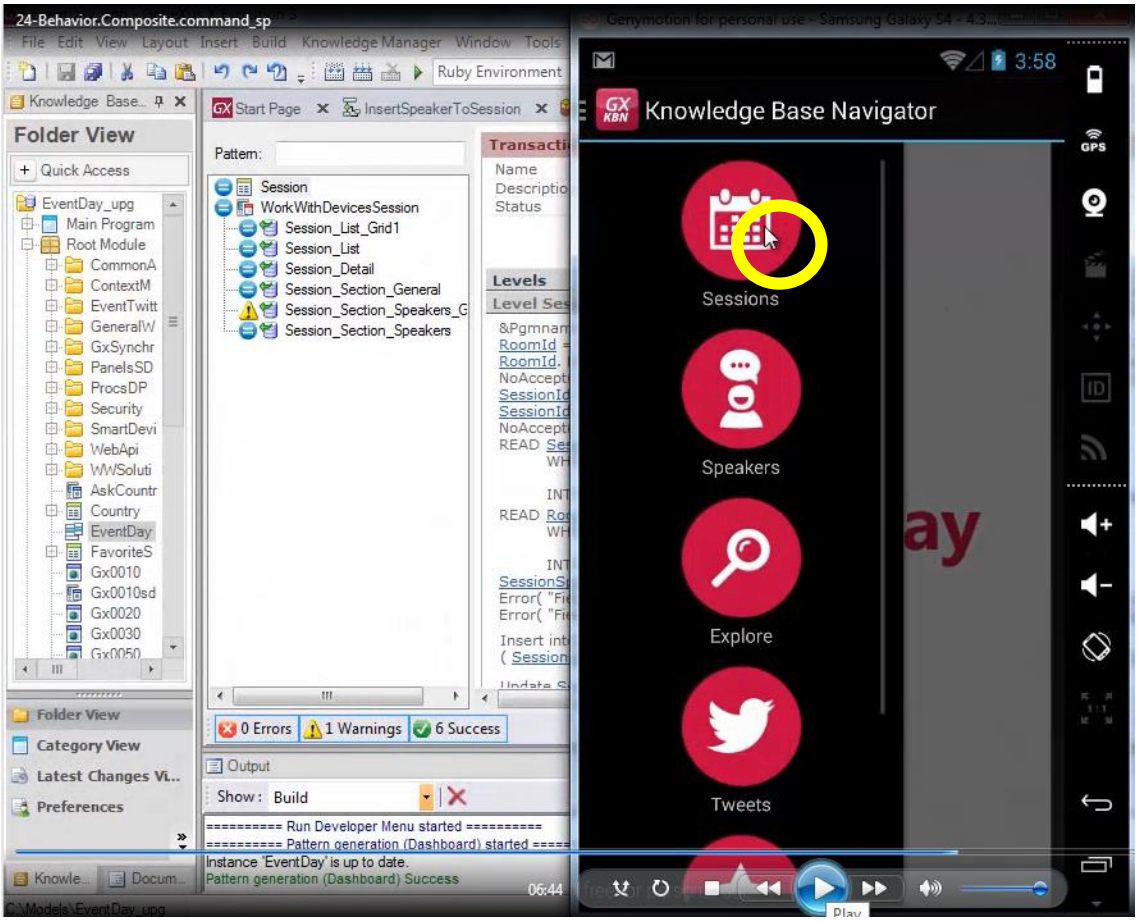


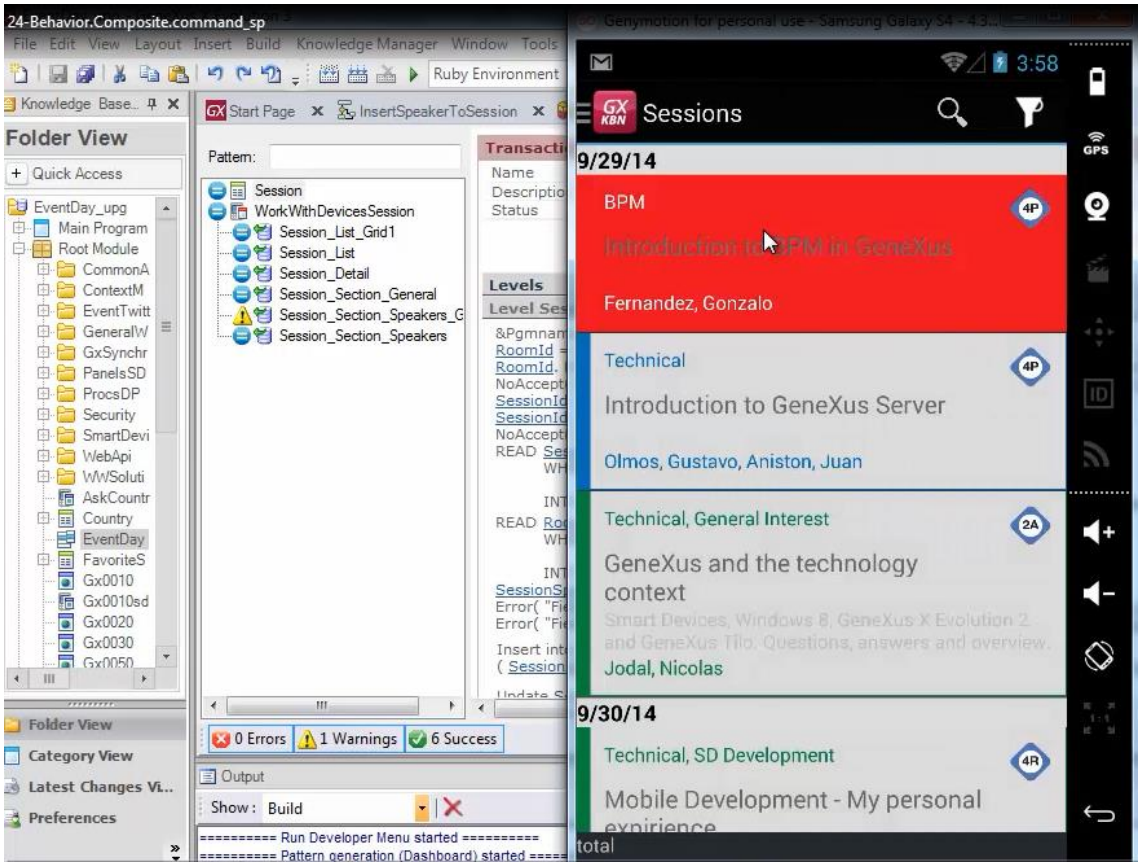
y enviar un identificador de sesión que estemos seguros que no existe



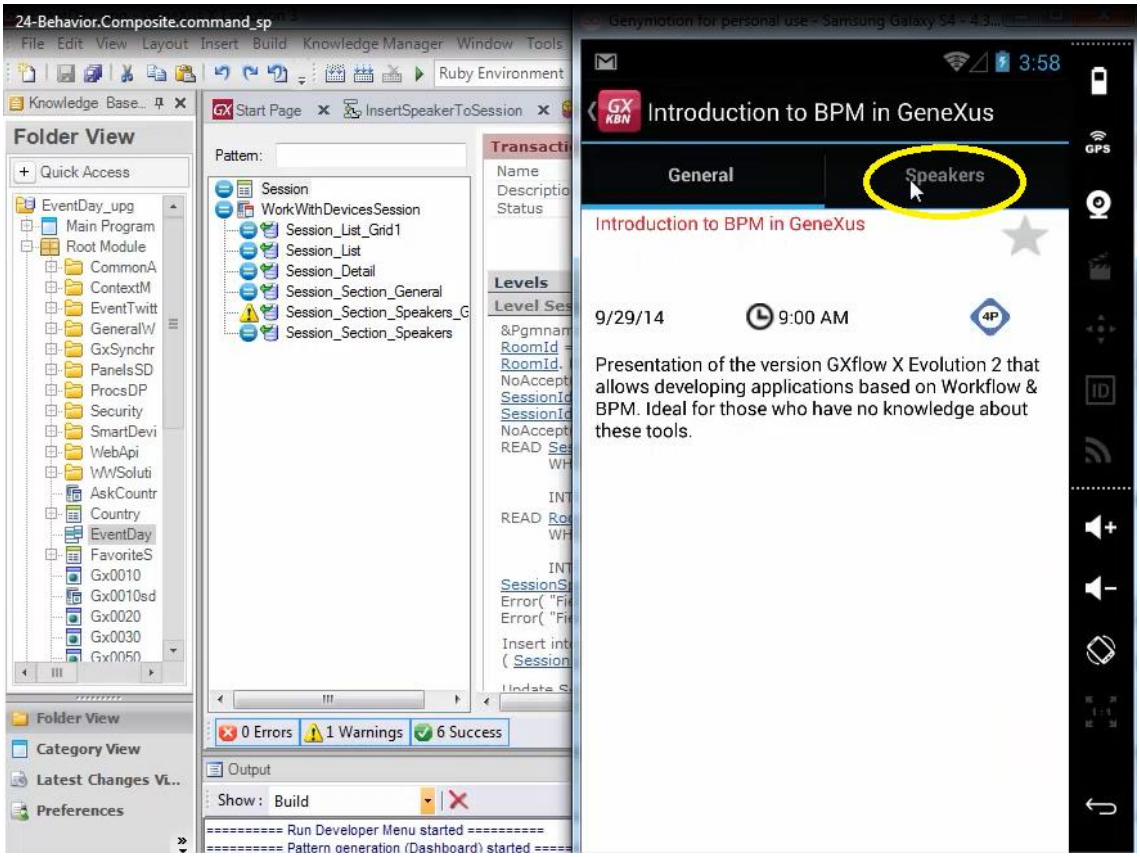
F5

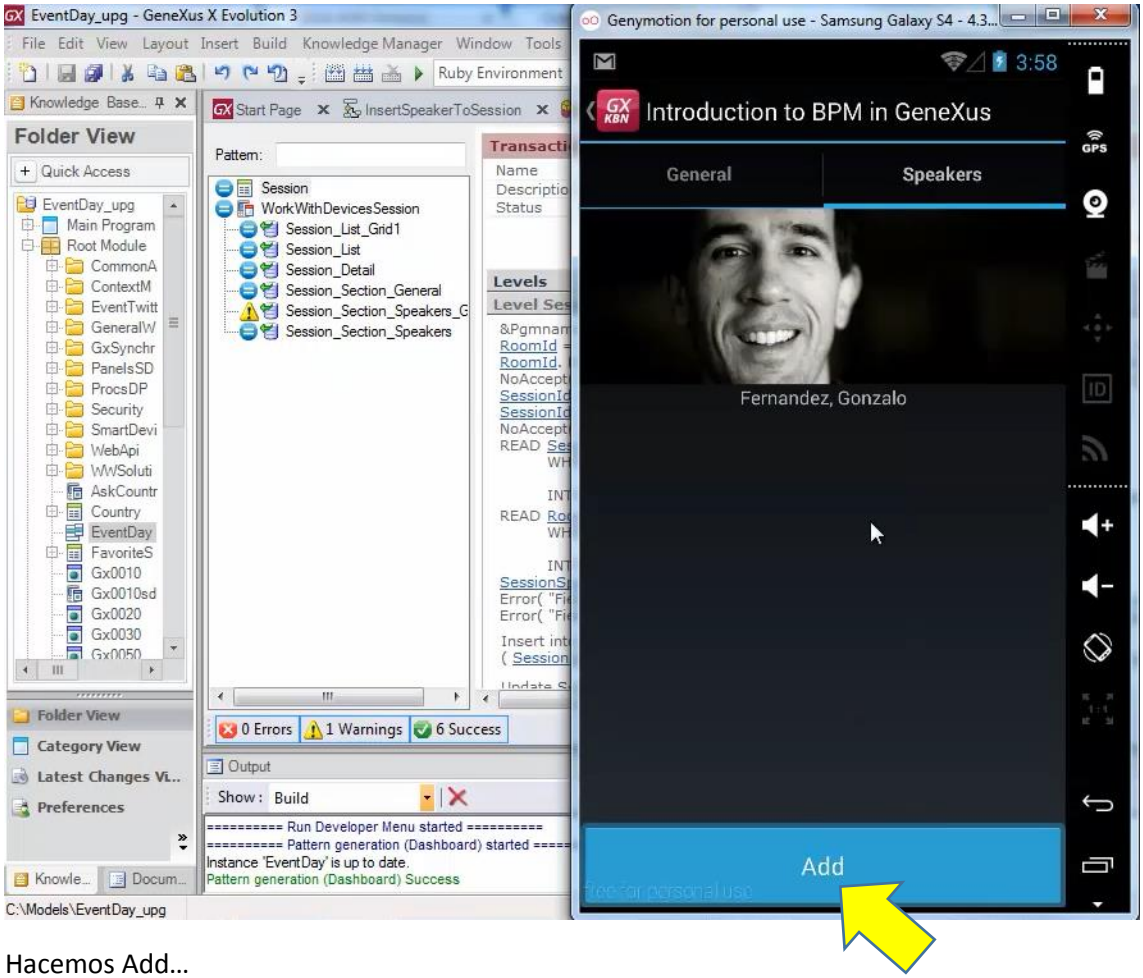

EventDay_upg - GeneXus X Evolution 3
Folder View
Patterns usable in this object (underlined means pattern is applied)
Apply this pattern on save
Level (Session)
List
Detail
Section (General)
Section (Speakers)
Event 'Add'
Composite
WorkWithDevicesSpeaker.Speaker.Detail.Insert(&speaker)
&messages = InsertSpeakerToSession(10000, &speaker.SpeakerId)
Confirm('Add as a contact?')
AddressBook.AddContact(&speaker.SpeakerName, &speaker.SpeakerSurname,
&speaker.SpeakerEmail, &speaker.SpeakerPhone, '',
&Speaker.SpeakerName, '')
// &speaker.SpeakerName = ""
// &speaker.Save()
msg('Success')
endcomposite
Endevent
F5


24-Behavior.Composite.command_sp
Folder View
Pattern:
Session
WorkWithDevicesSession
Session_List_Grid1
Session_List
Session_Detail
Session_Section_General
Session_Section_Speakers_G
Session_Section_Speakers
0 Errors
1 Warnings
6 Success
Instance 'EventDay' is up to date.
Pattern generation (Dashboard) Success
Knowledge Base Navigator
Sessions
Speakers
Explore
Tweets

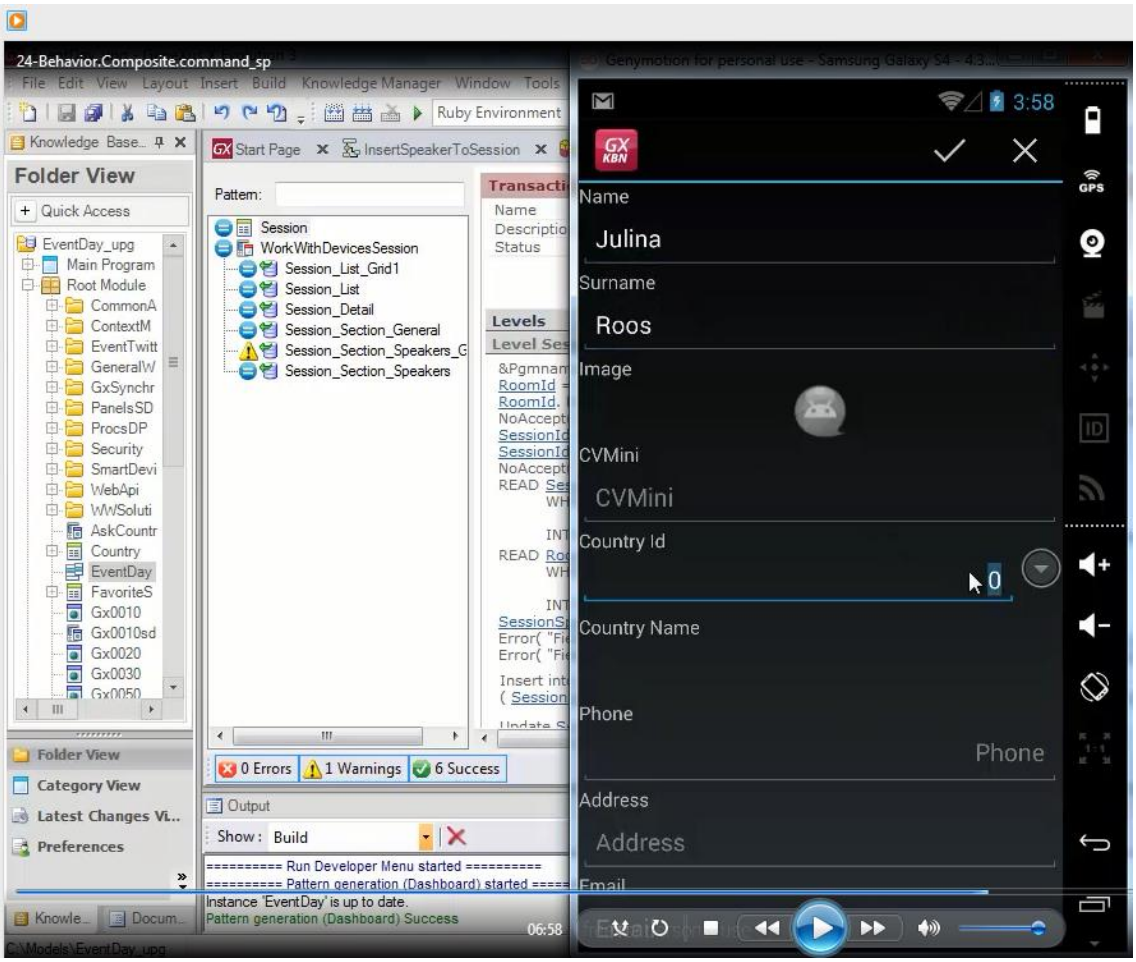


Vamos a editar una sesión

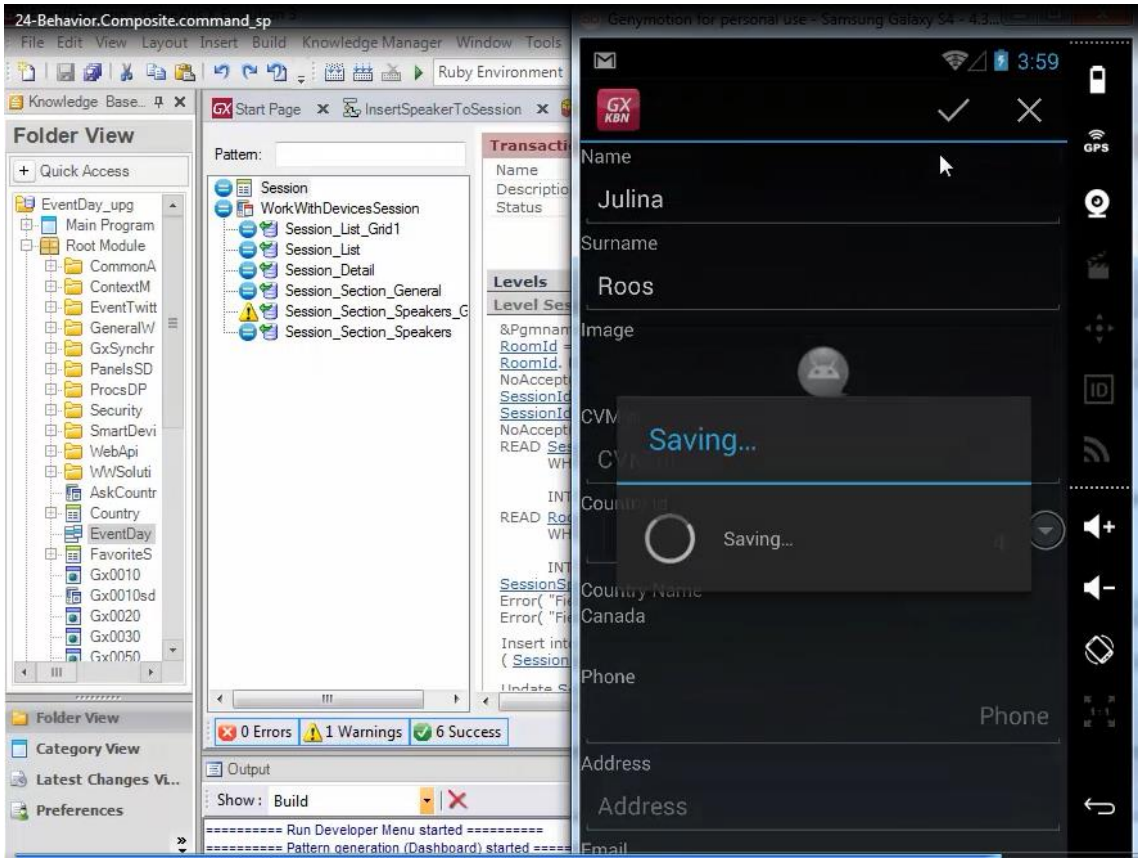


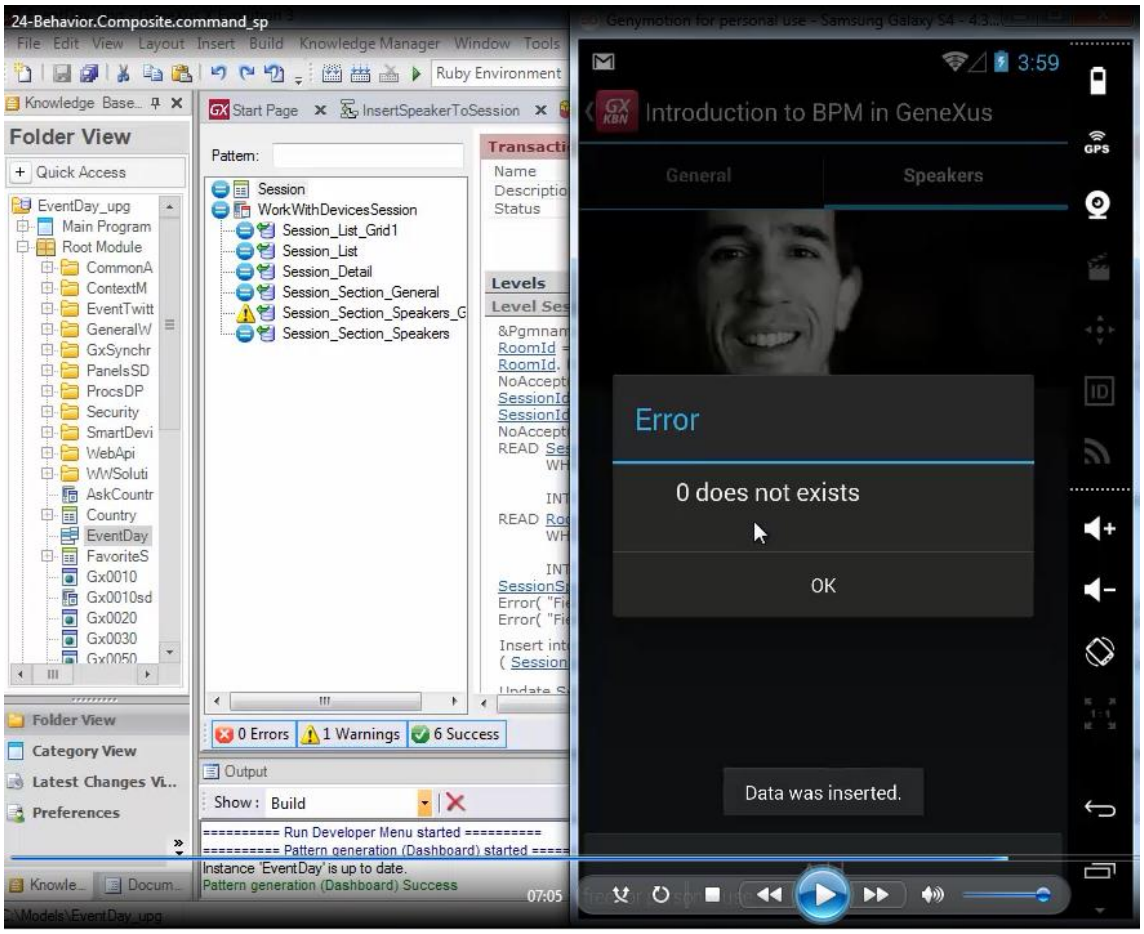

EventDay_upg - GeneXus X Evolution 3
Genymotion for personal use - Samsung Galaxy S4 - 4.3...
Introduction to BPM in GeneXus
General
Speakers
Fernandez, Gonzalo
Add
Folder View
Start Page
InsertSpeakerToSession
Session
WorkWithDevicesSession
Session_List_Grid1
Session_List
Session_Detail
Session_Section_General
Session_Section_Speakers
Session_Section_Speakers_G
0 Errors
1 Warnings
6 Success
Output
Show : Build
Instance 'EventDay' is up to date.
Pattern generation (Dashboard) Success
C:\Models\EventDay_upg

Hacemos Add…



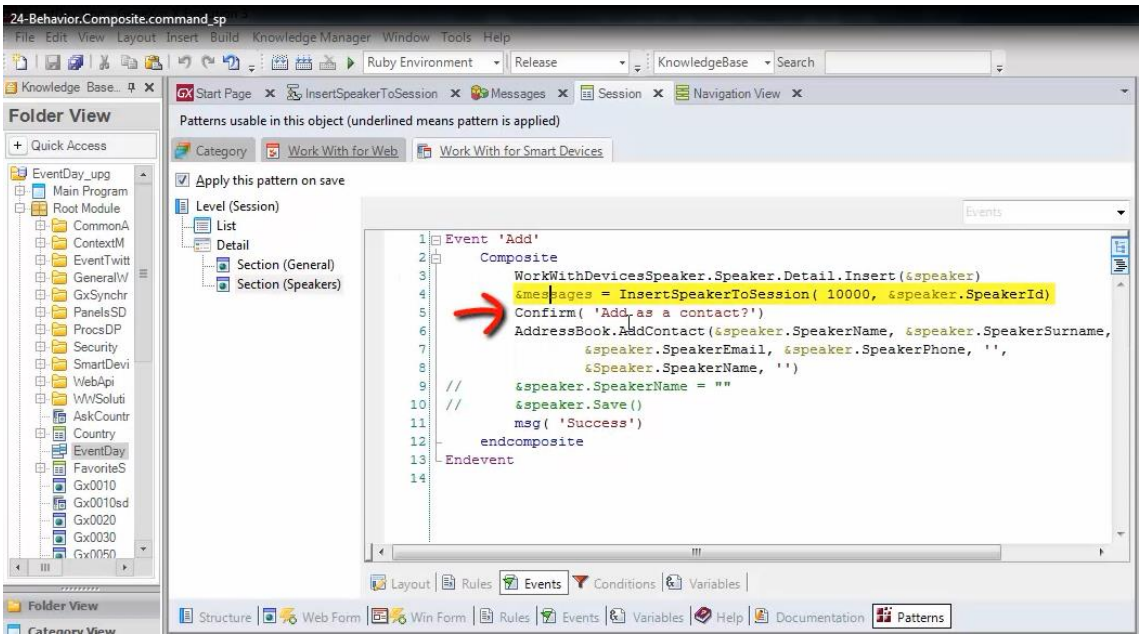
ponemos un código de país… y grabamos



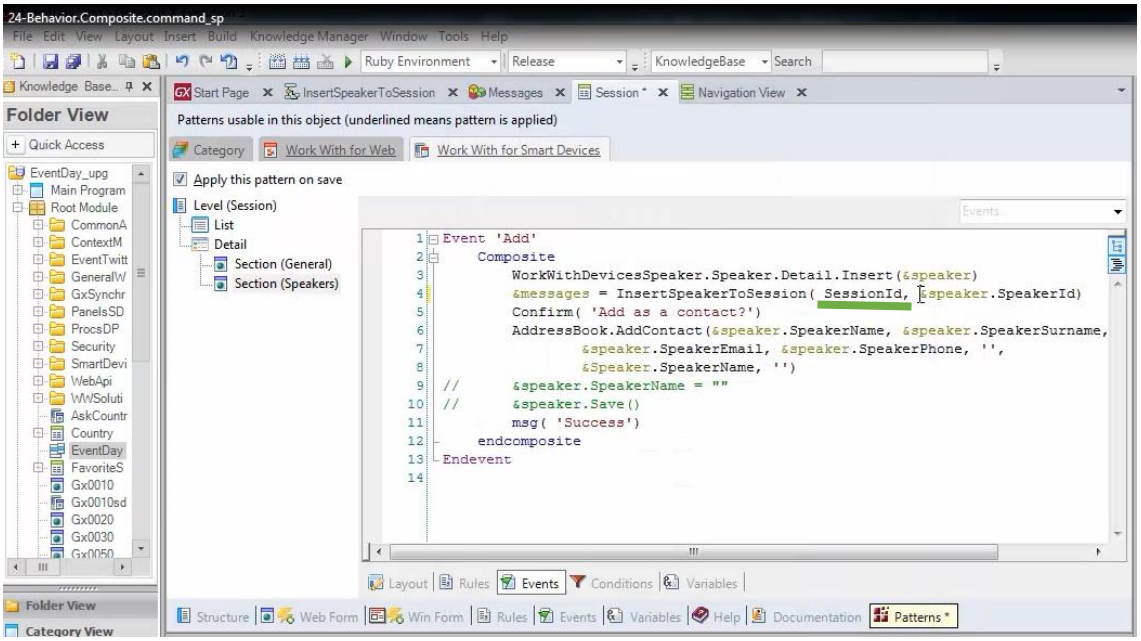


y vemos que nos está dando el error de que el registro no existe

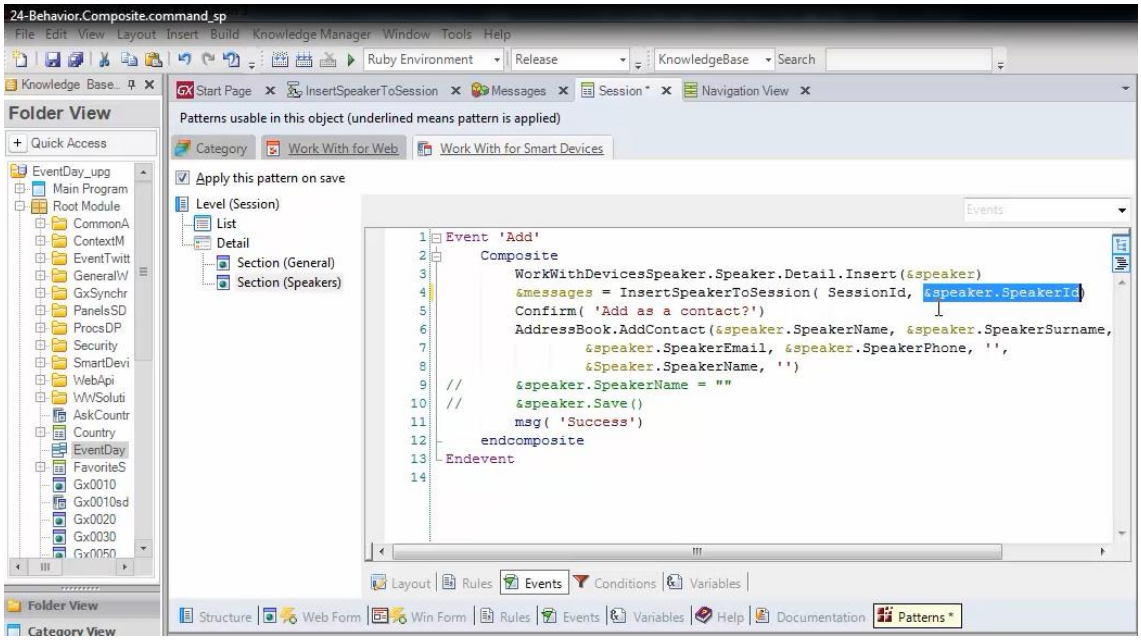
Damos OK y vemos que la ejecución se interrumpió.. es decir, no siguió ejecutándose lo que venía luego de la invocación al procedimiento



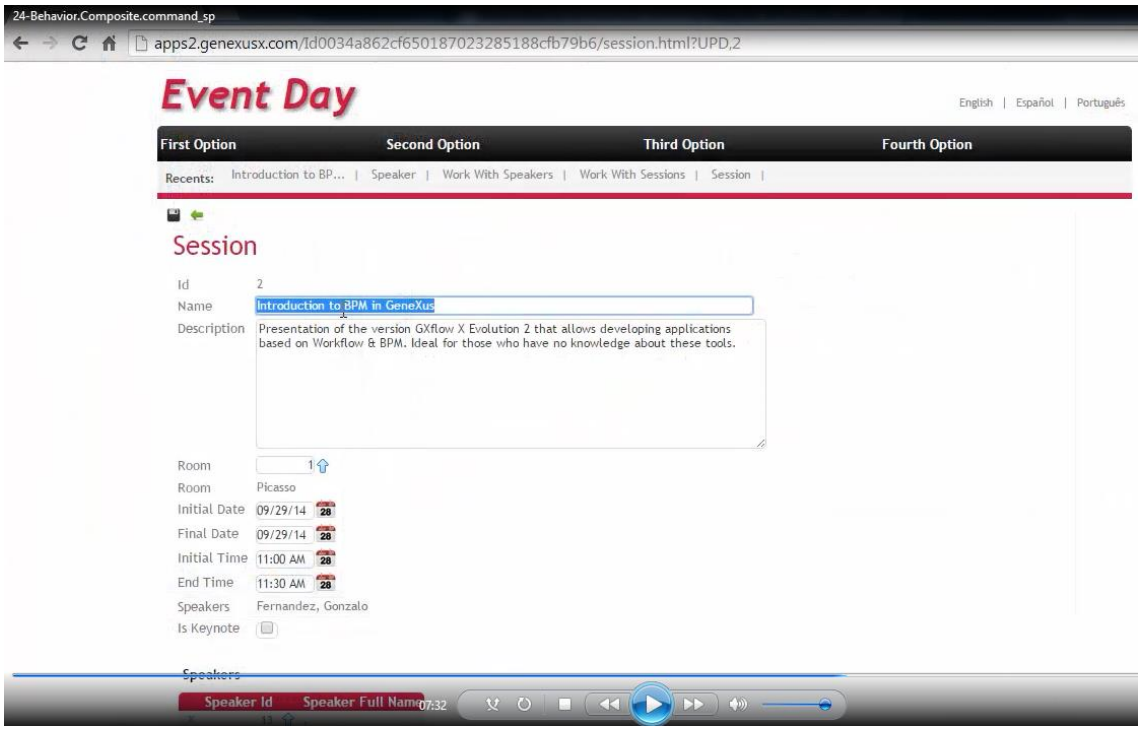
Si ahora en cambio, dejamos el atributo SessionId como estaba

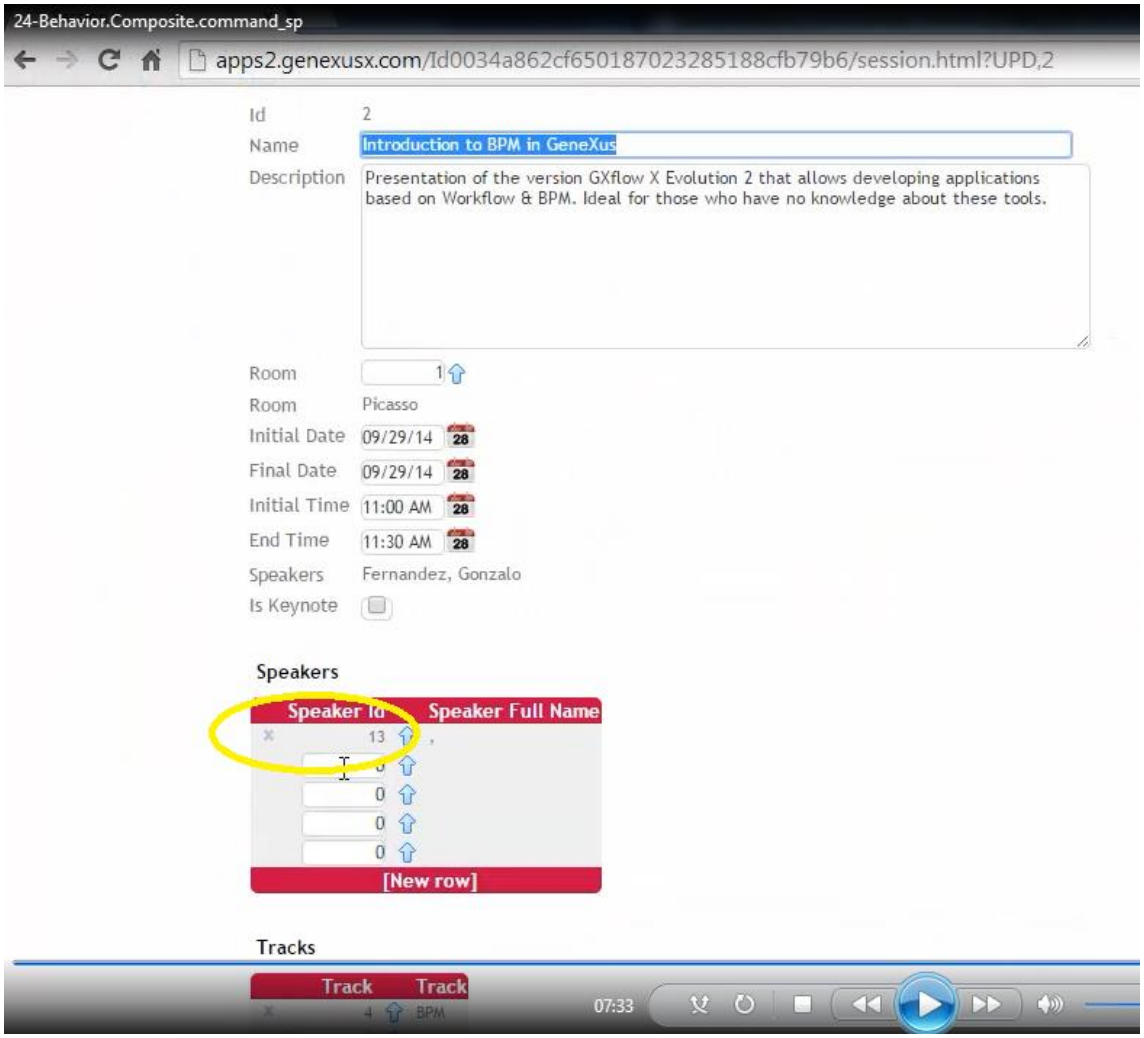


… y ahora especificamos un speaker que sí ya existiera…

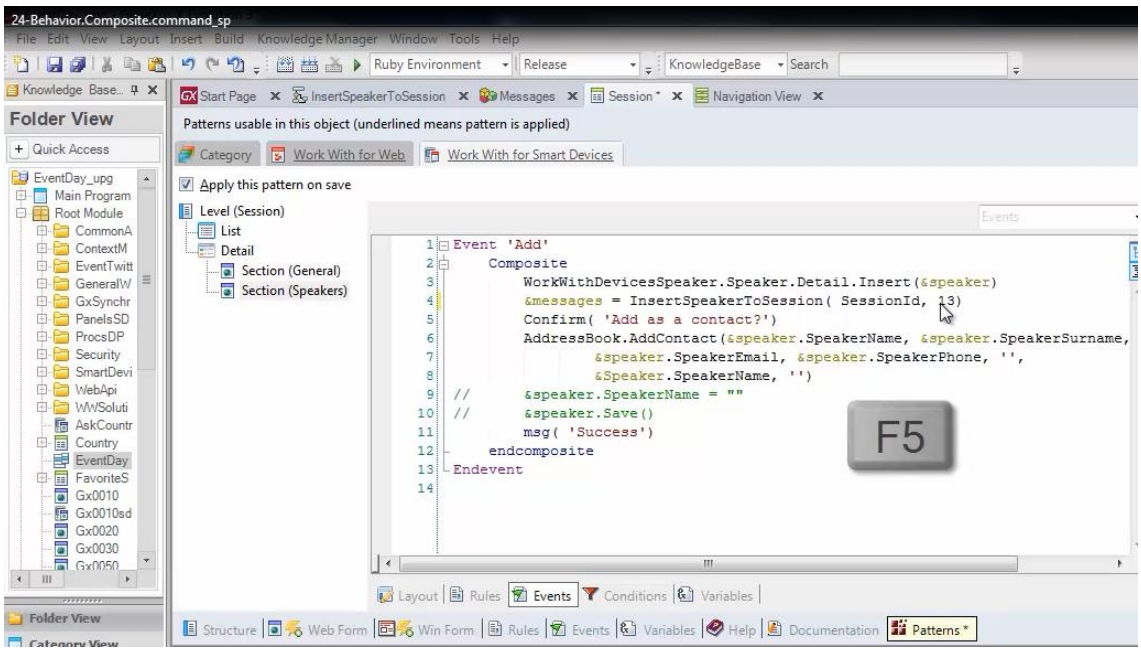


por ejemplo, para esta conferencia

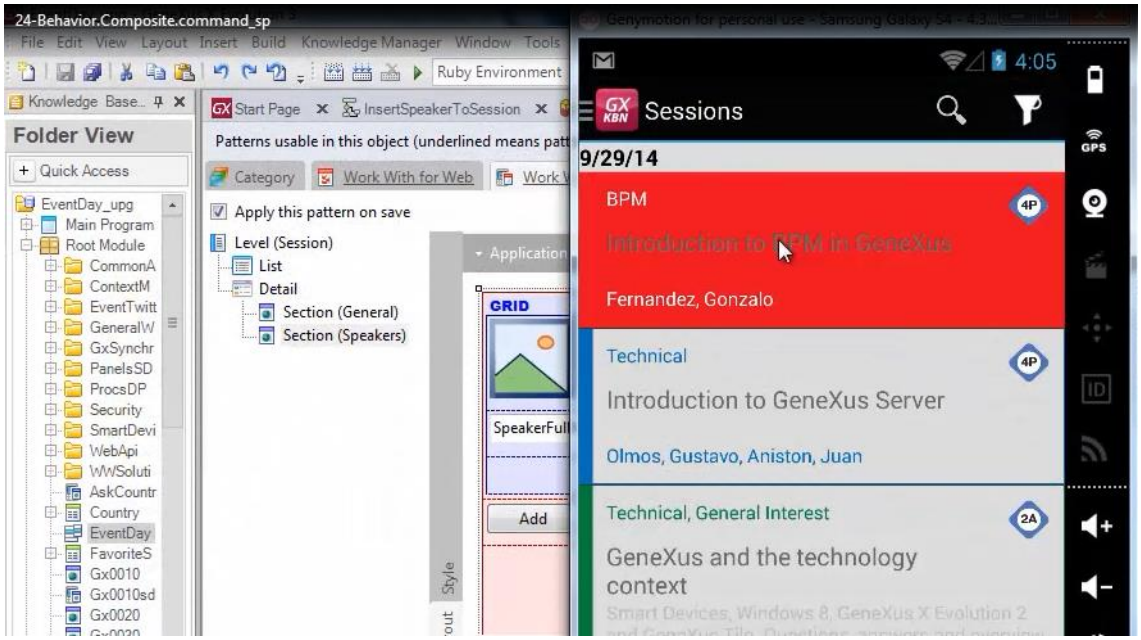


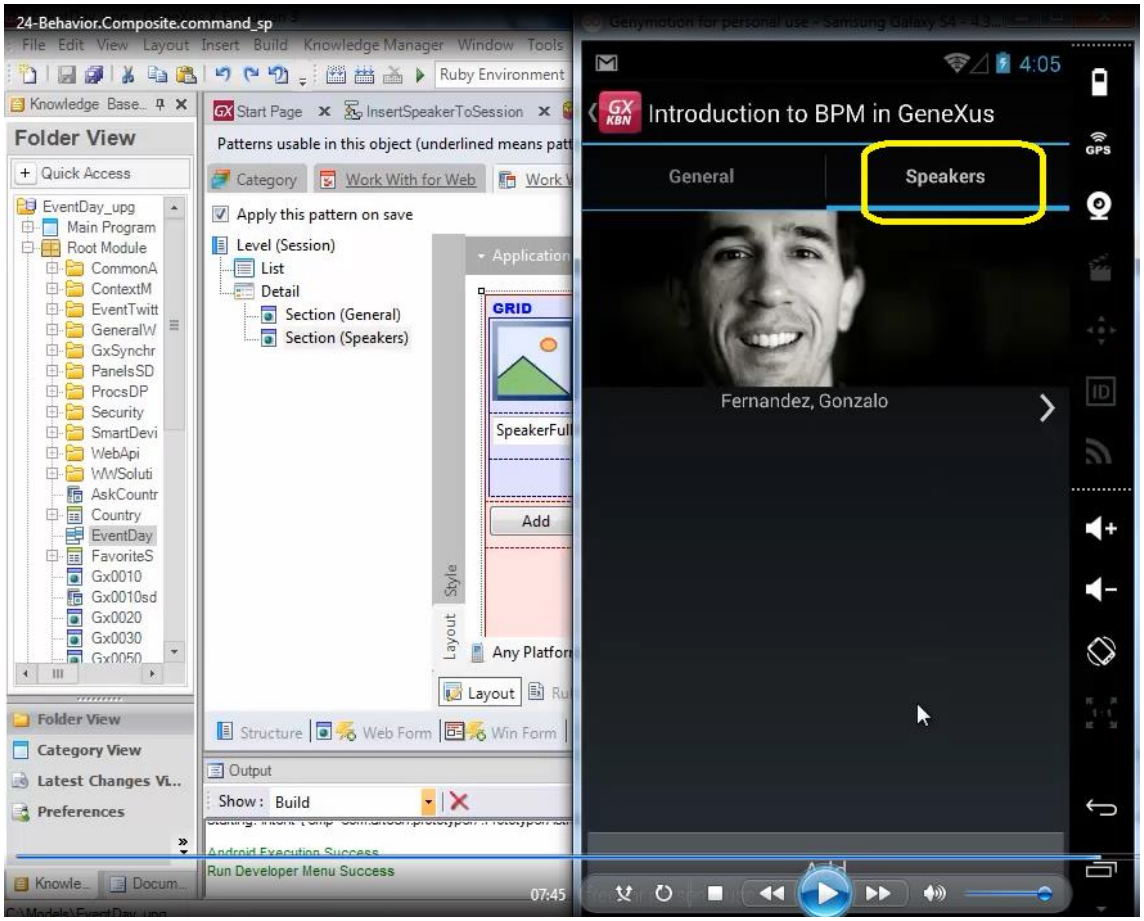


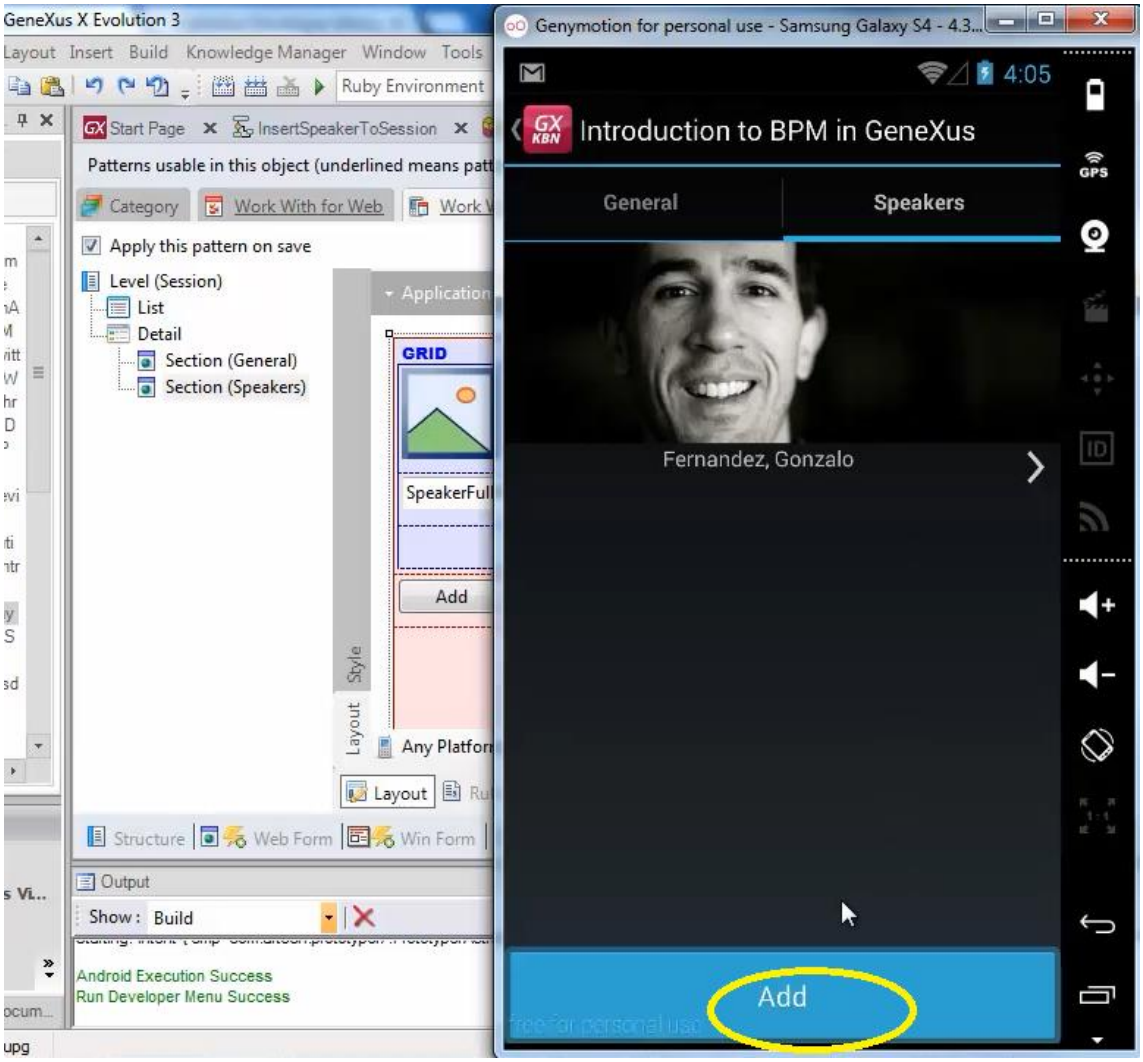
el SpeakerId=13



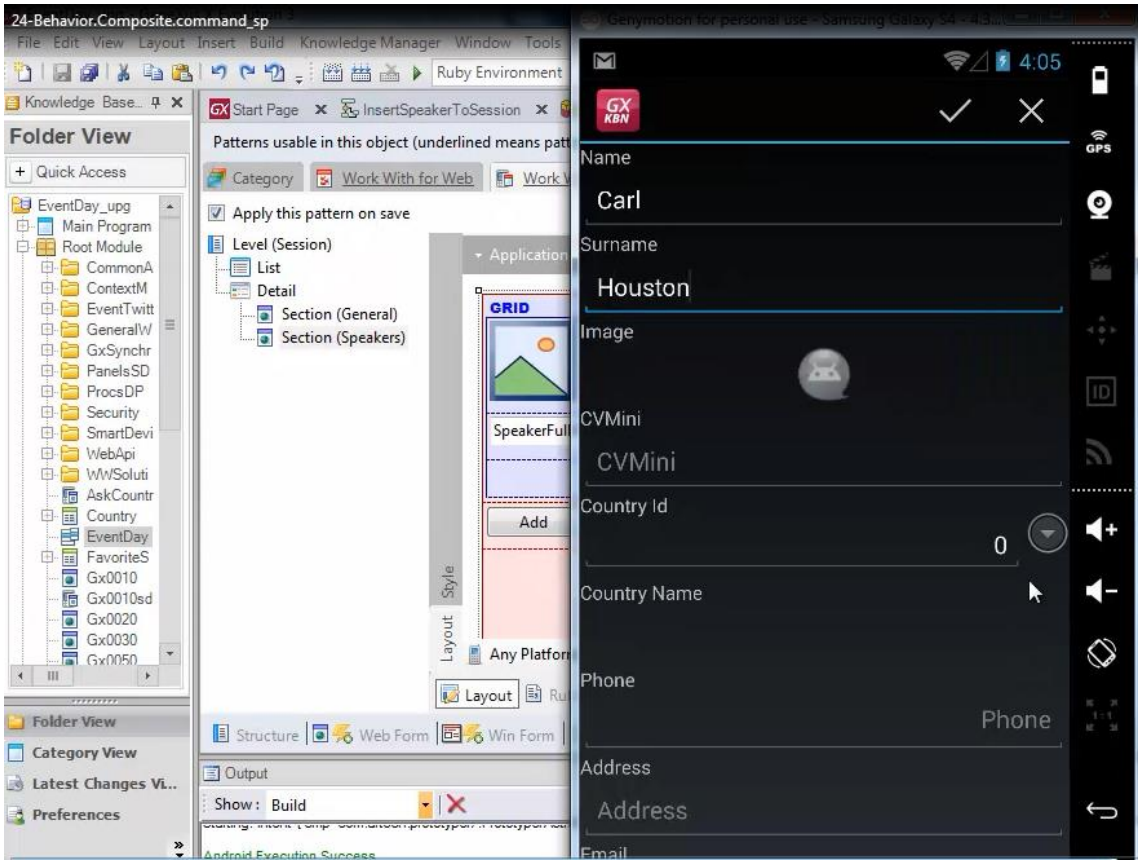
F5

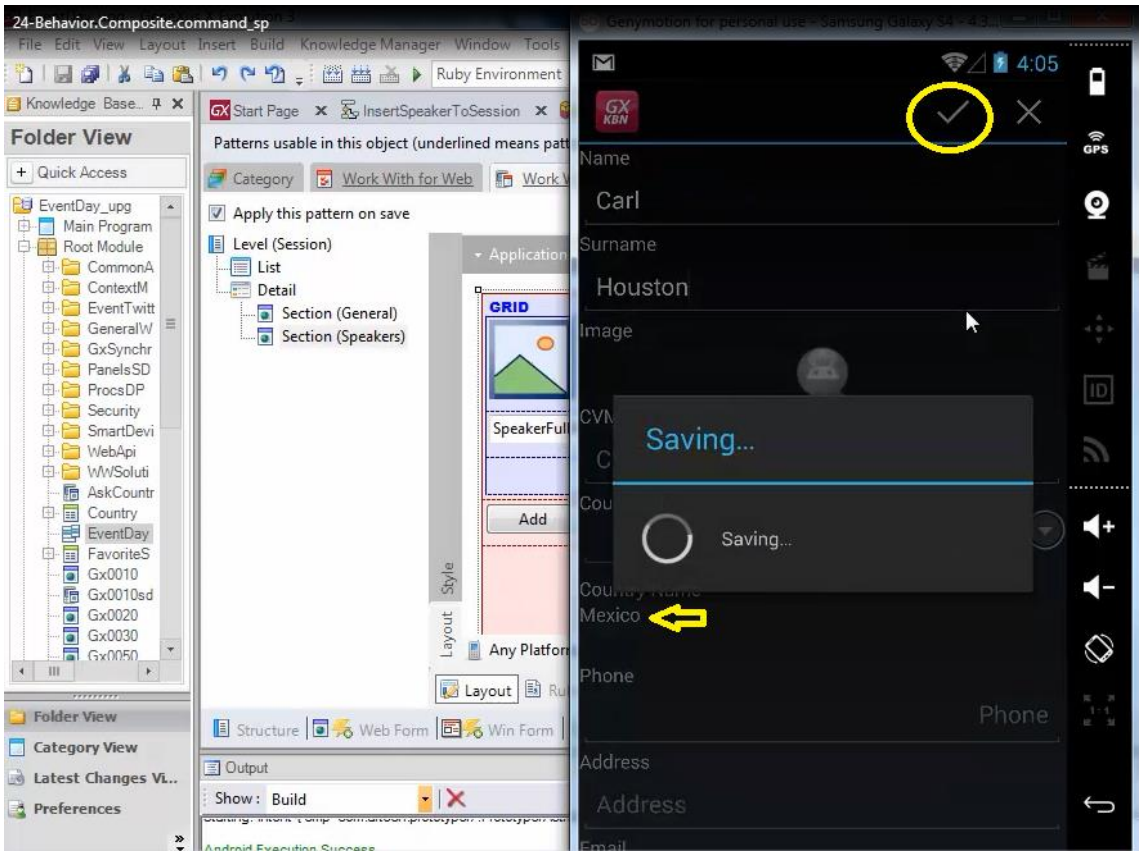

24-Behavior.Composite.command_sp
Folder View
Sessions
9/29/14
BPM
Introduction to BPM in GeneXus
Fernandez, Gonzalo
Technical
Introduction to GeneXus Server
Olmos, Gustavo, Aniston, Juan
Technical, General Interest
GeneXus and the technology context

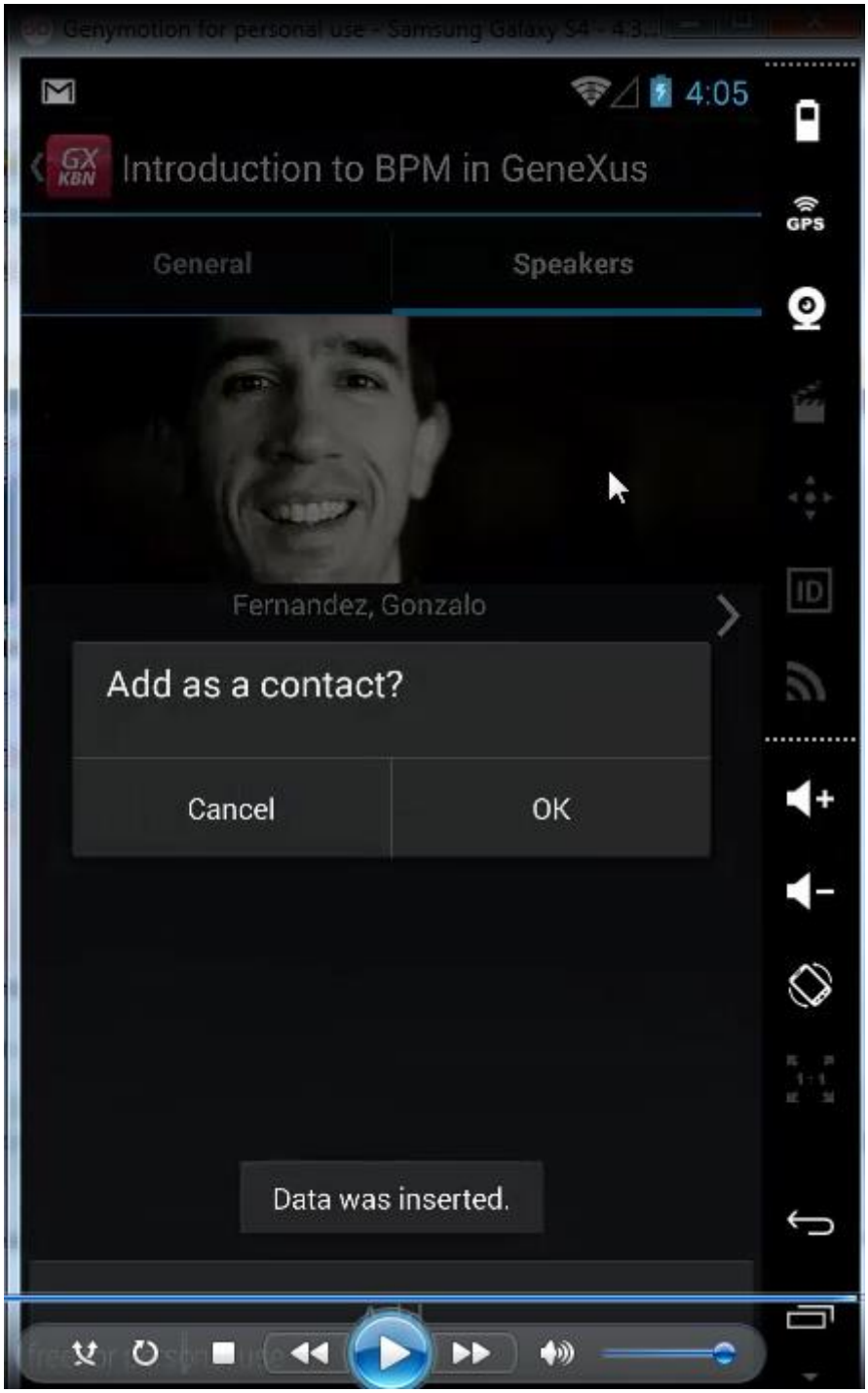

24-Behavior.Composite.command_sp
Folder View
Introduction to BPM in GeneXus
General
Speakers
Fernandez, Gonzalo
Run Developer Menu Success



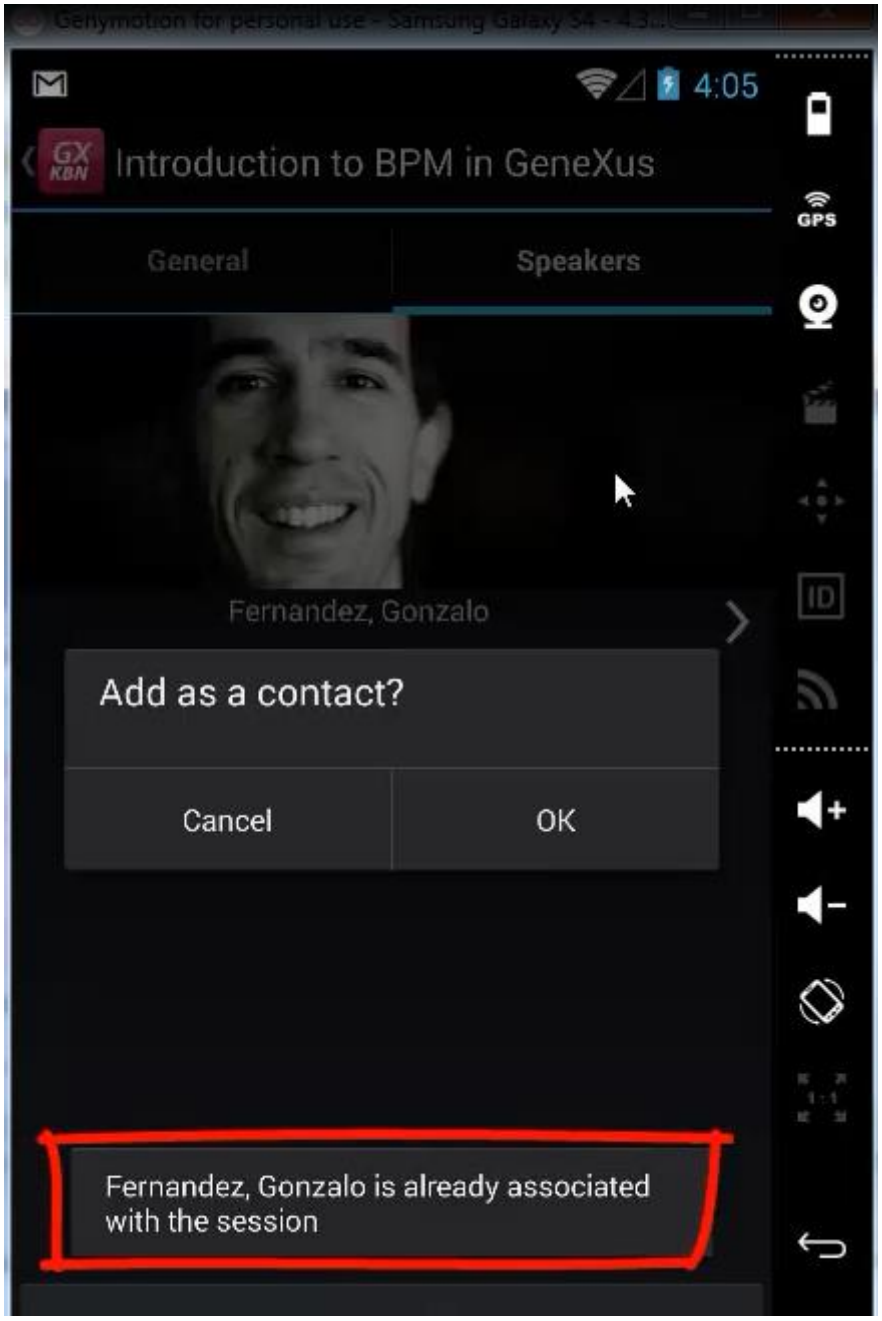
Video filmado con GeneXus X Evolution 3


24-Behavior.Composite.command_sp
Folder View
Quick Access
Start Page
InsertSpeakerToSession
Patterns usable in this object (underlined means patt
Category
Work With for Web
Apply this pattern on save
Level (Session)
List
Detail
Section (General)
Section (Speakers)
GRID
Add
Structure
Web Form
Win Form
Output
Show : Build
4:05
Name
Carl
Surname
Houston
Image
CVMini
Country Id
0
Country Name
Phone
Address
Email

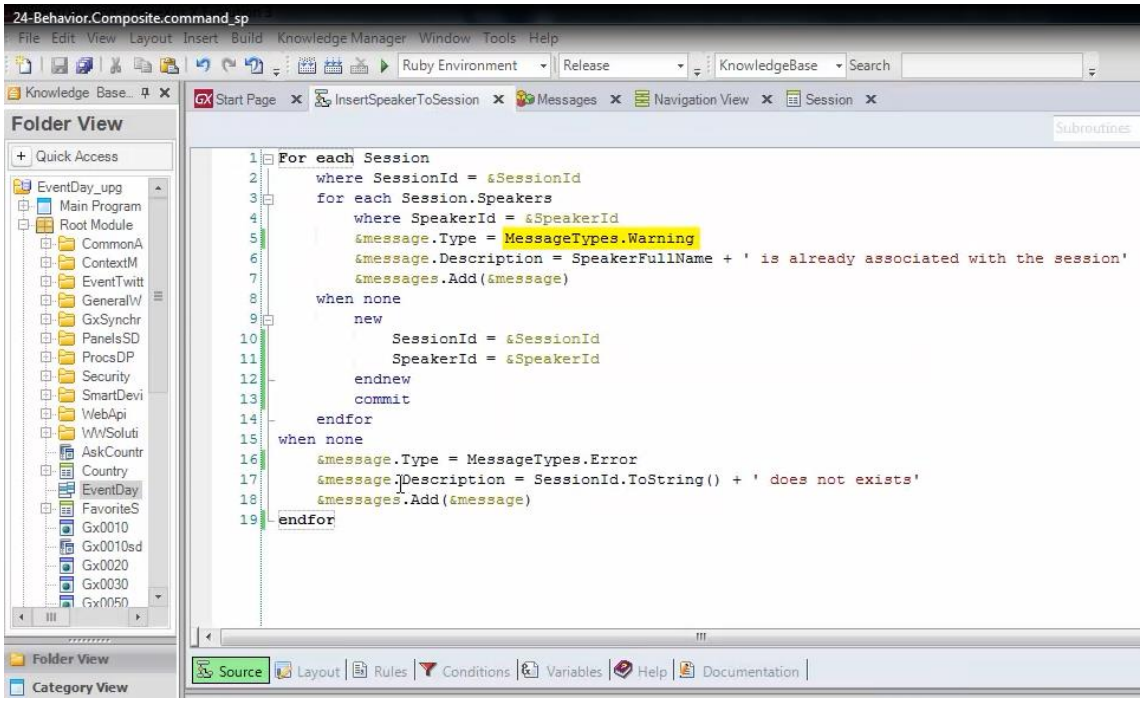

24-Behavior.Composite.command_sp
Folder View
Quick Access
Start Page
InsertSpeakerToSession
Patterns usable in this object (underlined means patt
Category
Work With for Web
Apply this pattern on save
Level (Session)
List
Detail
Section (General)
Section (Speakers)
GRID
Add
Structure
Web Form
Win Form
Output
Show : Build
4:05
Name
Carl
Surname
Houston
Image
Saving...
Saving...
Mexico
Phone
Address
Email

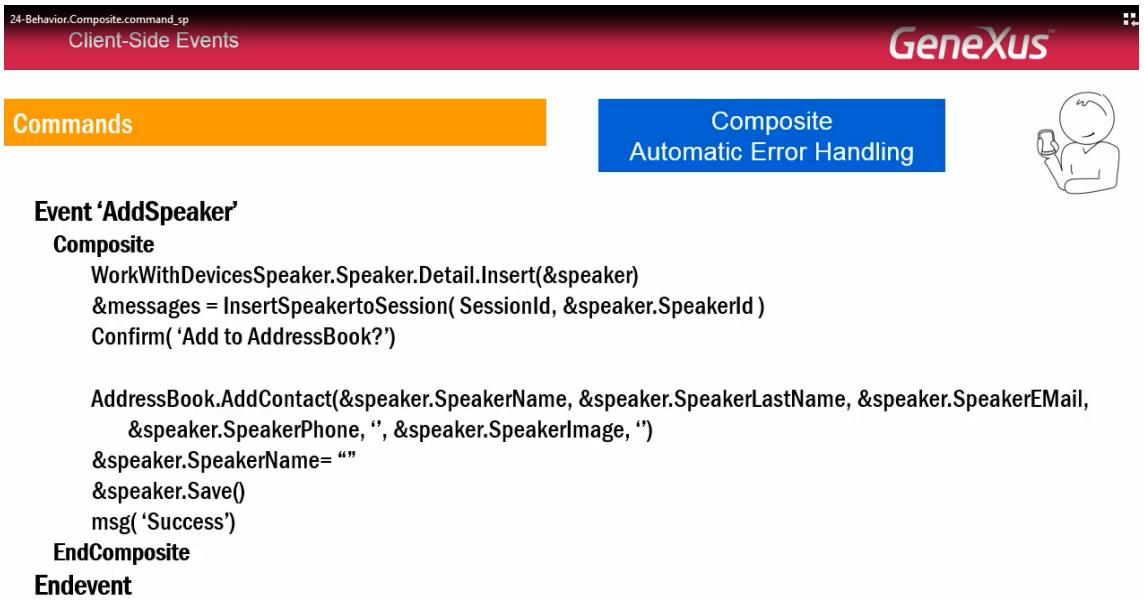


Video filmado con GeneXus X Evolution 3

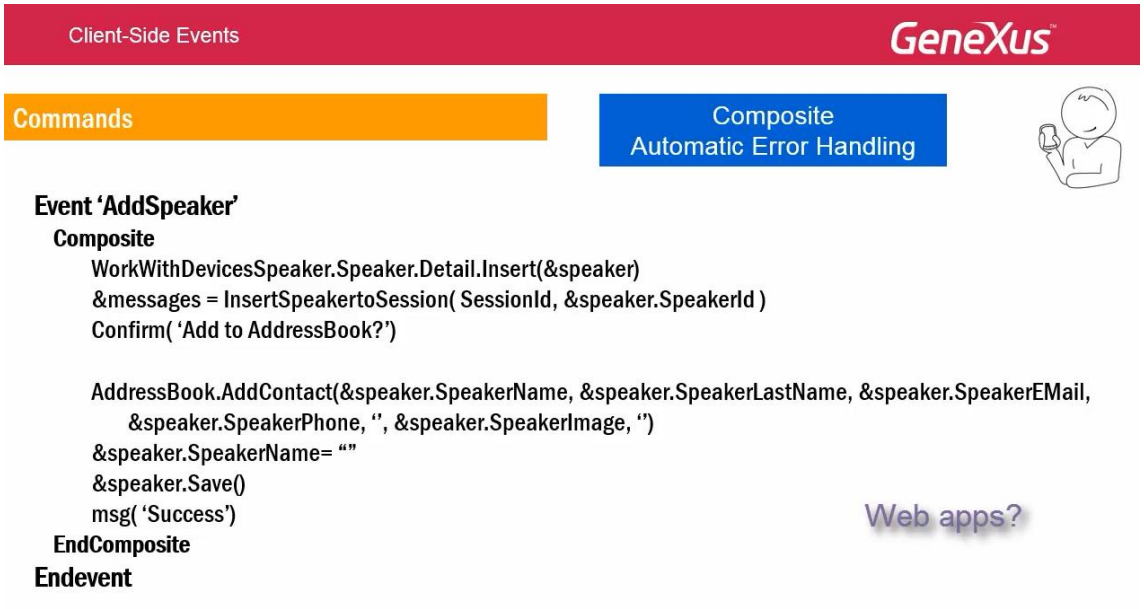


vemos que nos sale a modo de mensaje, pero no se interrumpe la ejecución, puesto que de ese tipo era el ítem de la colección



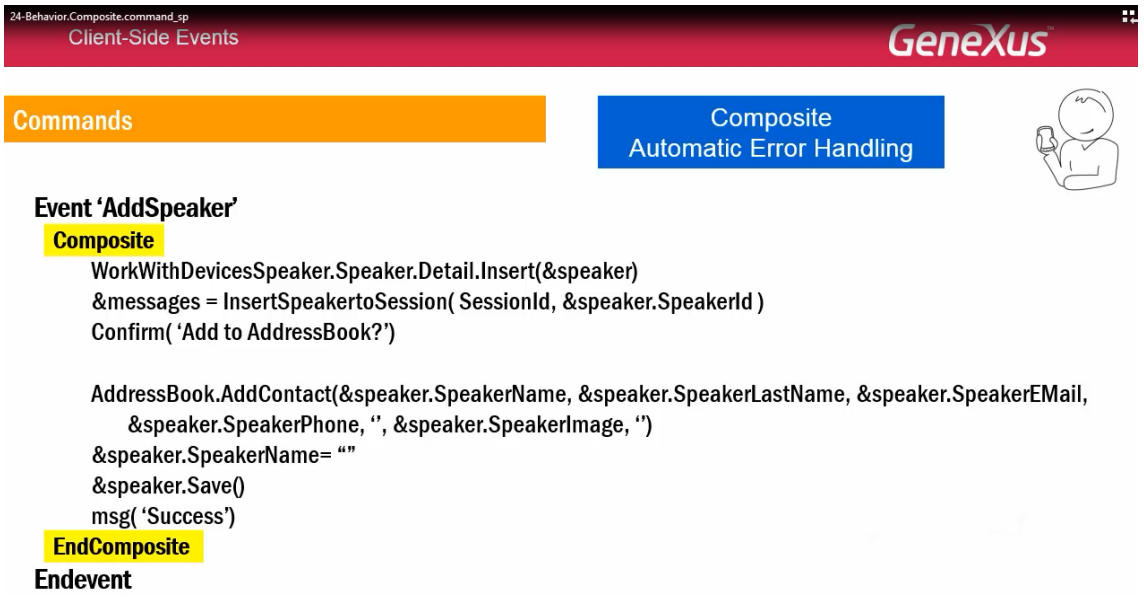


Estamos viendo en definitiva, la gran diferencia con las aplicaciones **web**

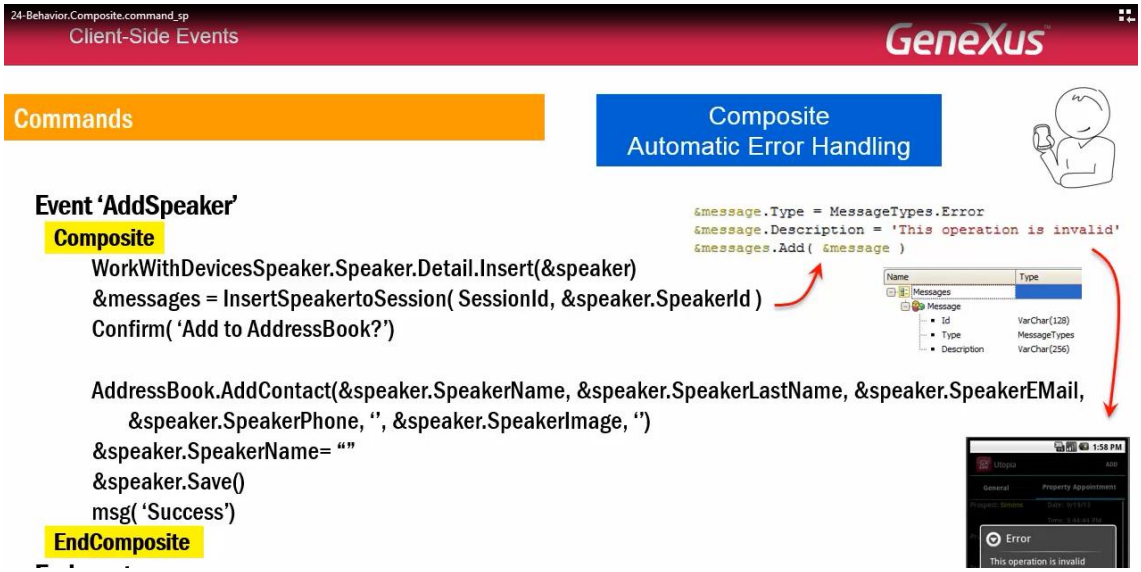


dado que en estas, cuando dentro de un evento un objeto llamado produce un error, no se interrumpe la ejecución; sigue en la sentencia siguiente.. y es el desarrollador el que debe encargarse de manejar los errores y programar las acciones a tomar.

El comando Composite en cambio

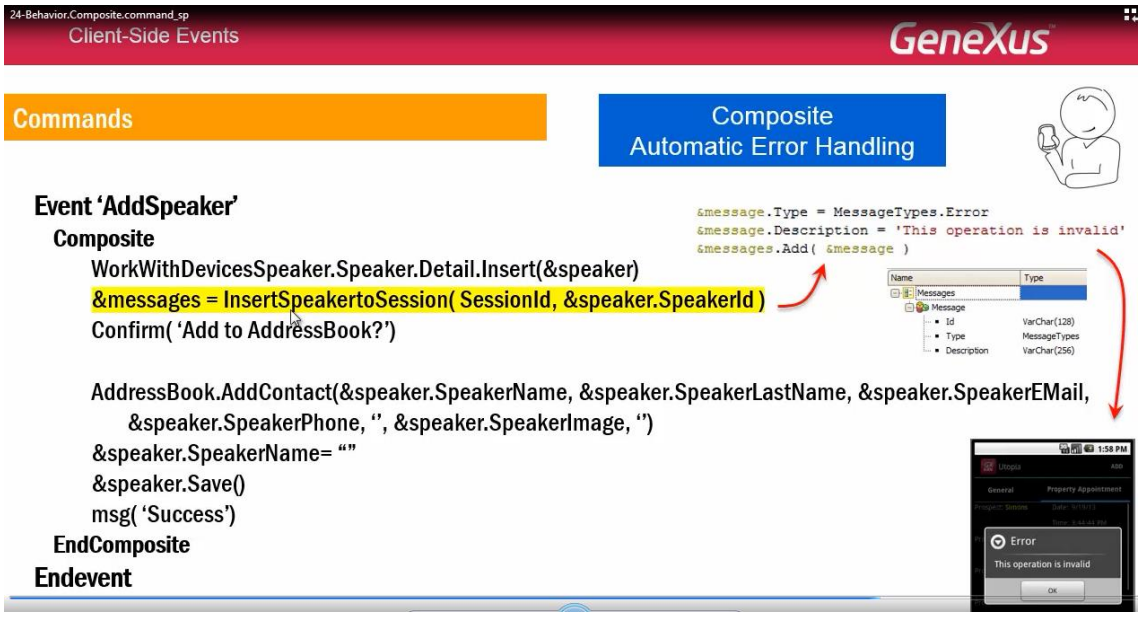


es el que hace que cuando ocurre un error en una secuencia de llamadas, se detenga la ejecución y se manejen los errores automáticamente, desplegándolos en la pantalla sin tener que implementar ninguna programación

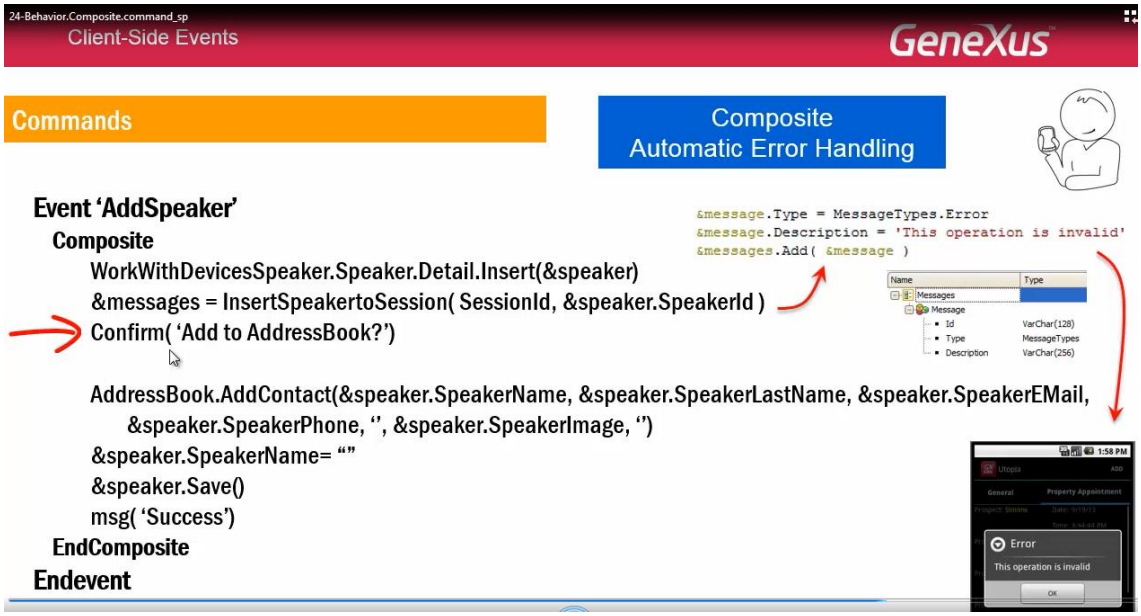


Este comando está implementado sólo en Smart Devices y es obligatorio en estos.

Por tanto, si la invocación al procedimiento

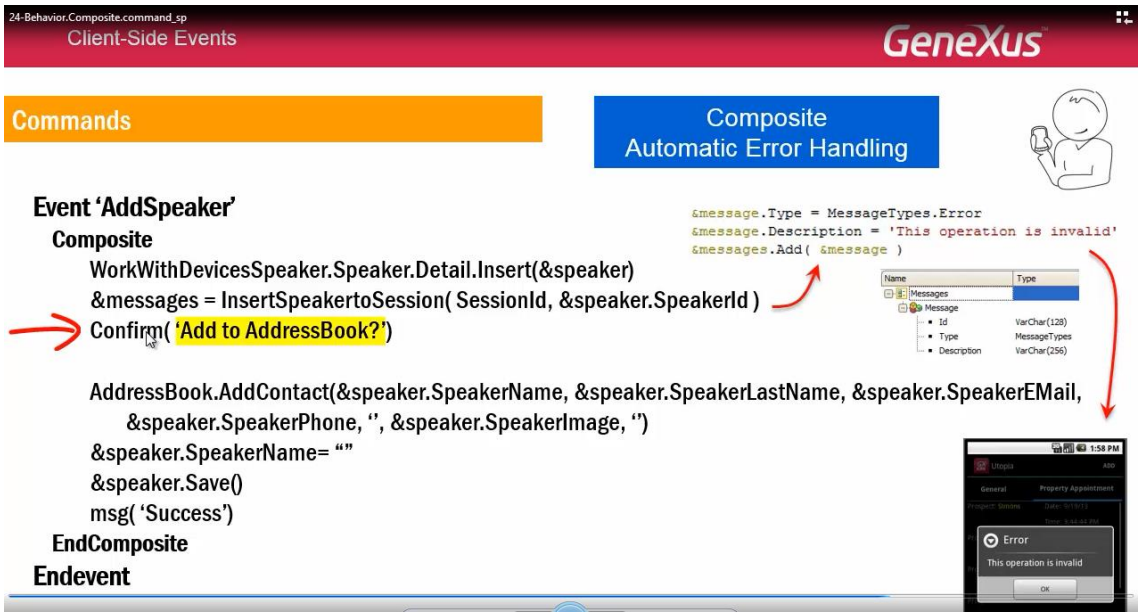


no arroja ningún error, se pasa a la invocación siguiente

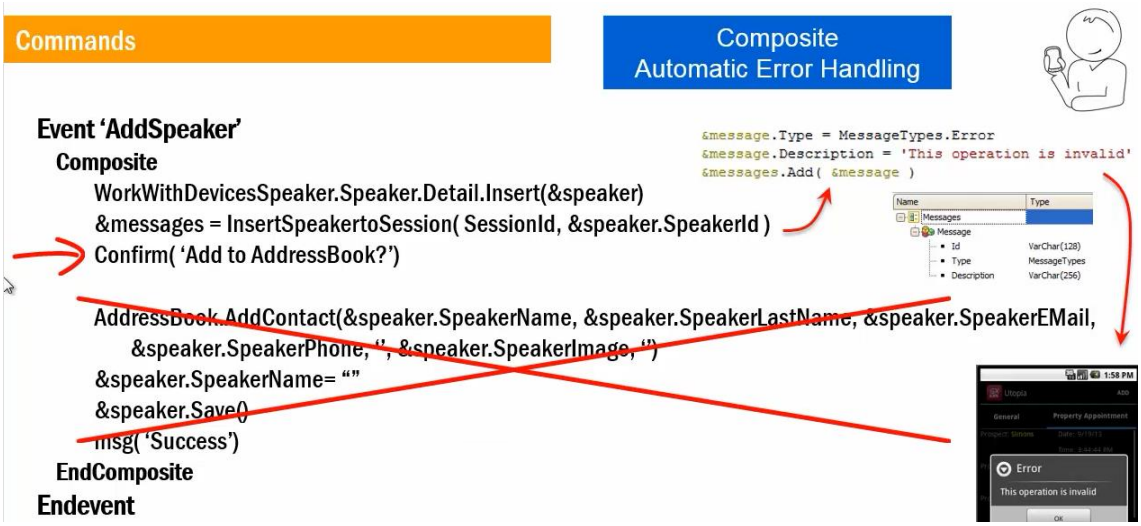


que es a la api: método confirm

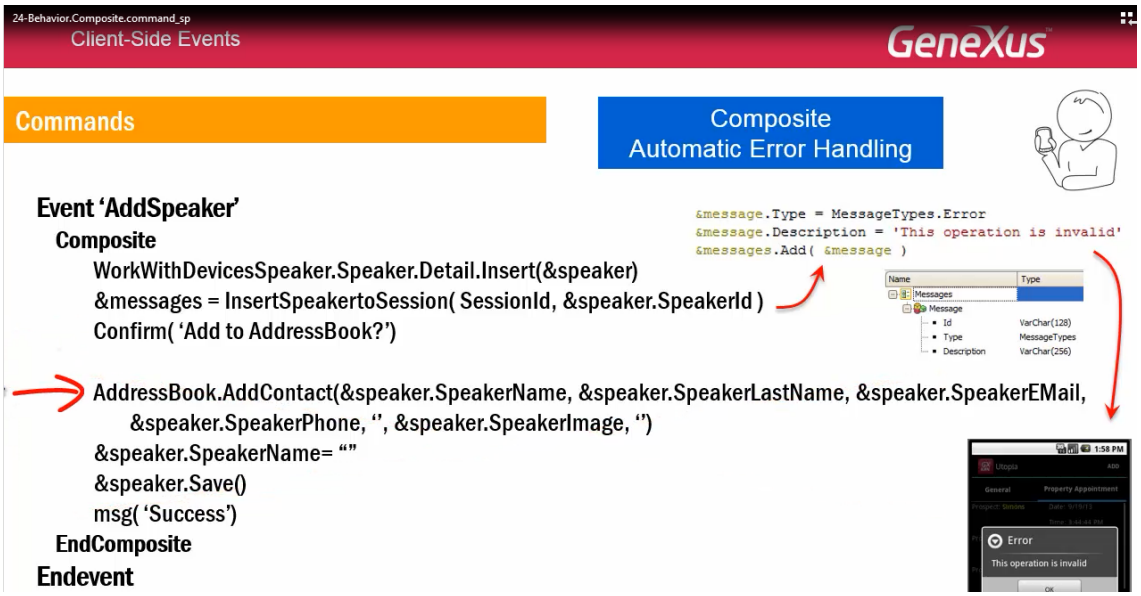
se despliega este mensaje al usuario



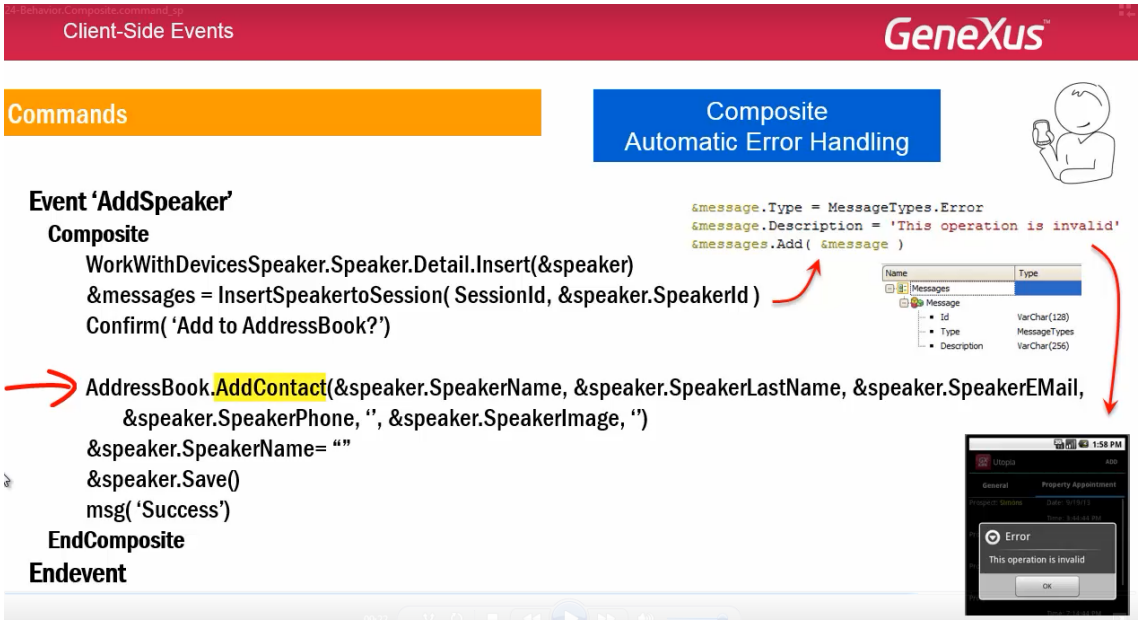
Si el usuario elije no confirmar, entonces se detiene la ejecución aquí y todo lo que sigue no se ejecuta..



Si por el contrario elije confirmar, entonces se invoca a la siguiente entrada

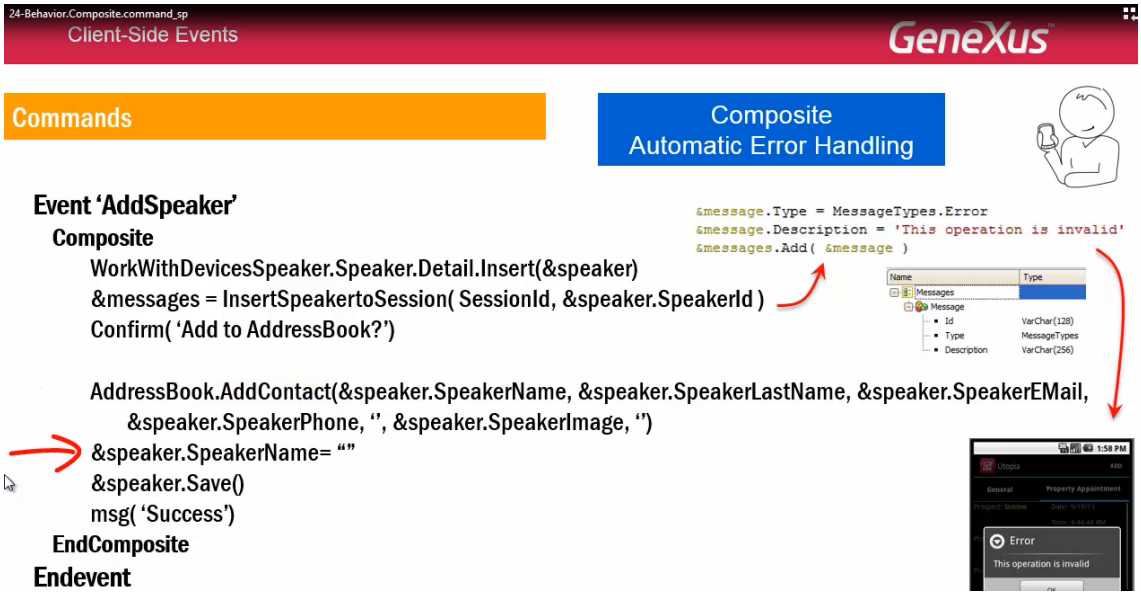


esto es a la api: AddressBook

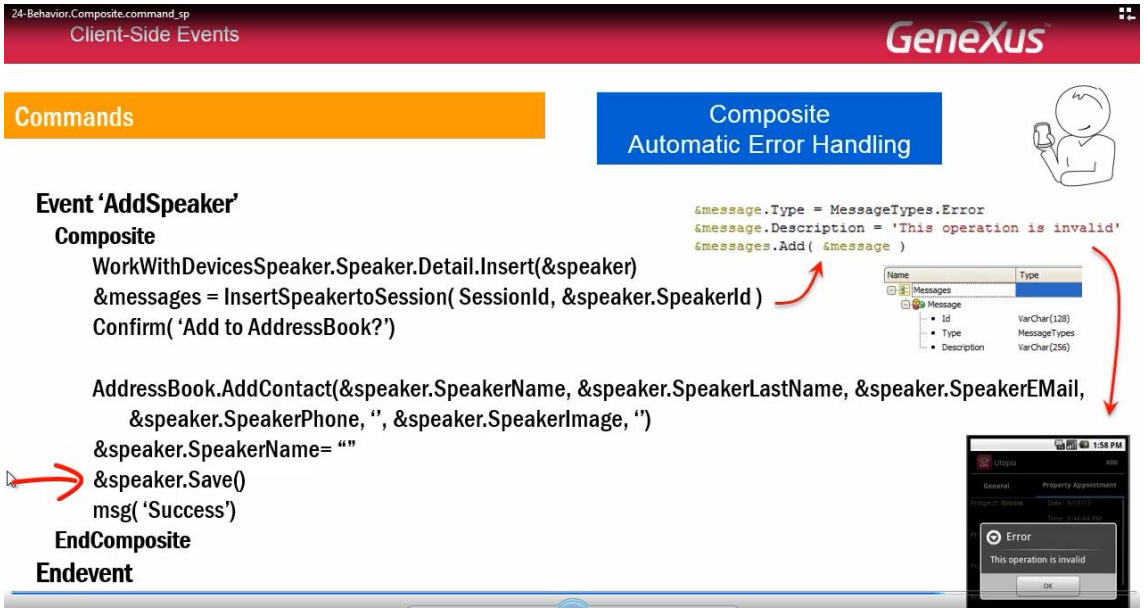


para agregar el contacto..

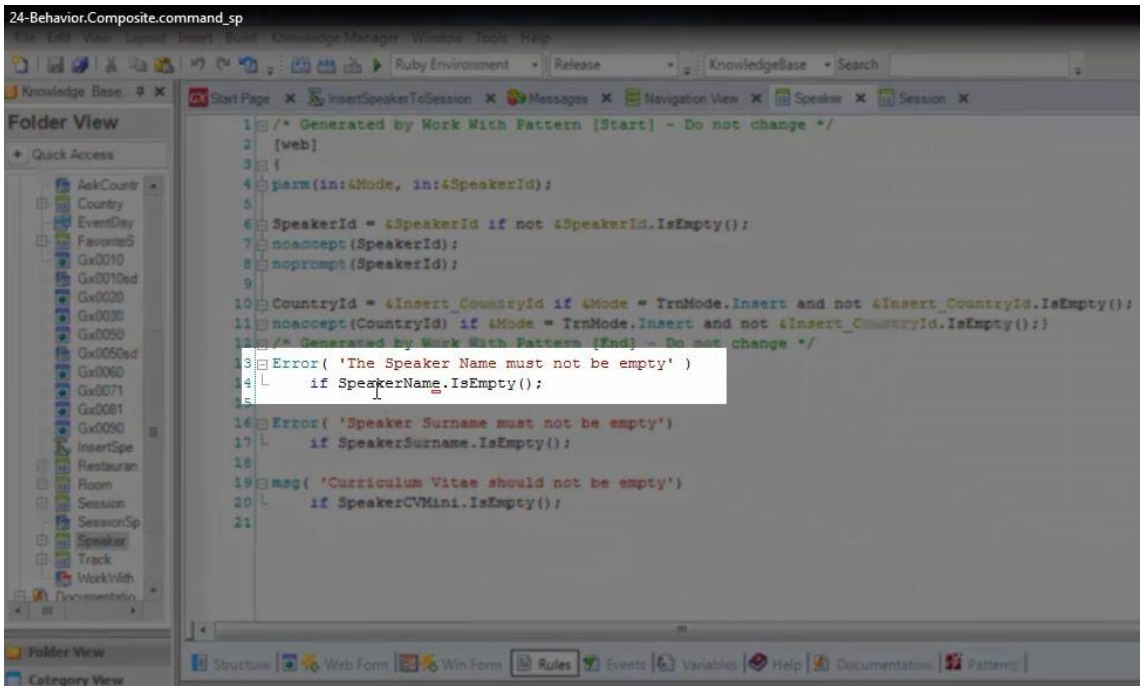
Otra vez, si esta operación es exitosa, se pasa a la siguiente sentencia



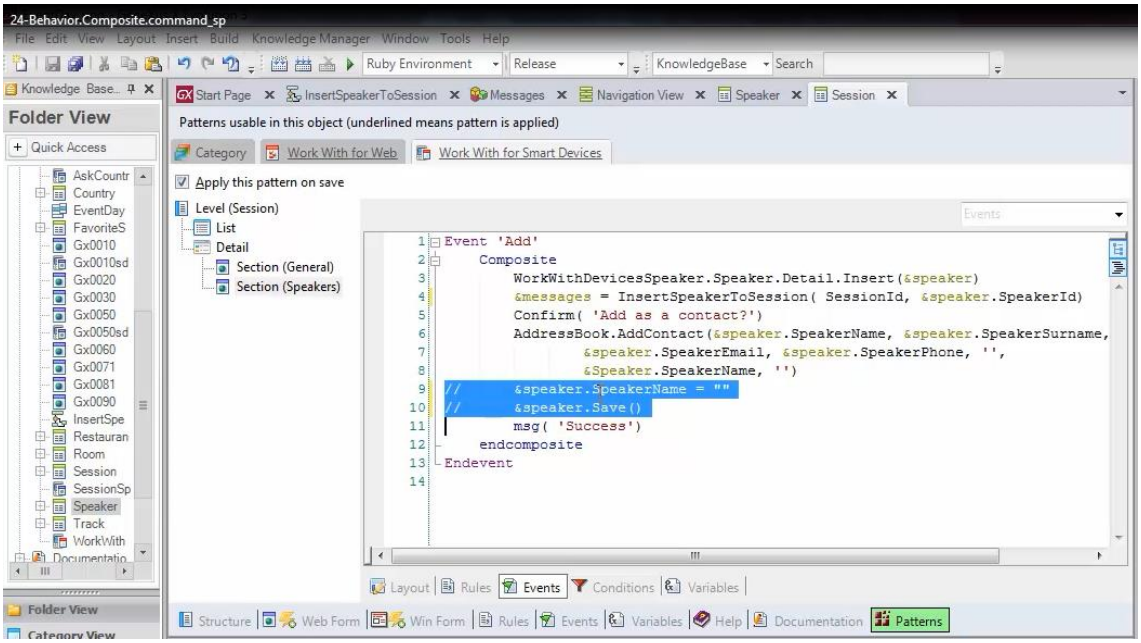
donde en este ejemplo estamos dejando vacío el SpeakerName del business component SPEAKER y estamos intentando salvar..



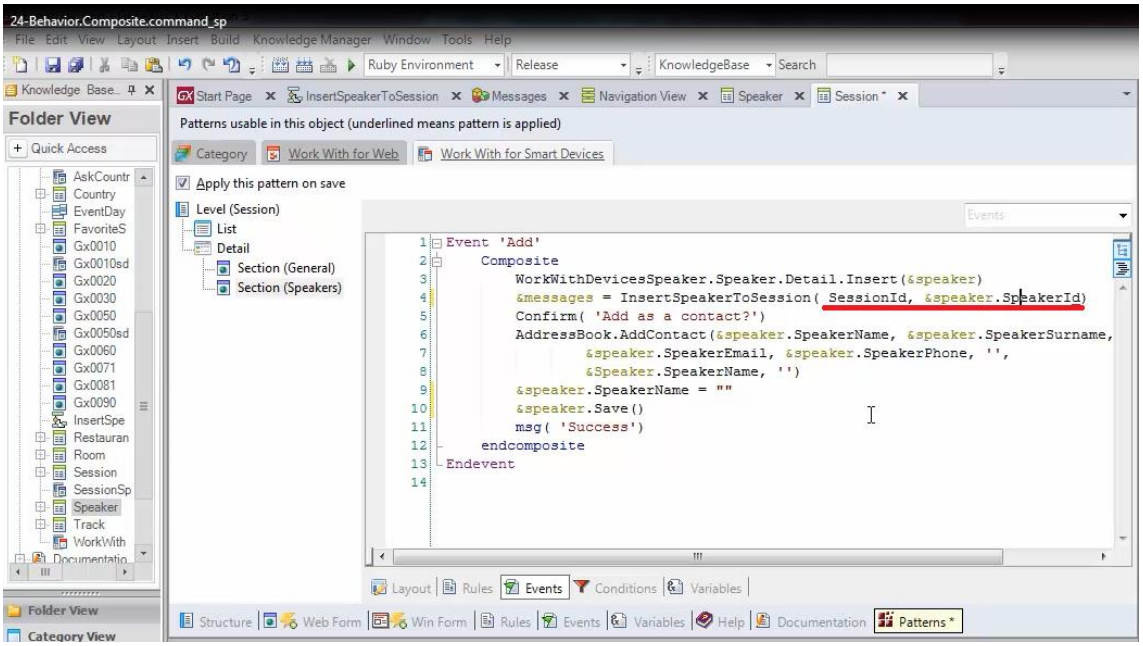
Esto va a arrojar un error puesto que si vamos a la transacción de Speakers vemos que teníamos programada la regla error para no dejar el nombre del orador vacío



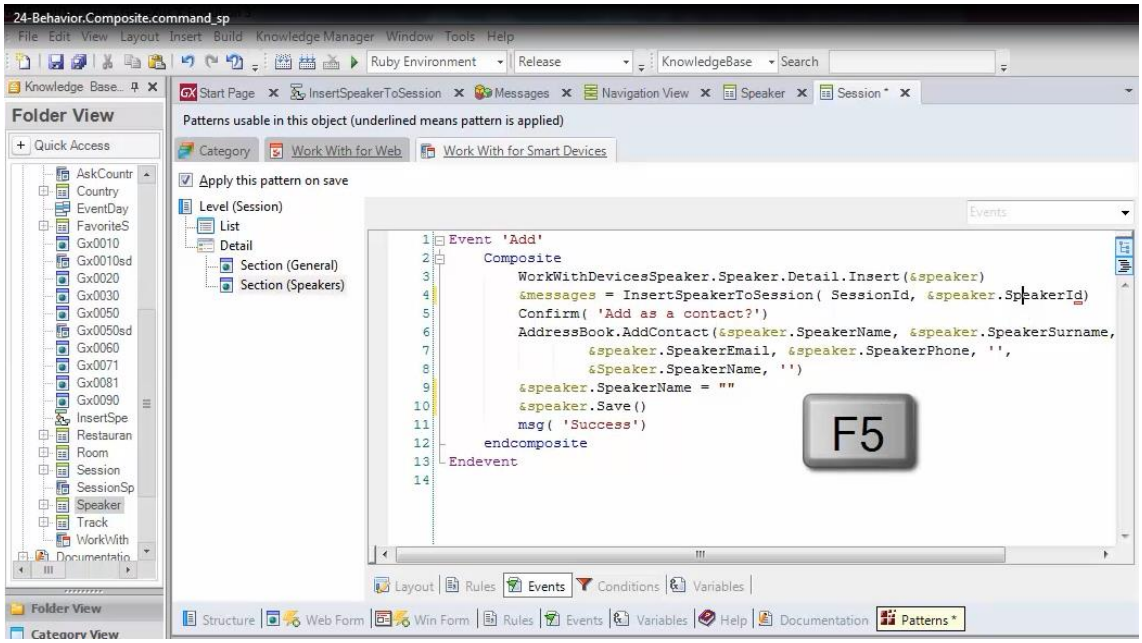
Por tanto, vamos a descomentar estas 2 cláusulas

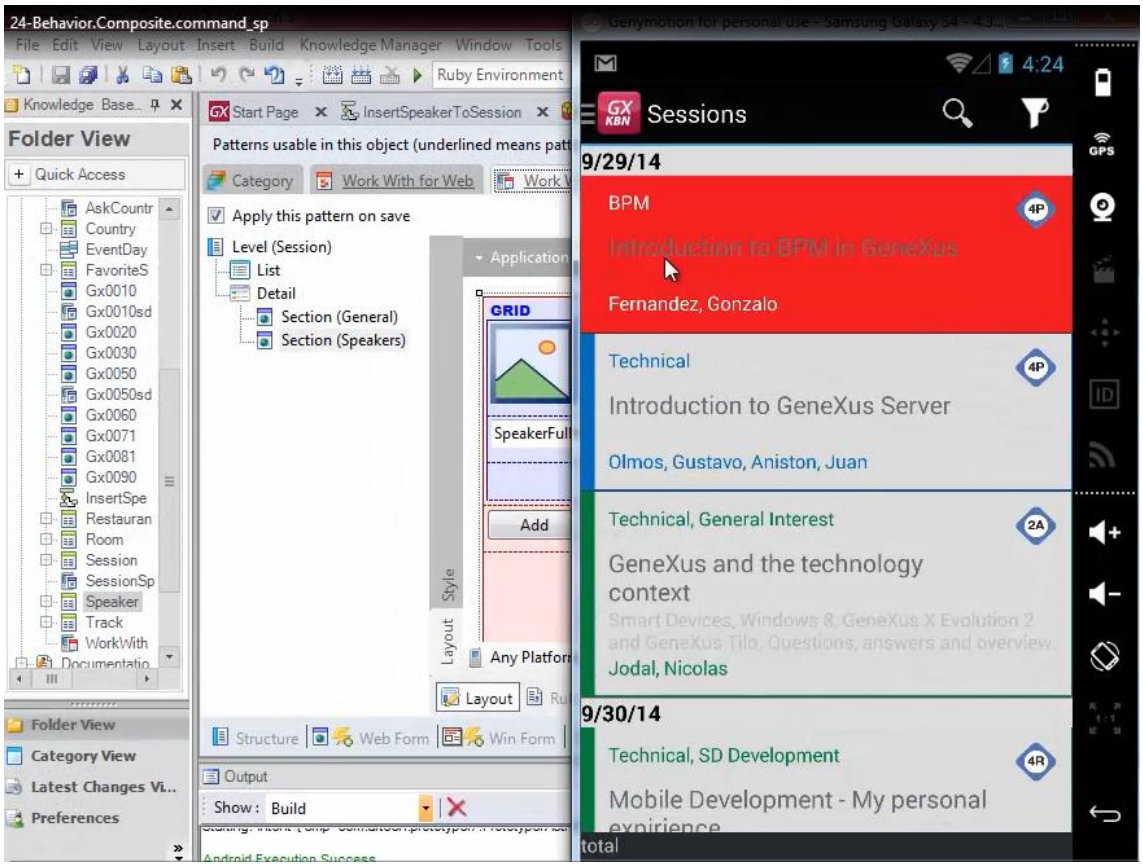


Observemos que restituimos aquí los parámetros correspondientes

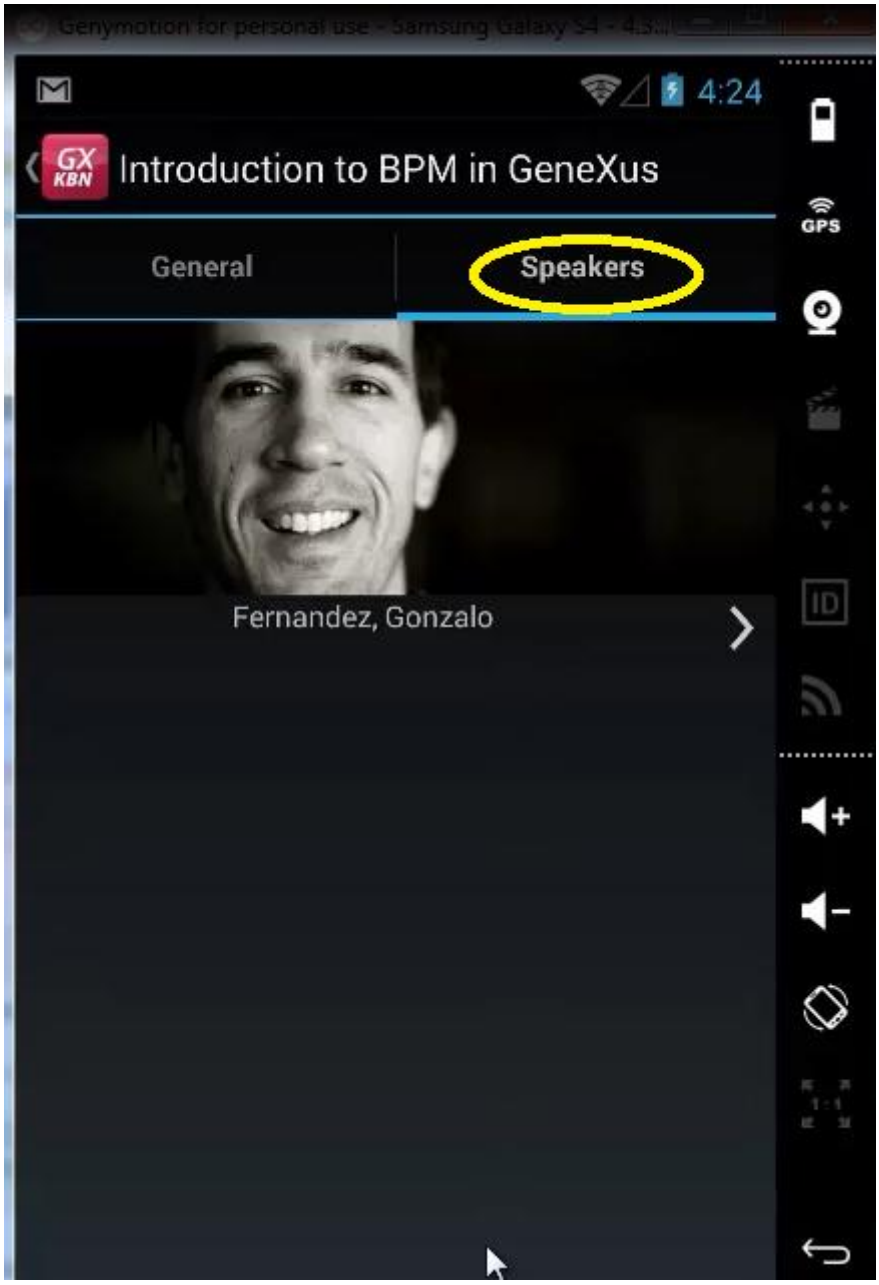


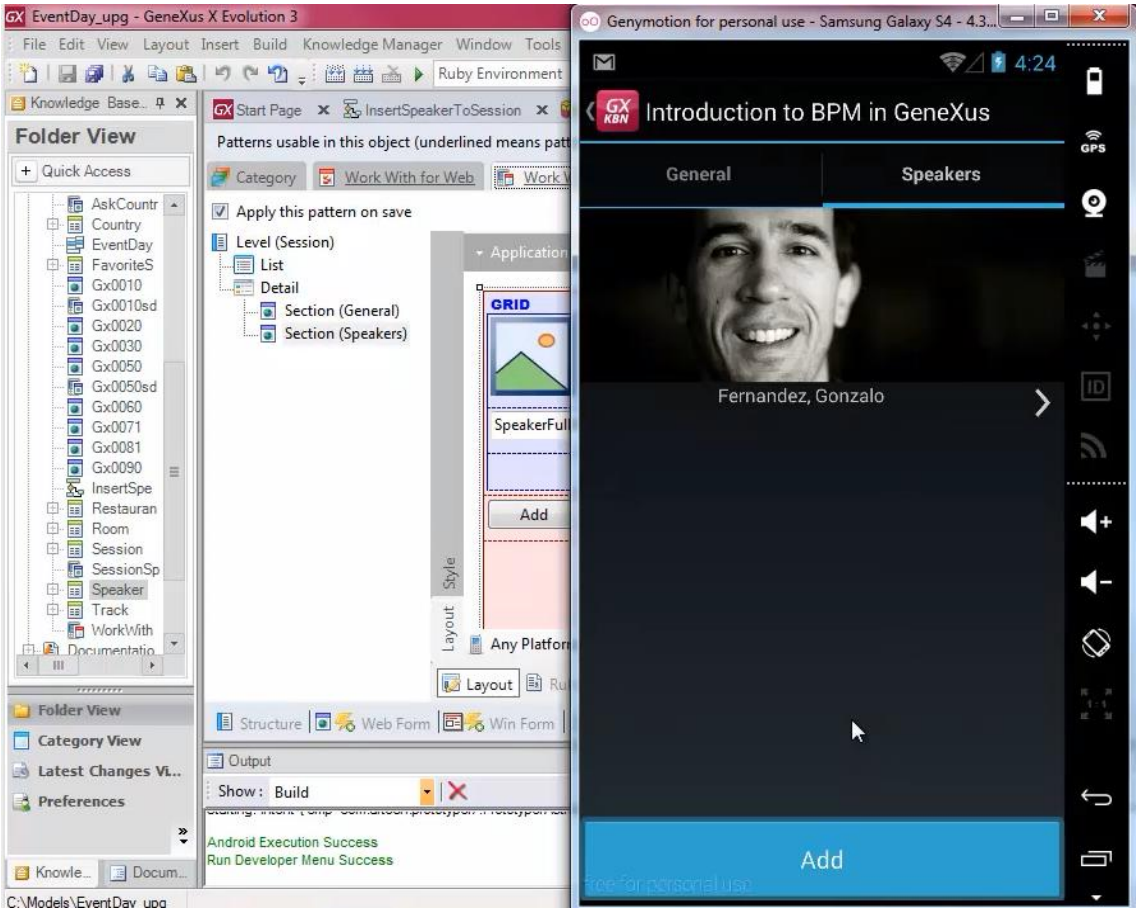
Y hagamos F5



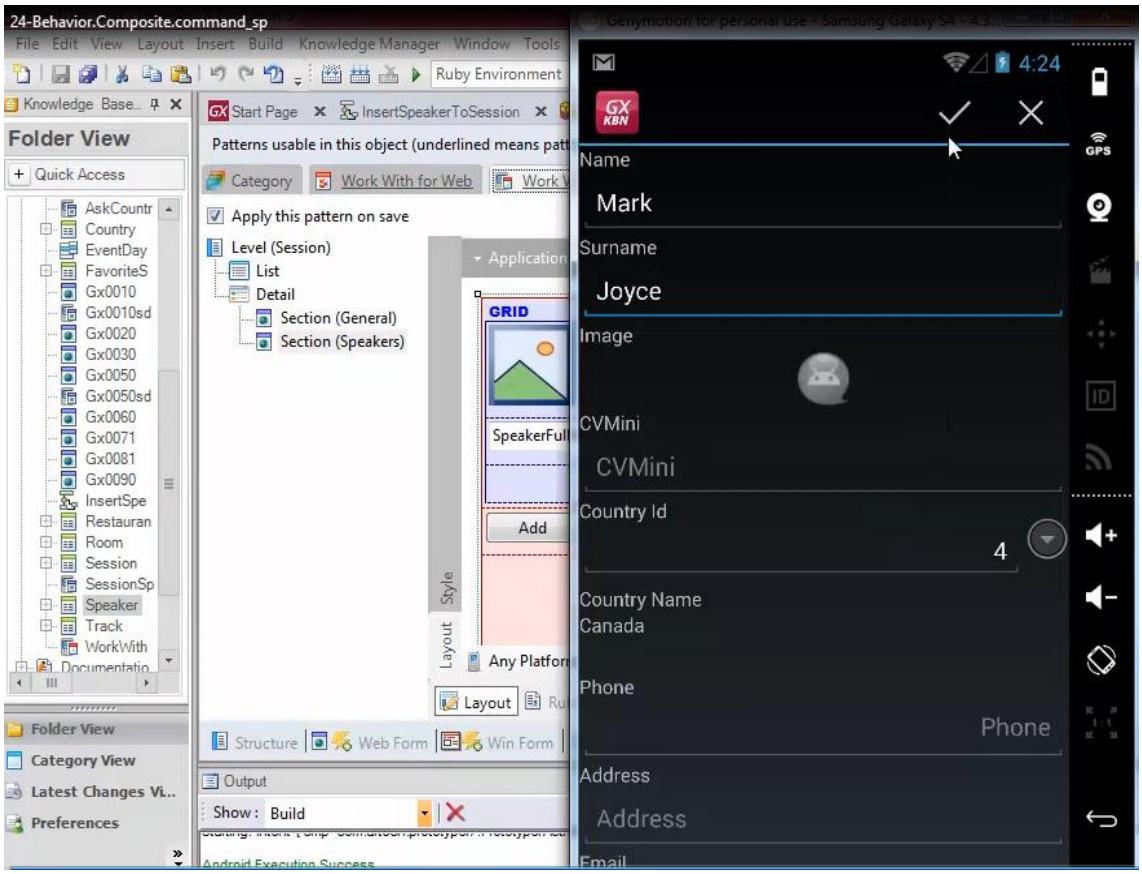


Vamos a la session

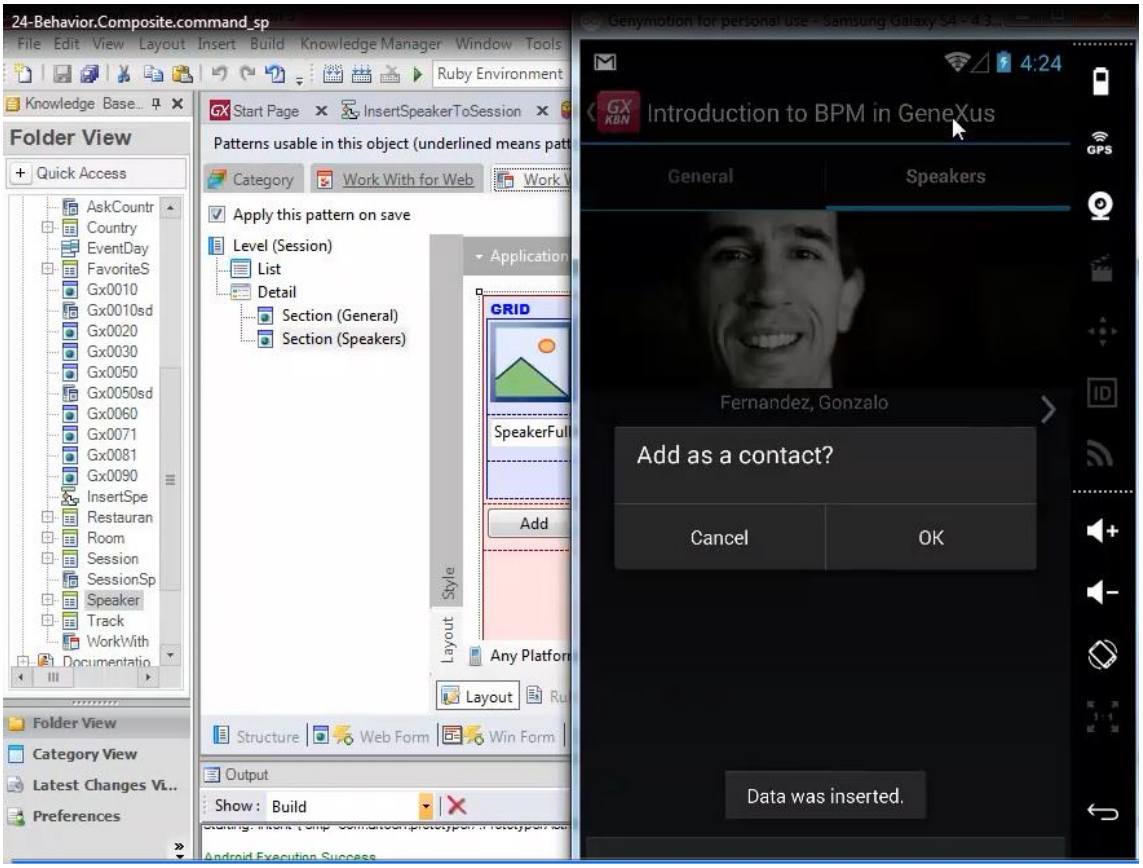

4:24
Introduction to BPM in GeneXus
General
Speakers
Fernandez, Gonzalo
GPS

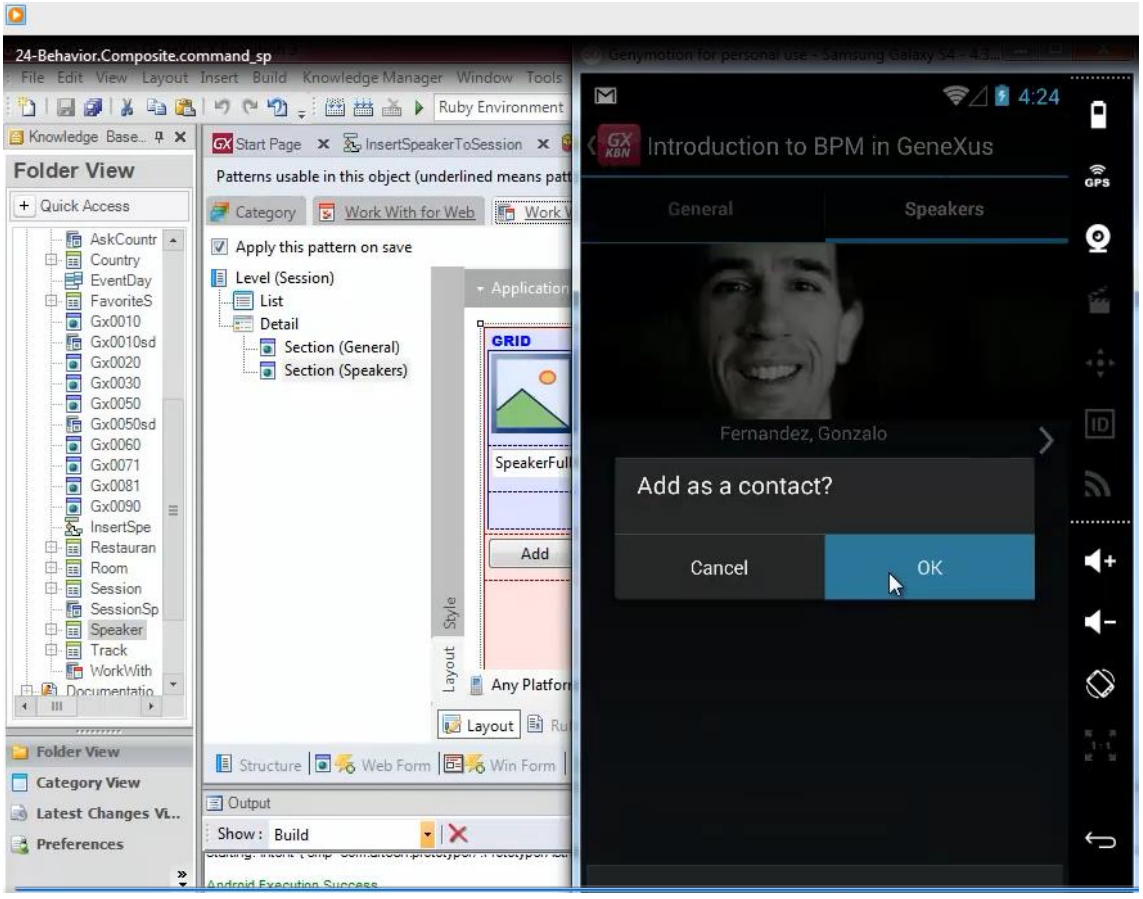


Agregar un orador…

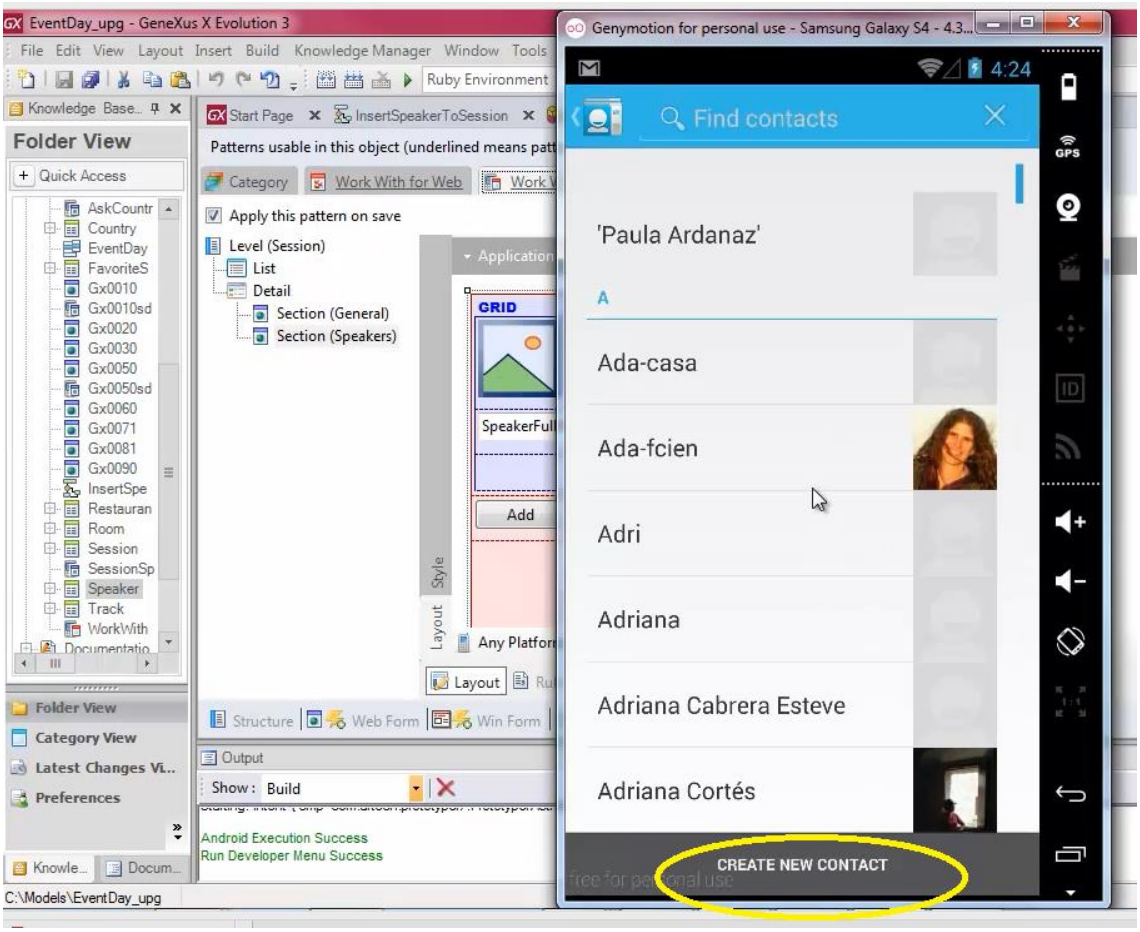


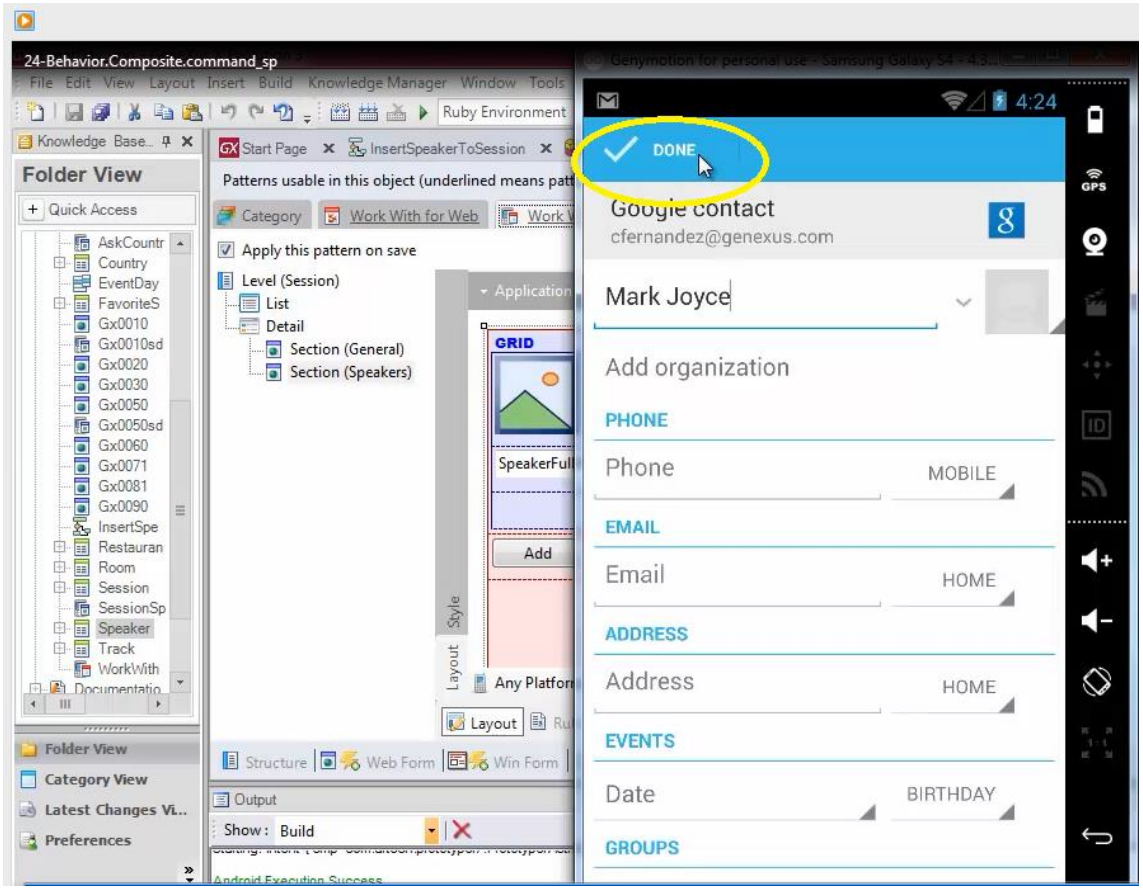
Video filmado con GeneXus X Evolution 3



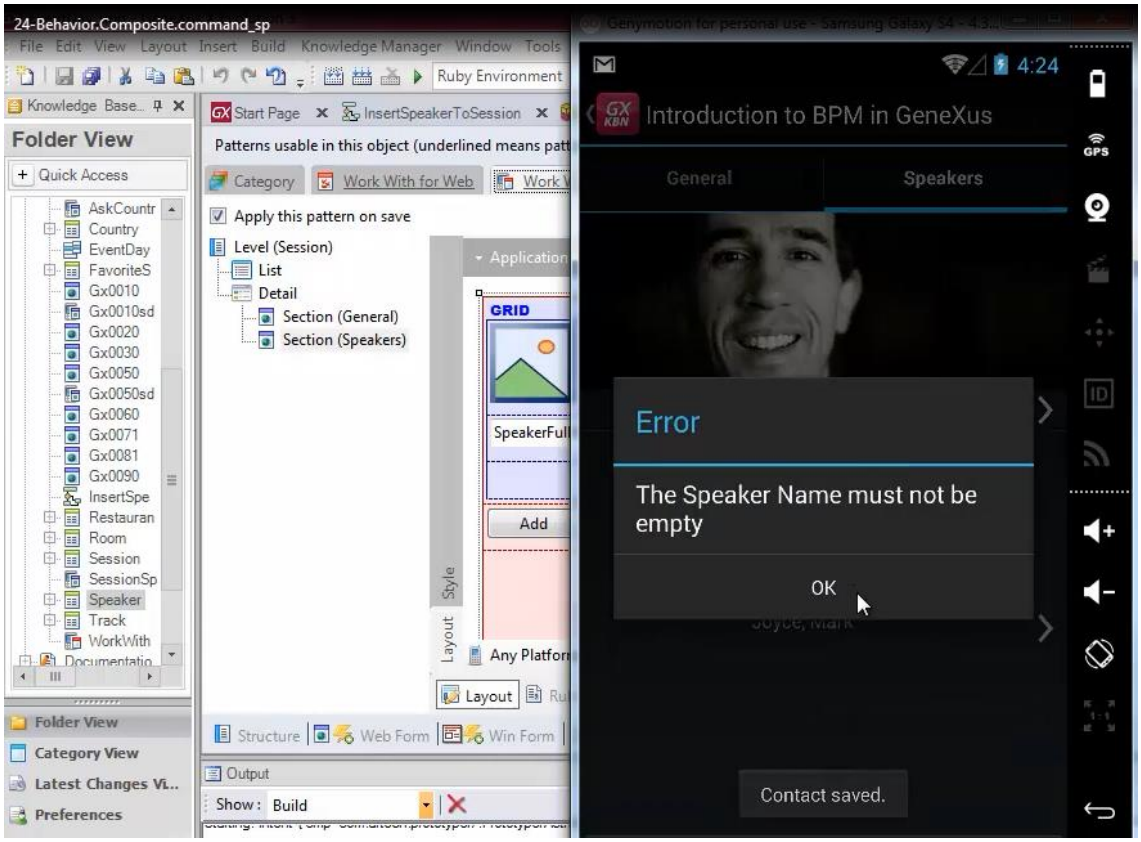


Agregamos a la libreta de direcciones..



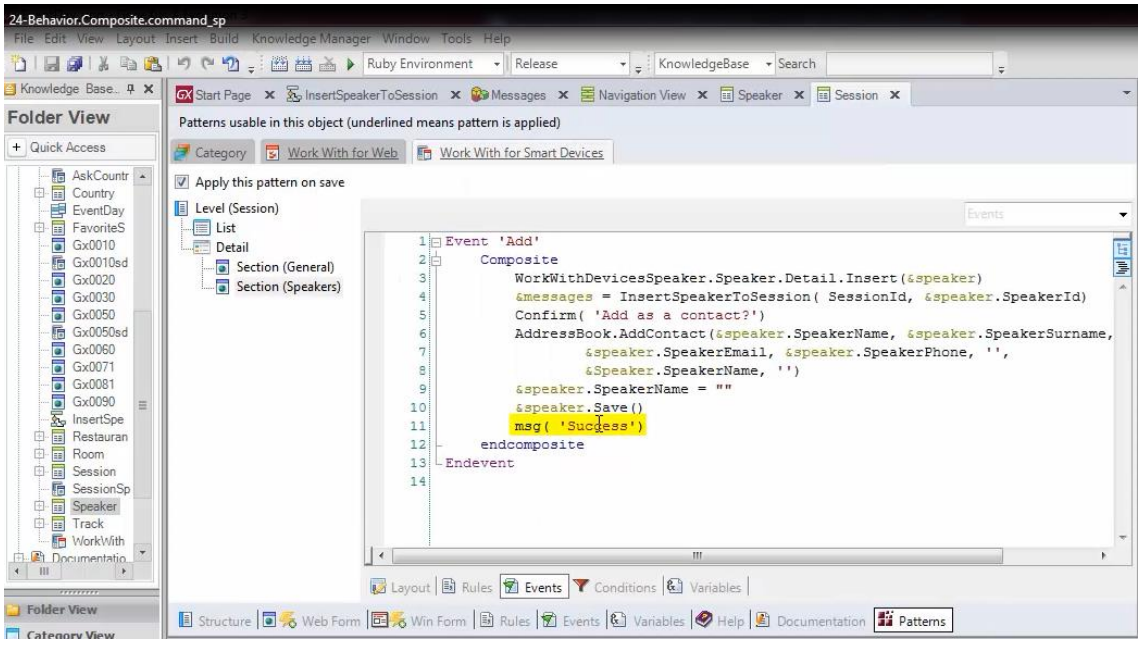


Video filmado con GeneXus X Evolution 3

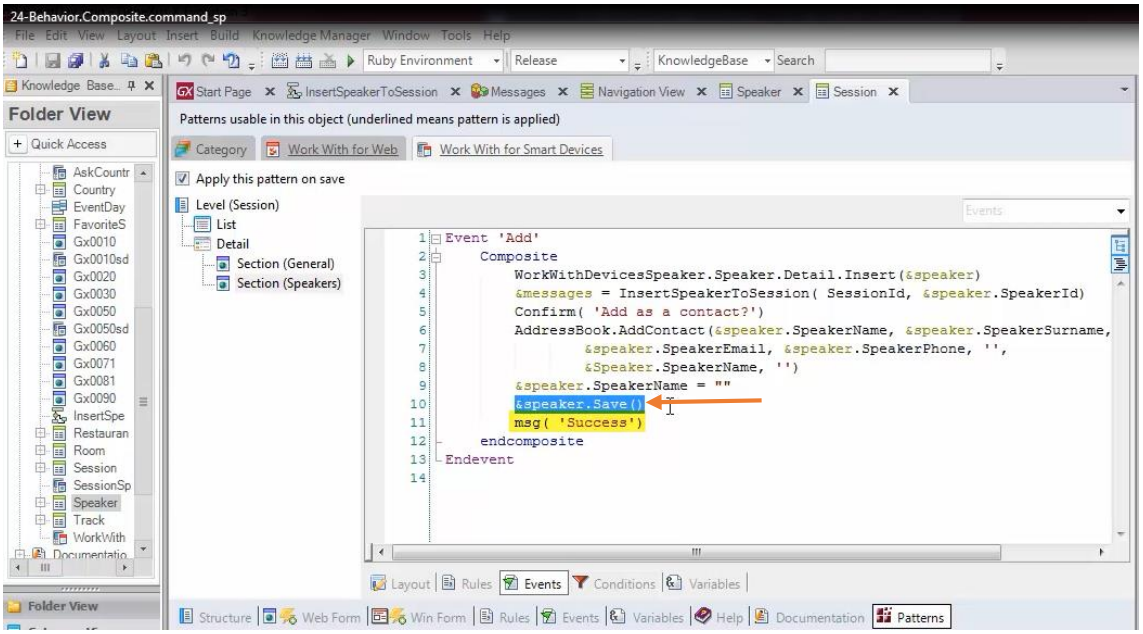


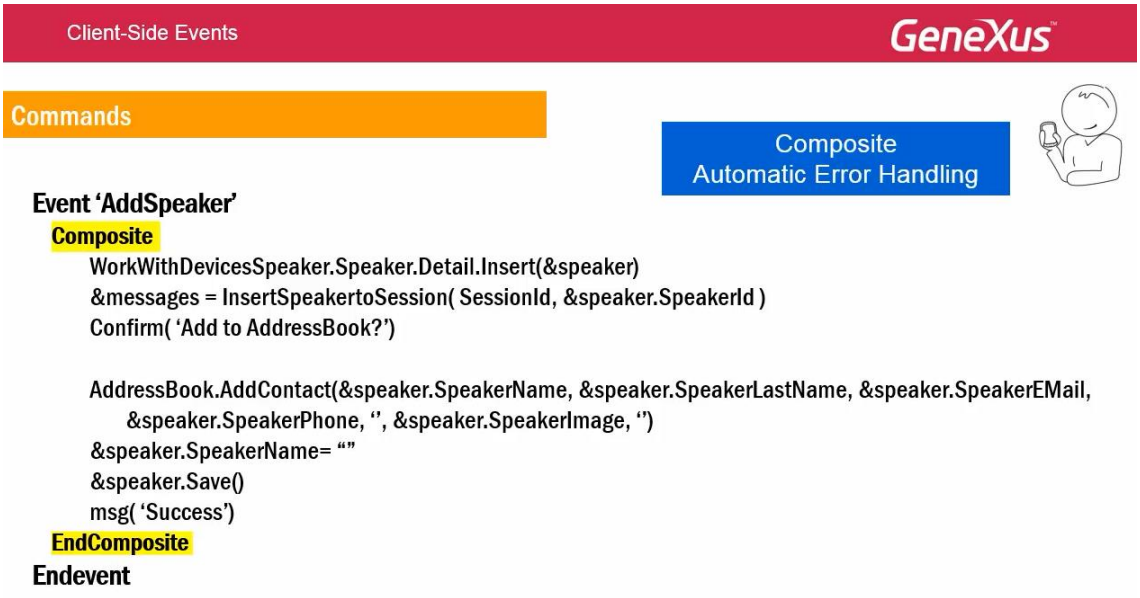
y vemos cómo se está desplegando el mensaje de error correspondiente a la regla.

Damos OK y podemos observar que el mensaje “Success”



no está mostrándose, puesto que la ejecución se interrumpió aquí:





Para terminar: si olvida colocar el comando, GeneXus se lo advertirá al grabar… y al tener que escribirlo usted recordará necesariamente la diferencia en el comportamiento de la ejecución y el manejo de errores de los eventos del cliente en objetos para Smart Devices.

Con esto terminamos de estudiar este importante comando.

Video filmado con GeneXus X Evolution 3