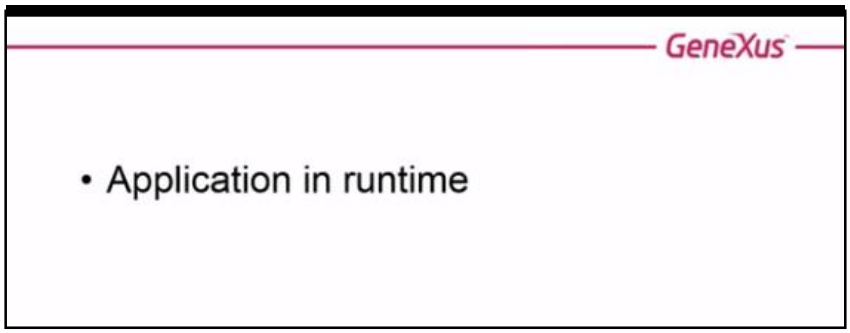
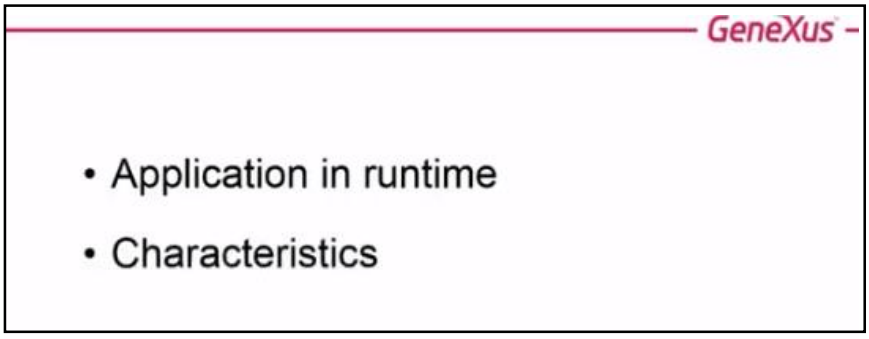
Demo: empezando a desarrollar la aplicación en su parte web

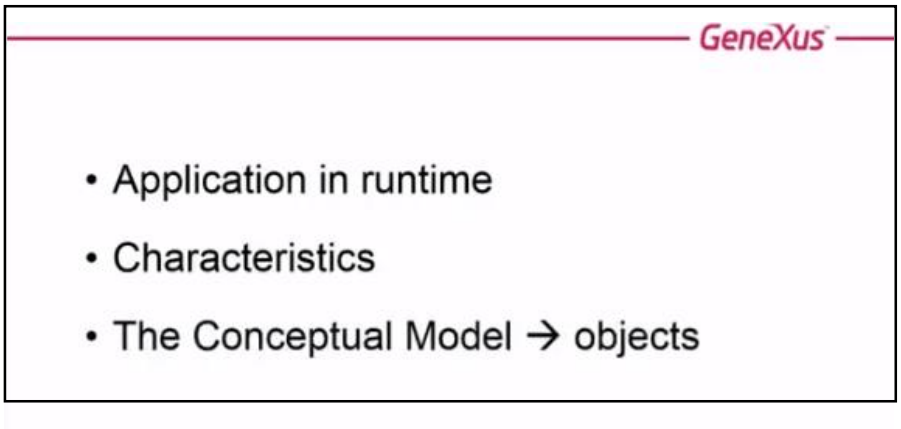
En videos anteriores, habíamos empezado por ver la aplicación a la que queremos llegar en ejecución.



Luego habíamos estudiado las características generales de las aplicaciones para Smart Devices:

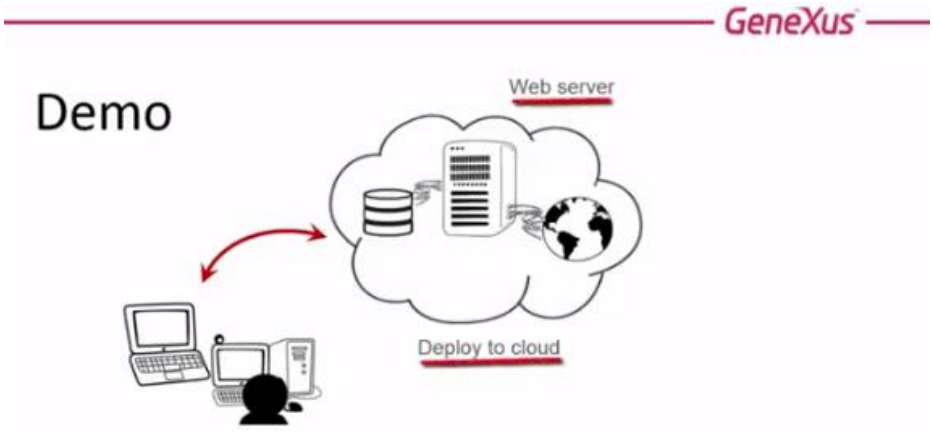


para pasar luego a estudiar su modelo conceptual y los objetos subyacentes:



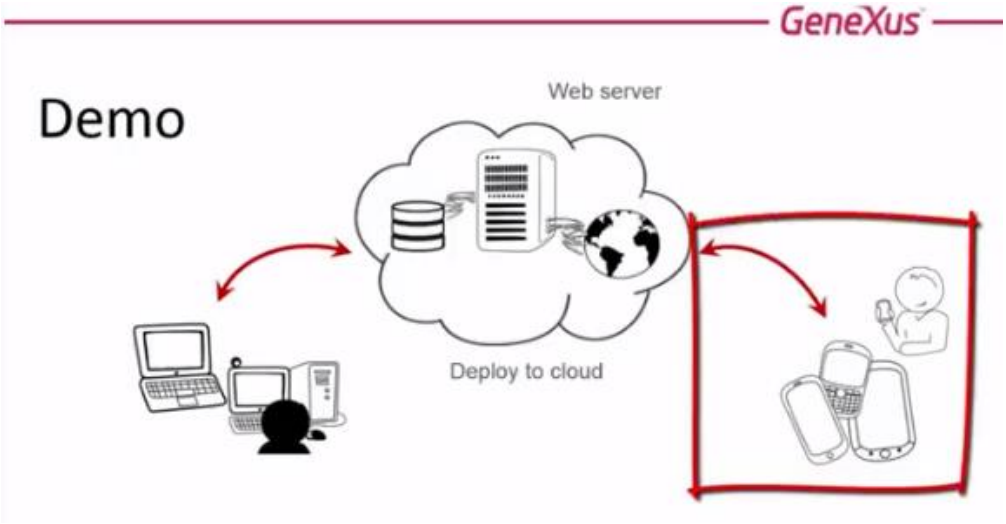
Llegó el momento de poner todo esto en práctica.

En este video construiremos la aplicación **desde cero, para web, prototipando en la nube,**



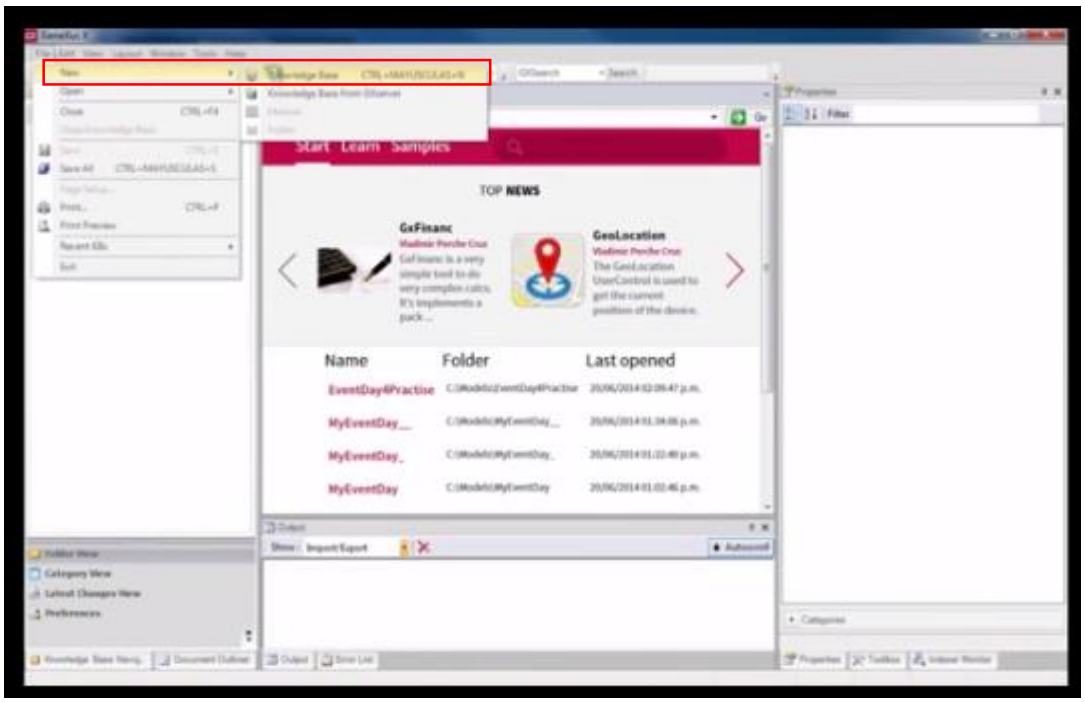
… y en el siguiente, la continuaremos, para agregar la parte de Smart Devices.

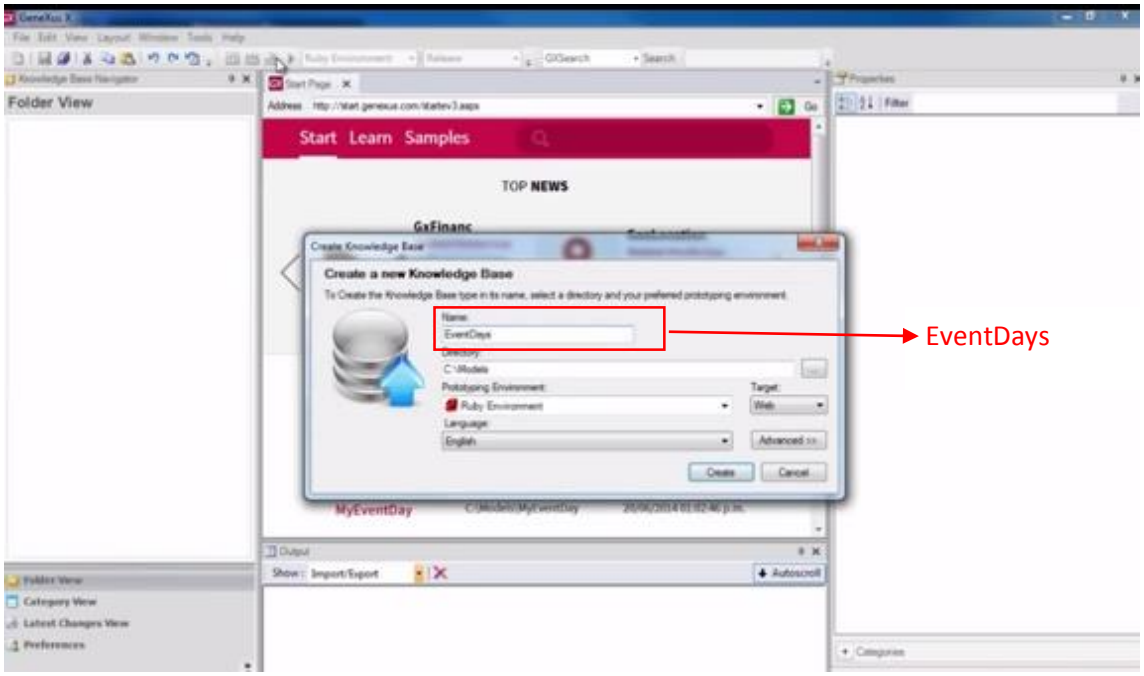
Puede saltearse este video si desea ir directamente a la implementación para Smart Devices…



Vamos a empezar a construir nuestra aplicación: EventDay

Empecemos por crear una KB de cero…

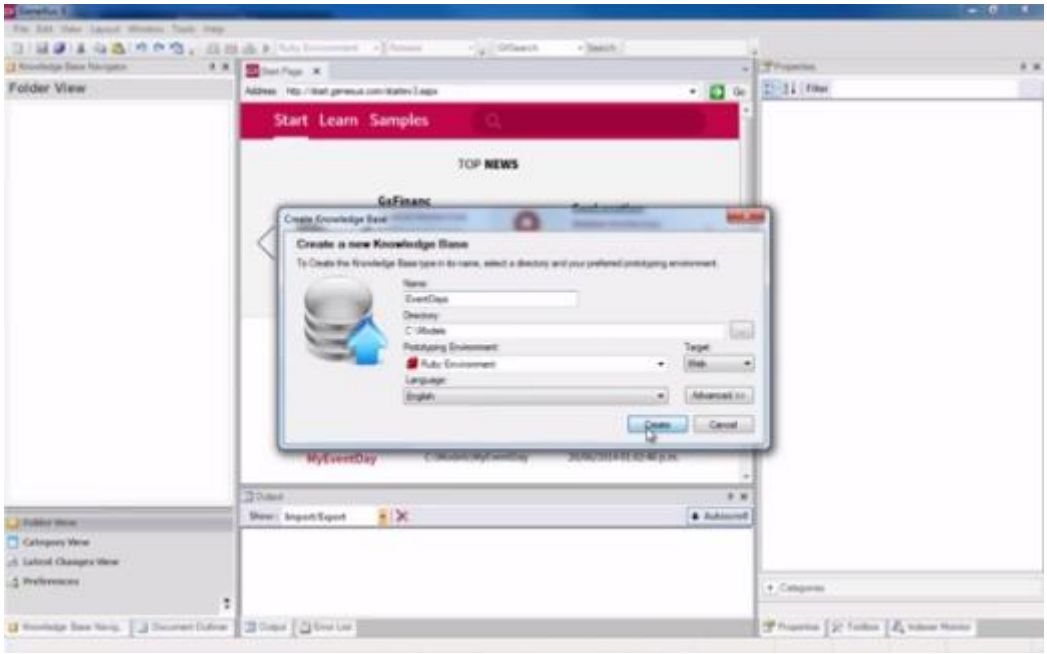




Vamos a dejar por defecto el environment predefinido Ruby.

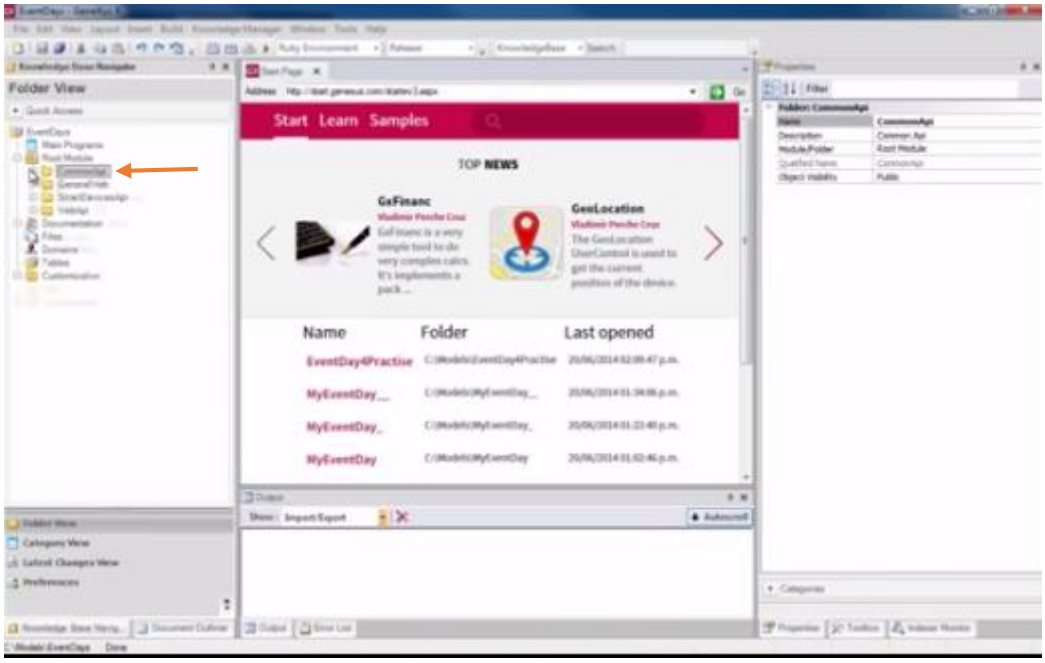
Luego en el práctico usaremos .Net.

Y creamos la KB:



Observemos que aparecen los folders:

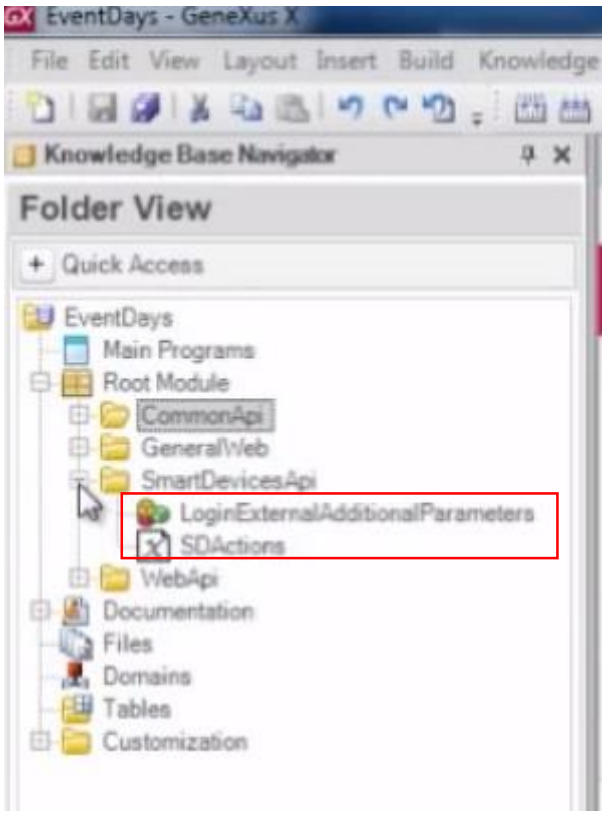
→ **CommonApi**



con un par de objetos..

→ **GeneralWeb** (ya conocido)

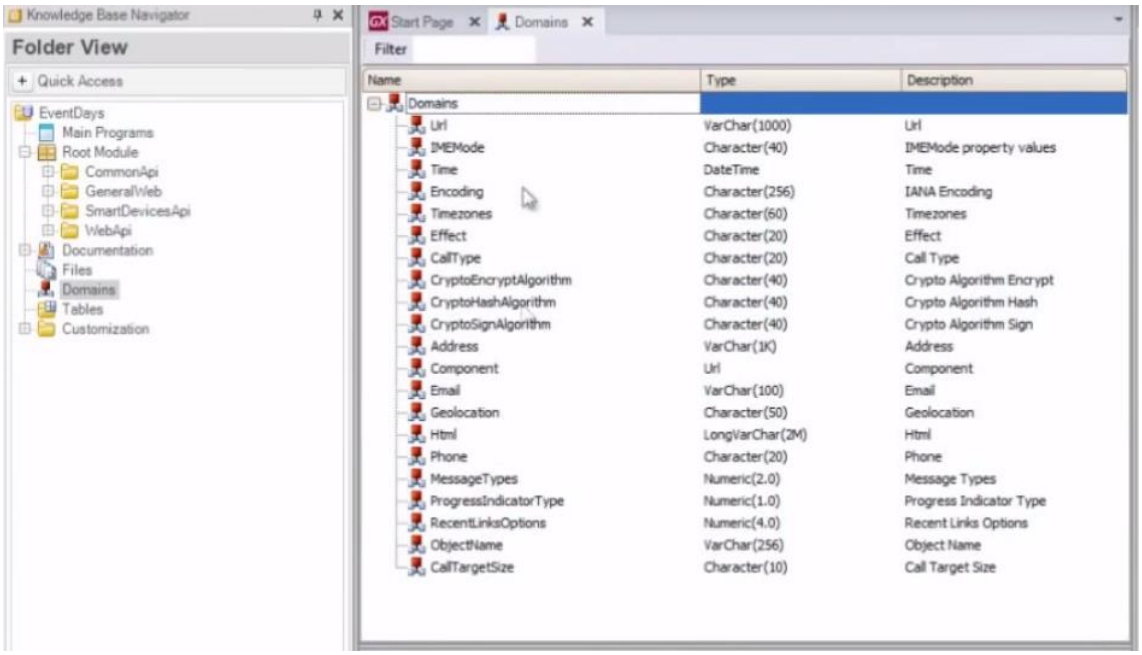
→ **SmartDevicesApi:** que por ahora sólo tiene estos dos objetos



y luego veremos cuándo se puebla con más objetos este folder

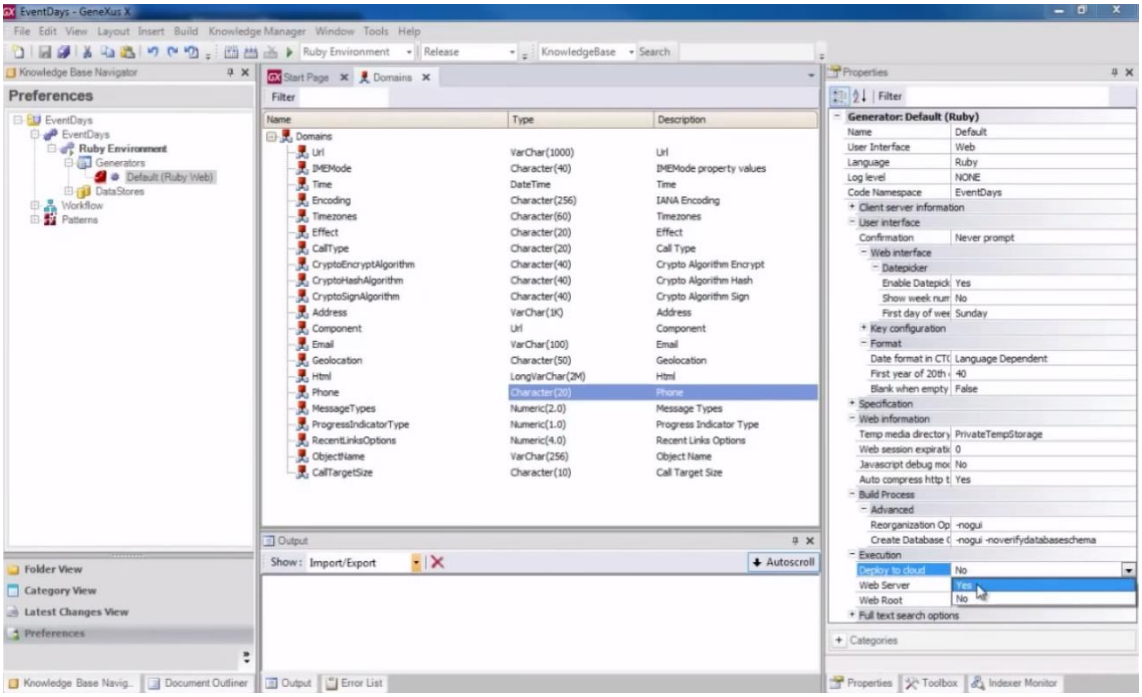
→ y el **WebApi**.

Y si vamos a los dominios



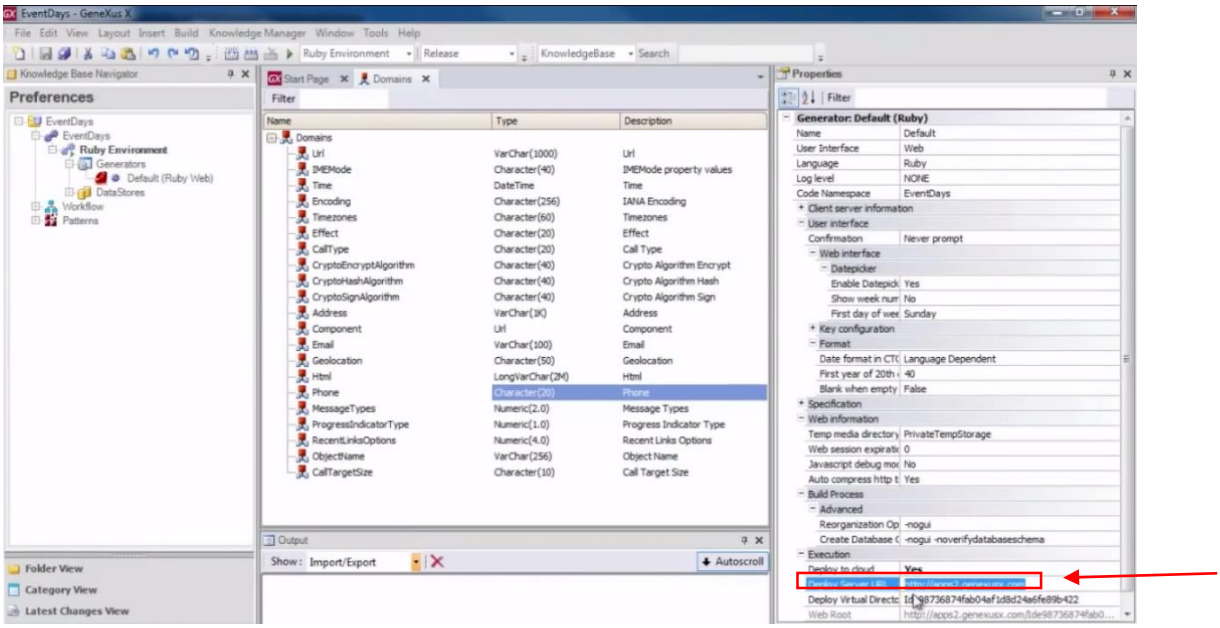
vemos que tiene estos dominios predefinidos (entre ellos: Address, Email, Geolocation, Phone) que usaremos en nuestra aplicación para Smart Devices, conocidos como **Dominios Semánticos**.

Veamos la propiedad “Deploy to Cloud” que por defecto para el Generador web Default está en NO. Vamos a cambiarla a YES.

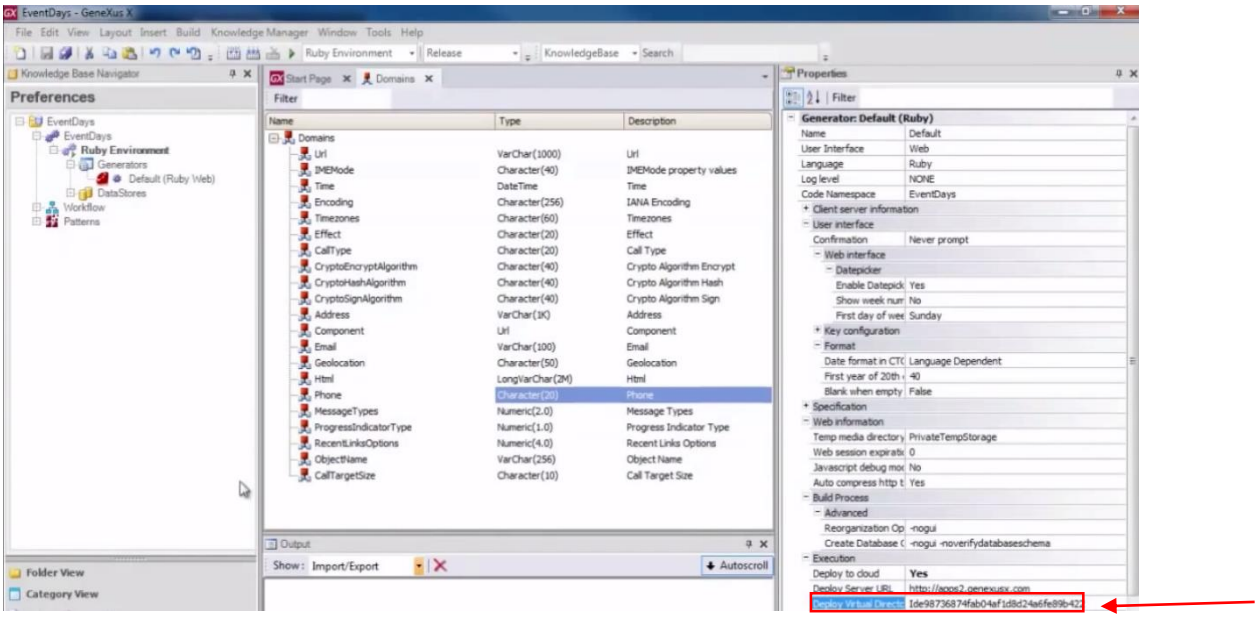


De esta manera vamos a prototipar en la nube.

Vemos acá donde se encuentra la URL de la nube, del hosting en la nube:

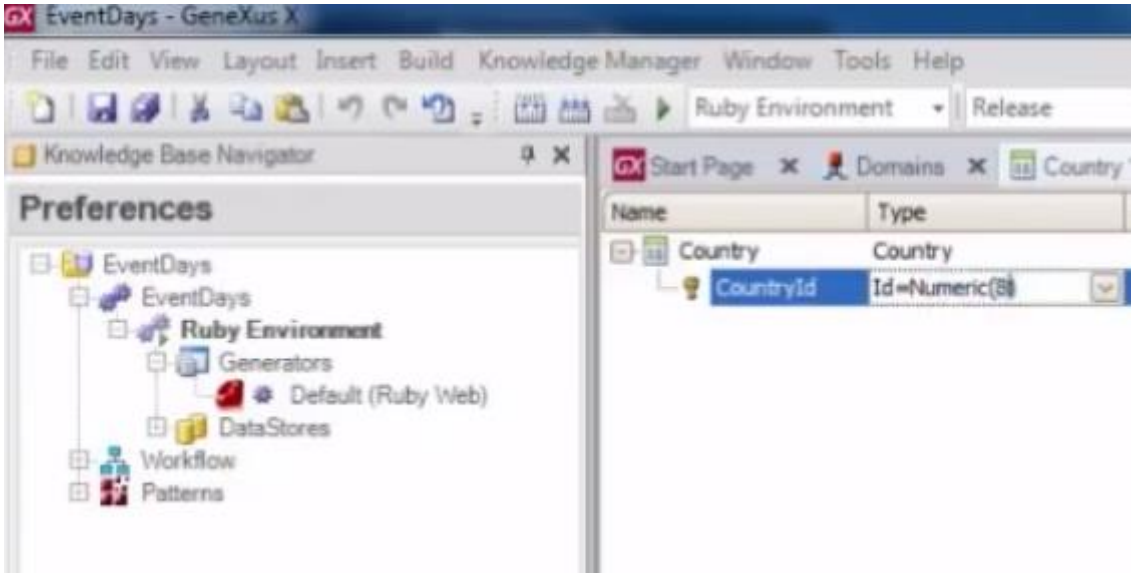


y aquí el directorio virtual:

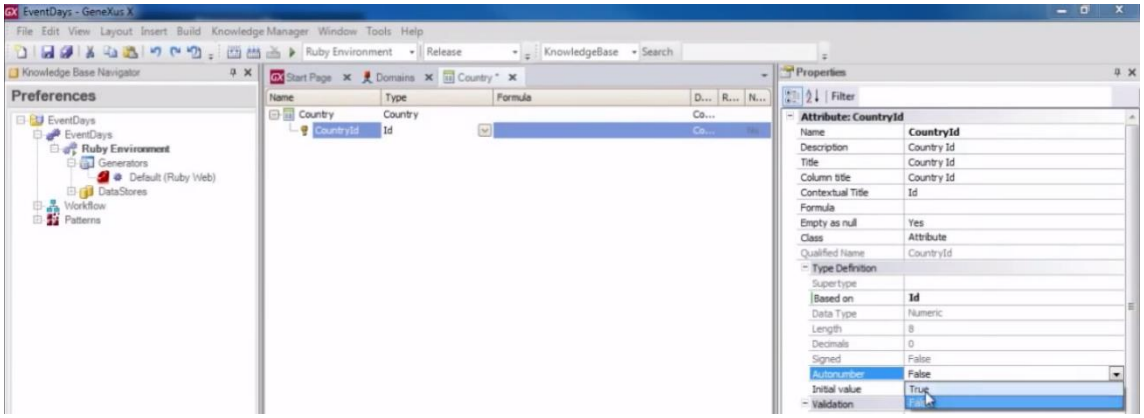


Vamos a empezar por crear nuestras primeras transacciones.

La transacción Country, cuyo identificador es CountryID, dominio ID estamos definiendo, (numérico de 8):

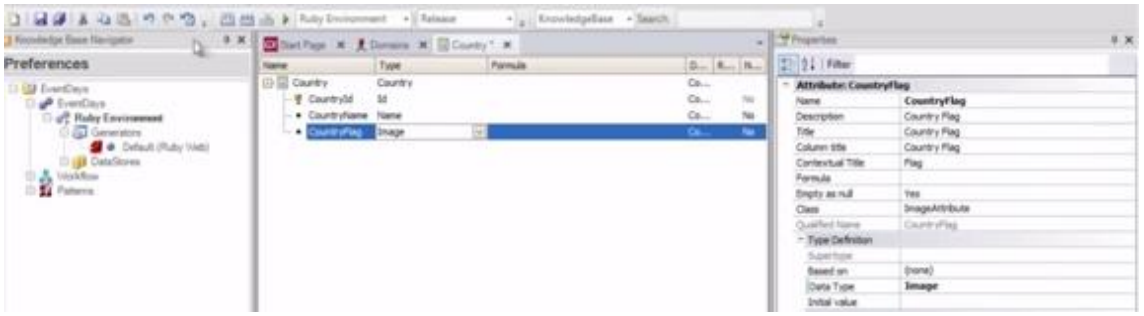


Vamos a decir que se autonumere el atributo:



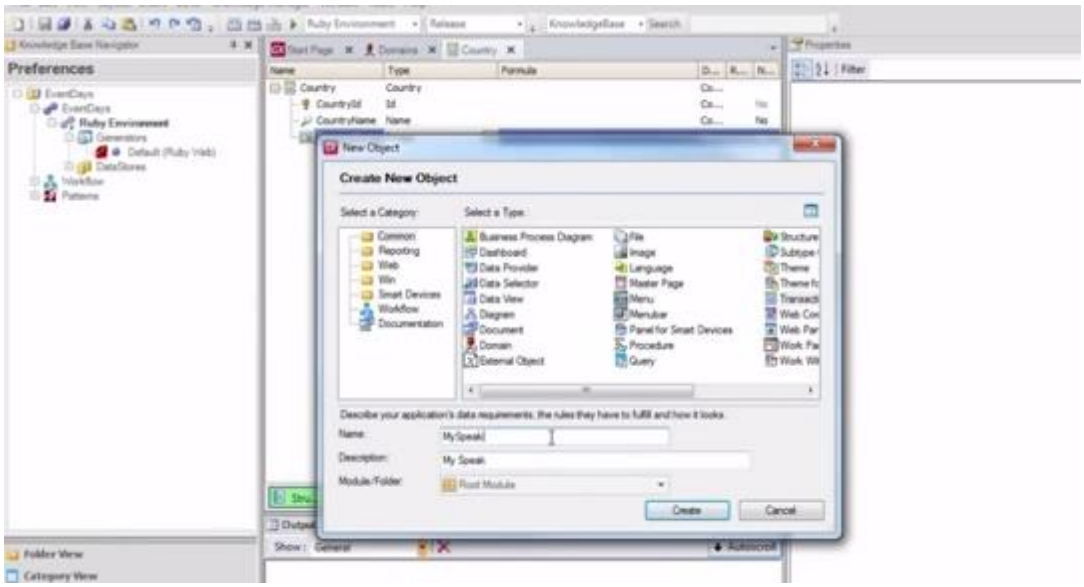
Luego definimos CountryName, de dominio: Name (varchar de 20)

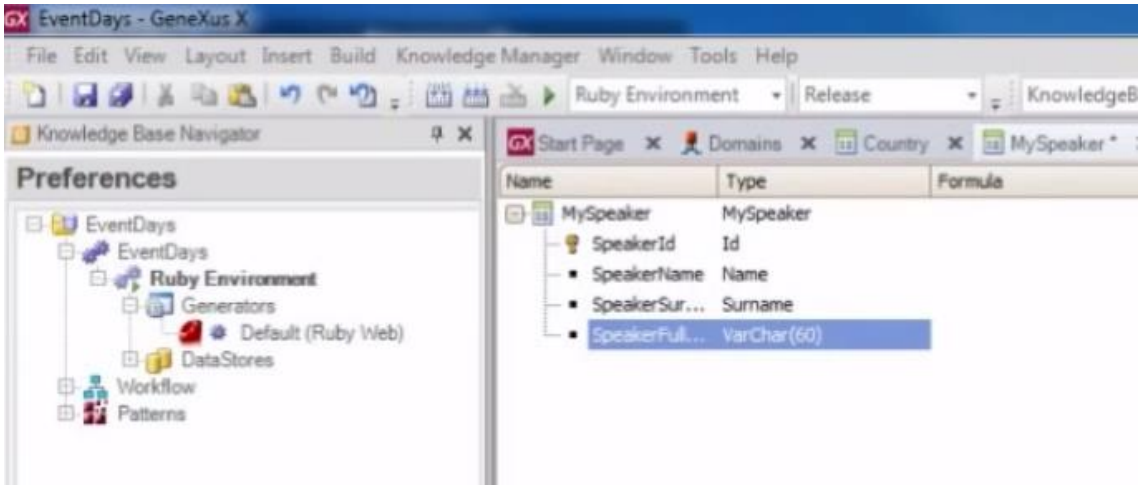
y CountryFlag para almacenar la bandera del país, de tipo Image:



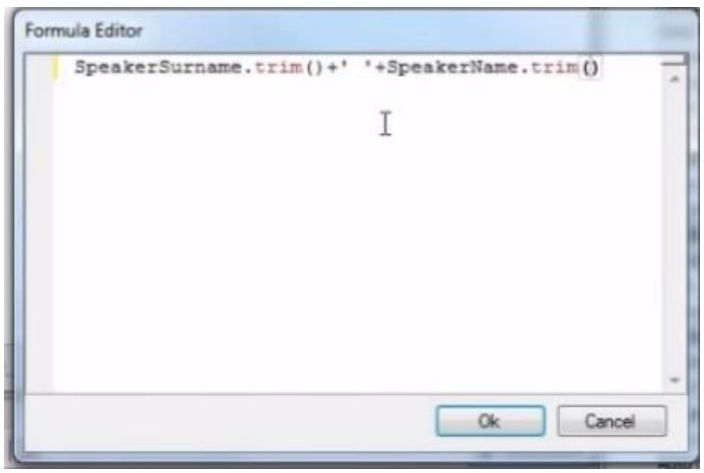
Grabamos…

Y ahora definimos la transacción MySpeaker:





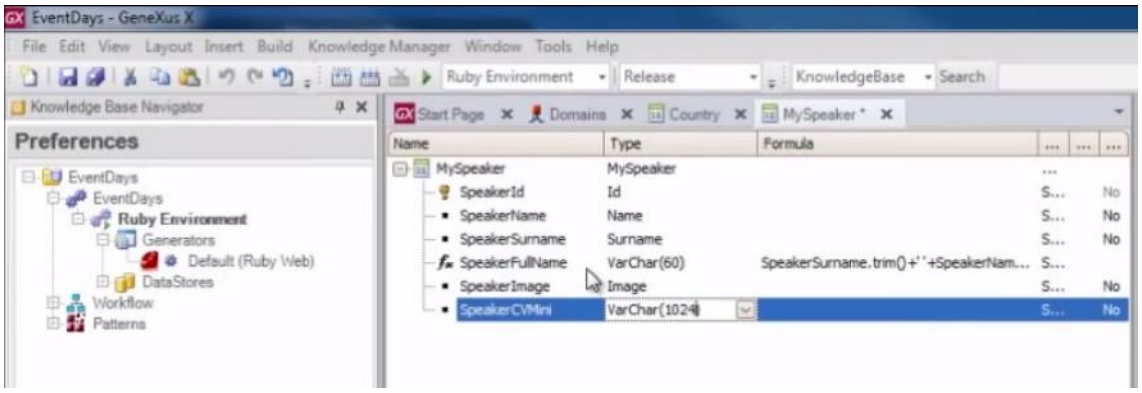
Luego definimos el atributo: SpeakerFullName, de tipo varchar de 60, que va a ser un atributo fórmula:



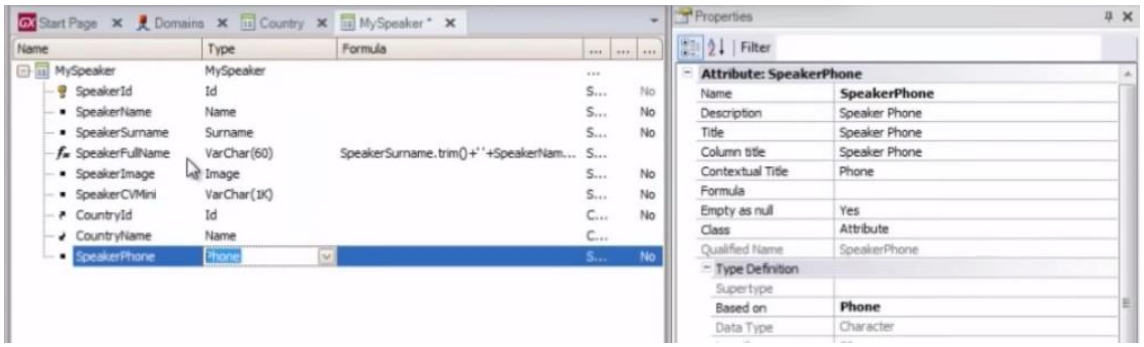
(SpeakerSurname, al que le aplicamos el método trim para quitarle los espacios en blanco, dejamos un espacio y SpeakerName, para tener el nombre completo del orador).

Luego definimos el atributo: SpeakerImage, para para tener la foto del orador.

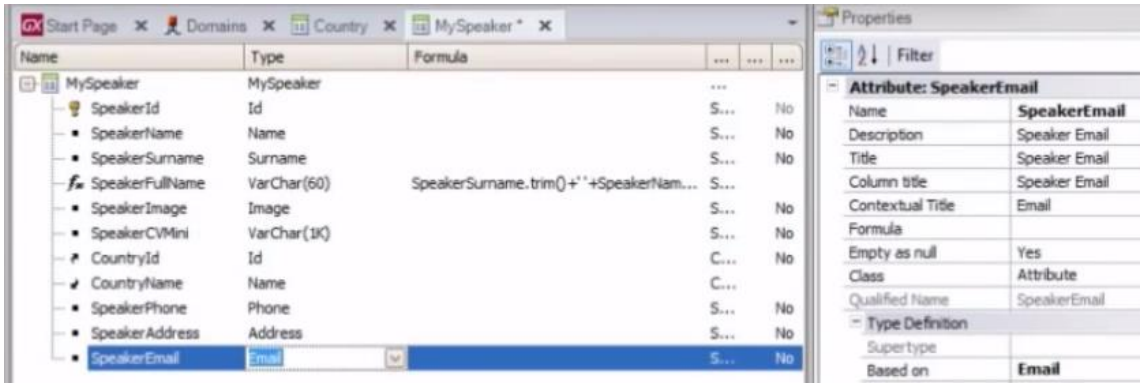
SpeakerCVMini, para tener un resumen del Currículum Vitae del orador…



El país del orador… e inferido su nombre… el teléfono y observemos que está asumiendo el dominio semántico “phone” que luego veremos en acción



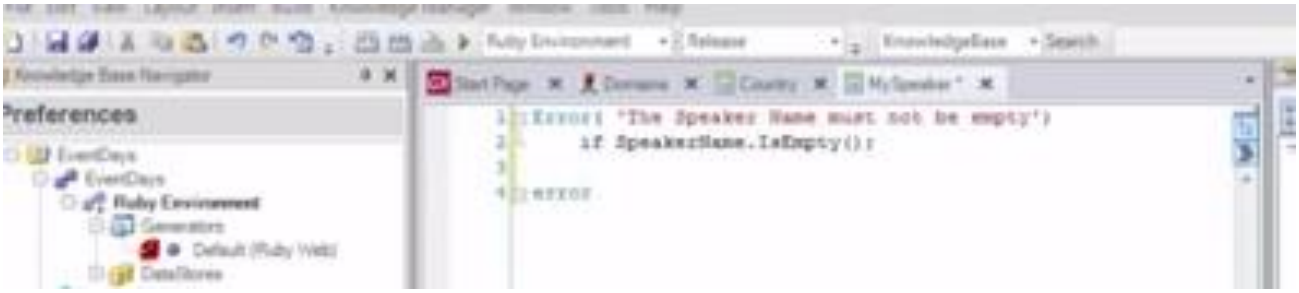
la dirección del orador y por último el e-mail.



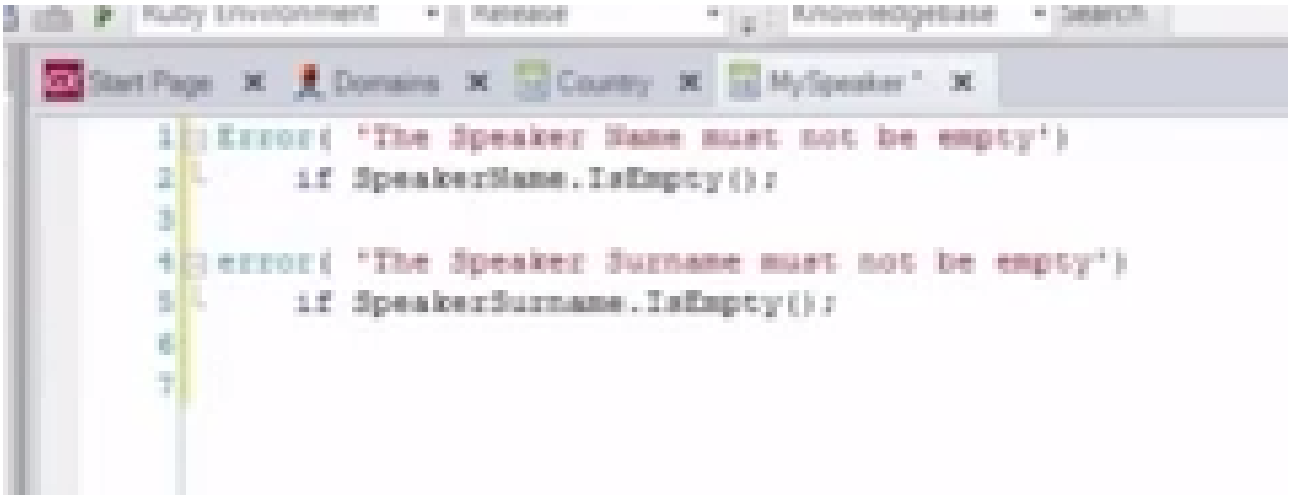
Grabamos…

Vamos a especificar que también se autonumere el identificador, grabamos nuevamente, y vamos agregar algunas reglas a esta transacción.

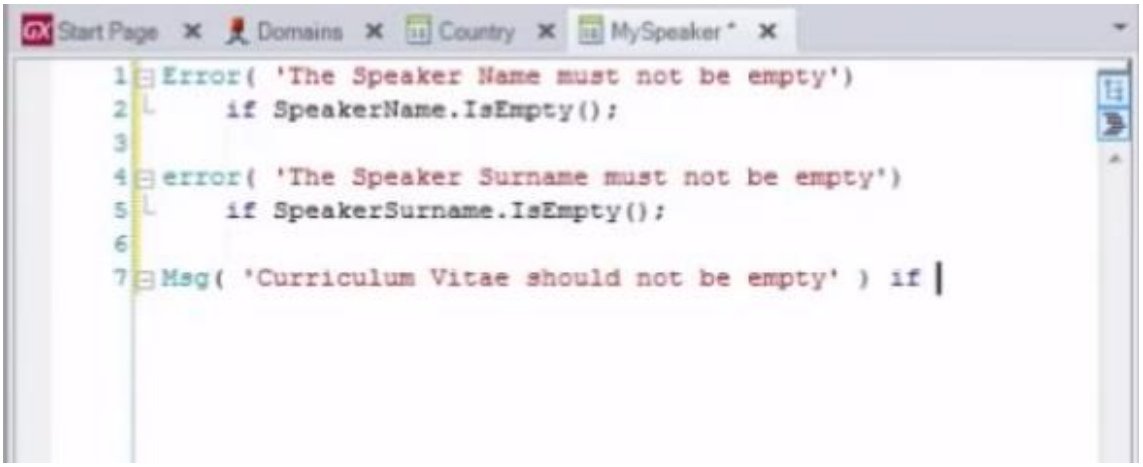
Por ejemplo esta regla de error que se va a disparar cuando el SpeakerName esté vacío.



Esta otra de error también, que se va a disparar cuando el apellido del speaker este vacío



y un mensaje, ya no un error



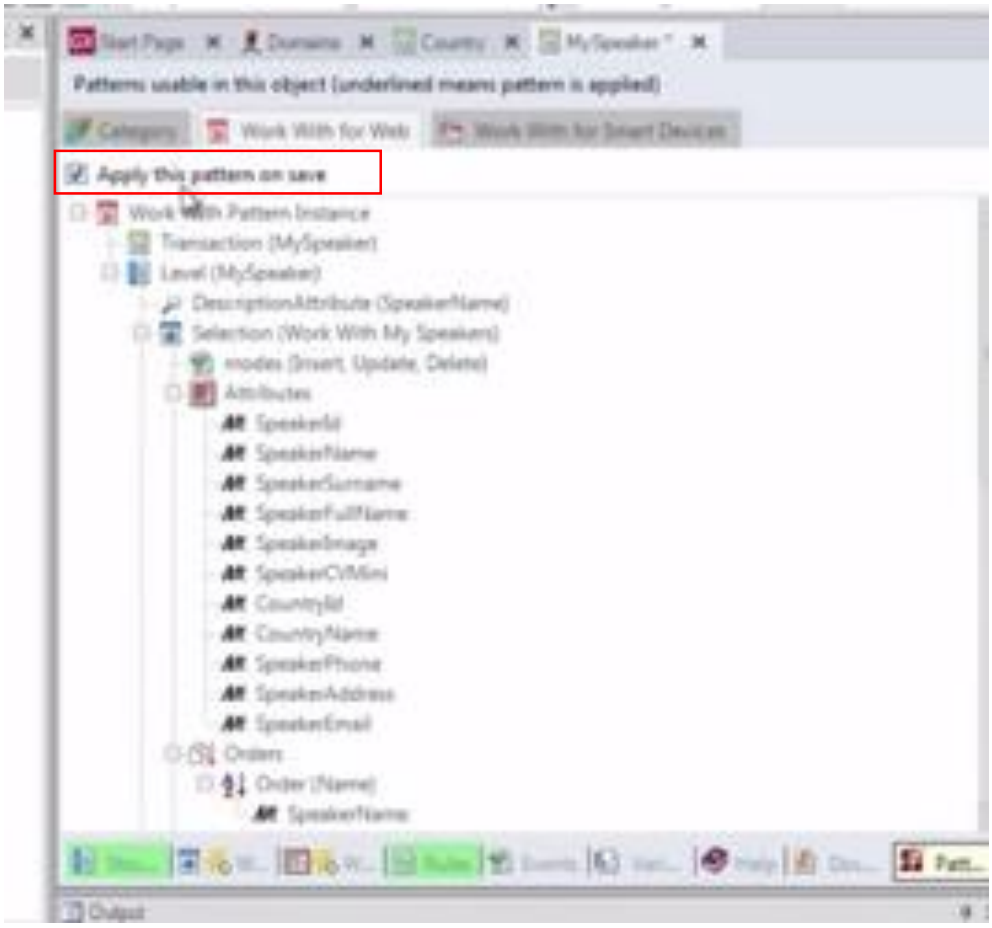
si el CV del orador está vacío.

Grabamos.

A partir de este momento tenemos dos caminos posibles: O nos concentramos en Desarrollar la parte web de la aplicación o nos concentramos para desarrollar la parte para Smart Devices, o lo vamos haciendo en paralelo.

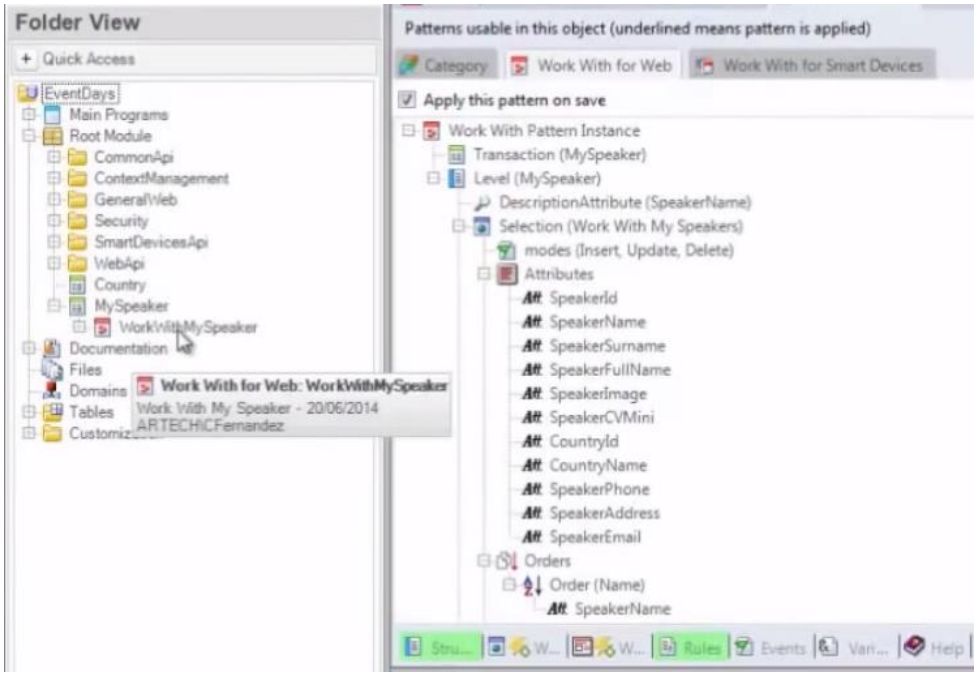
Vamos a empezar por la parte Web.

Lo primero que haríamos sería ir a la sección de Patterns, de esta transacción MySpeaker y aplicarle el pattern: Work With for web.



Como vemos aquí tenemos el archivo de instancia correspondiente.

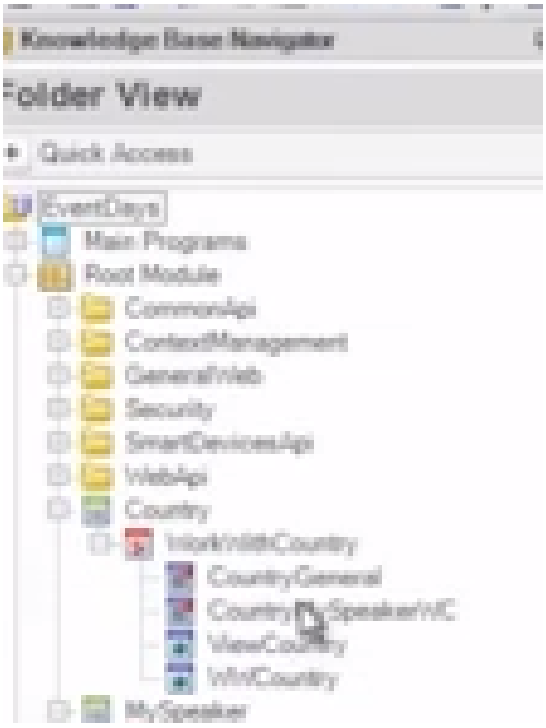
Si vamos al folder view vamos a observar que tenemos aquí el objeto MySpeaker y que cuando grabamos se va a crear bajo el objeto, tanto el archivo de instancia como los objetos que van a implementar el pattern.



Aquí lo vemos:

- el archivo de instancia
- y los objetos que lo implementan: los web panels y el web component.

Lo mismo, si vamos a la transacción Country, a su sección de Patterns, y aplicamos el “Work With for Web” vamos a ver en el Folder View que se crea tanto el archivo de instancia (que no es un objeto), como los objetos que sí implementan lo que dice este archivo de instancia.

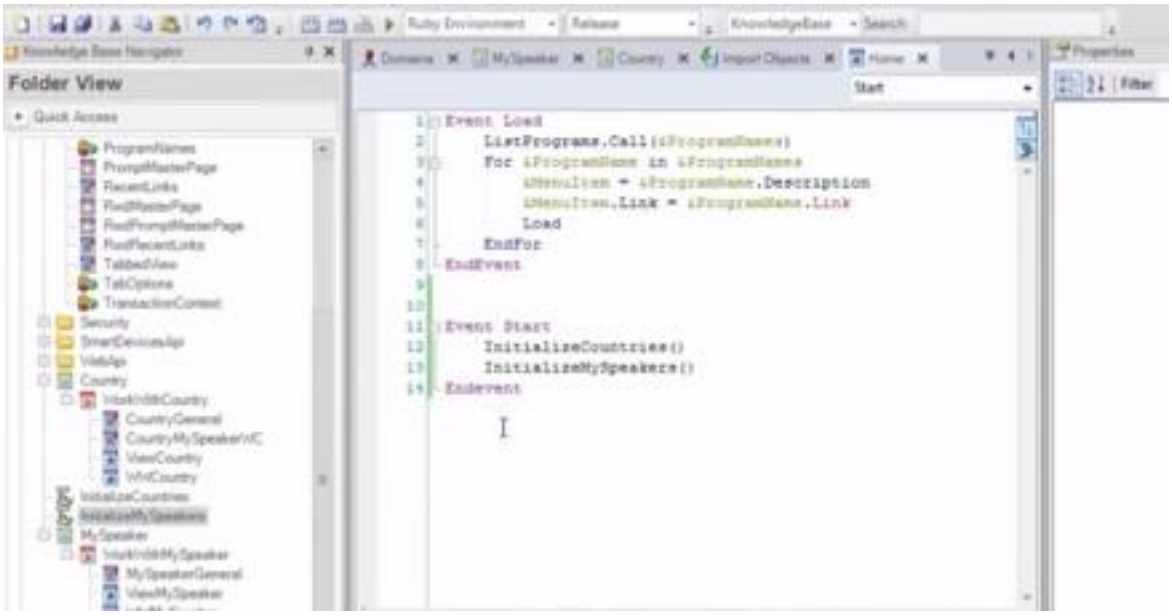


Además sabemos que se ha creado este Web Panel **Home** para invocar a ambos Work With.

Para inicializar ambas transacciones con datos, me voy a importar un xpz que contiene dos procedimientos de inicialización.

Este es el xpz. Lo voy a cargar y voy a importarlo.

Entonces vamos al Home y en el evento Start vamos a invocar a ambos procedimientos de manera de simplificarnos la carga y no tener que hacerlo manualmente. Entonces vamos a inicializar los países y vamos a inicializar con datos los oradores.

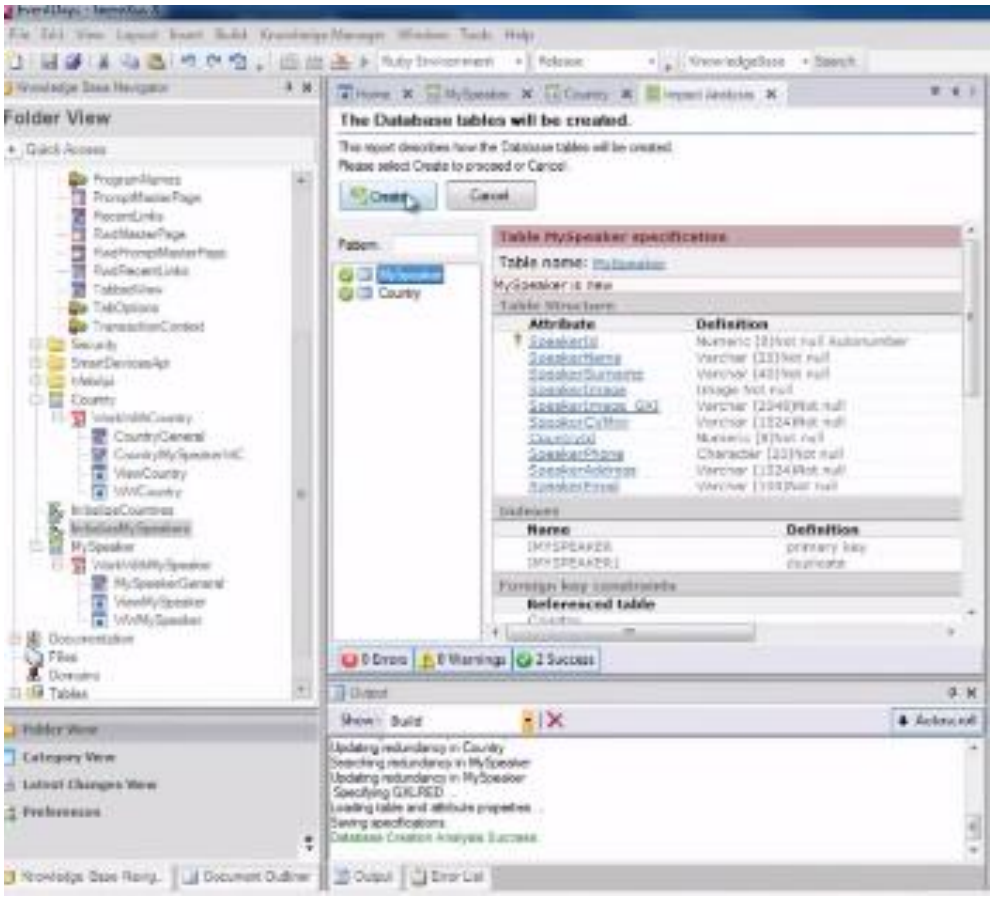


Grabamos…

Y ahora ya estaríamos listos para probar la aplicación web.

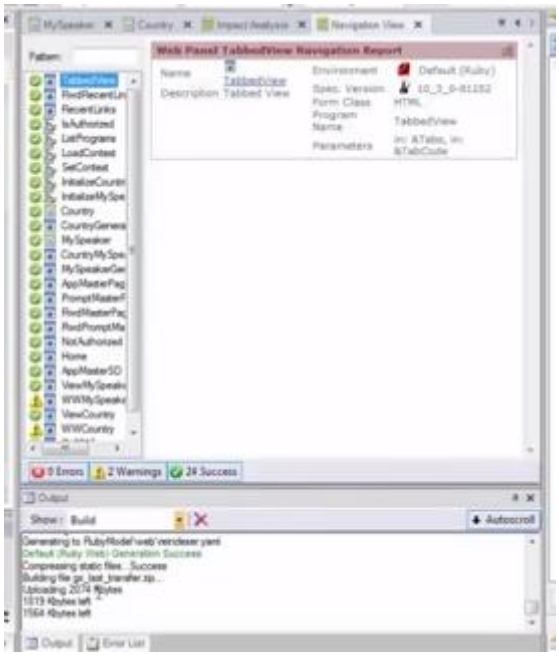
Damos F5…

Creamos las tablas en la base de datos



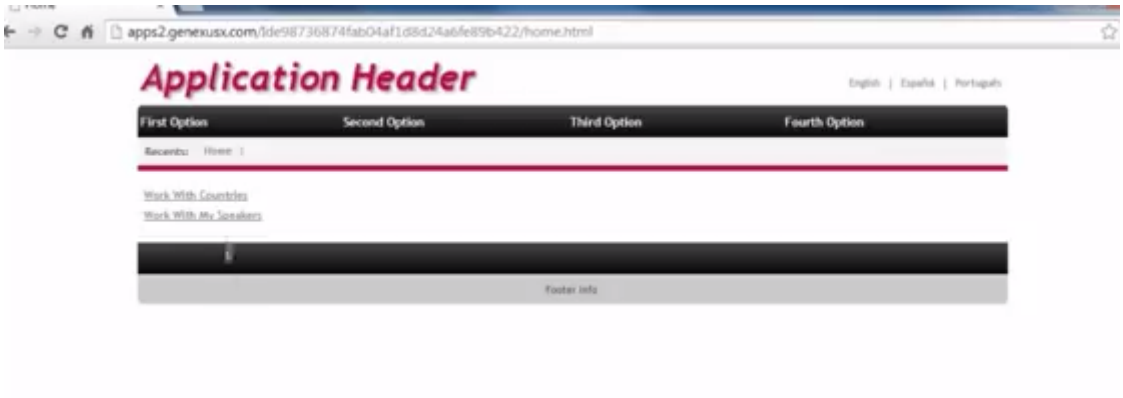
estas tablas se crearán en la base de datos en la nube.

Vemos como se está subiendo la aplicación a la nube…

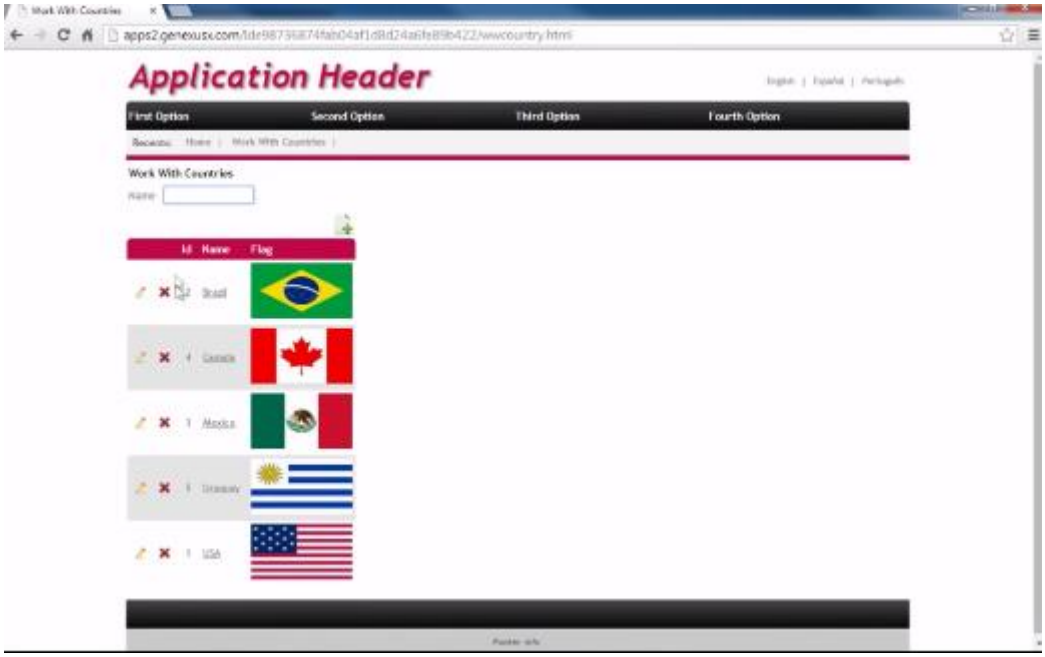


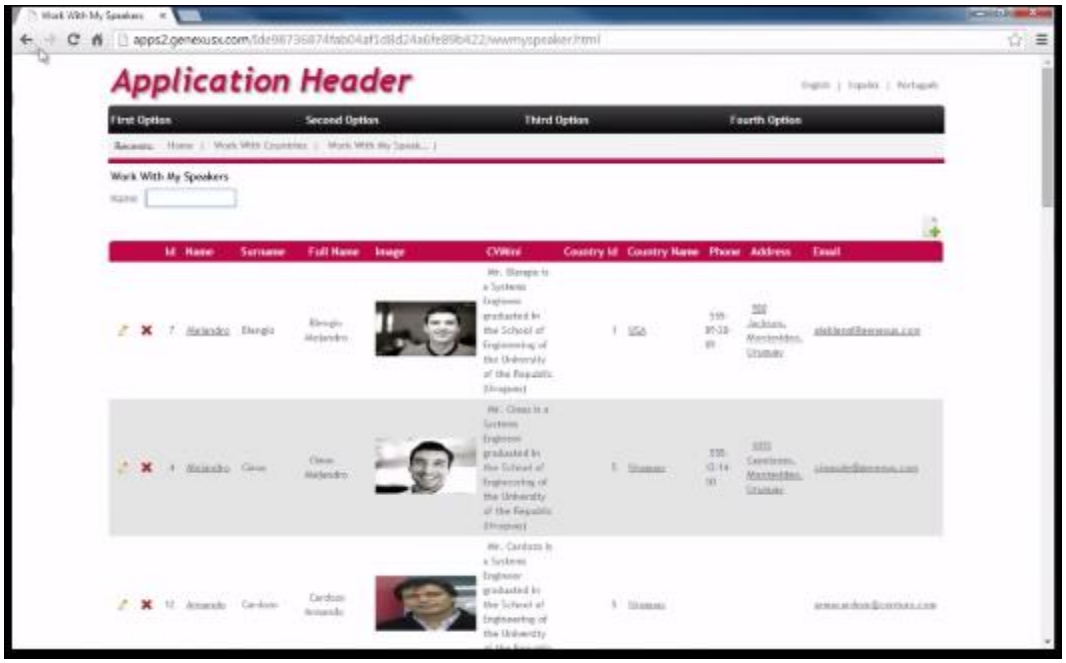
y cómo se nos abre el Developer Menú, que contiene entonces un link por cada objeto.

Ejecutemos el Home, el objeto que creó el pattern work with al ser aplicado… (min 8:40)



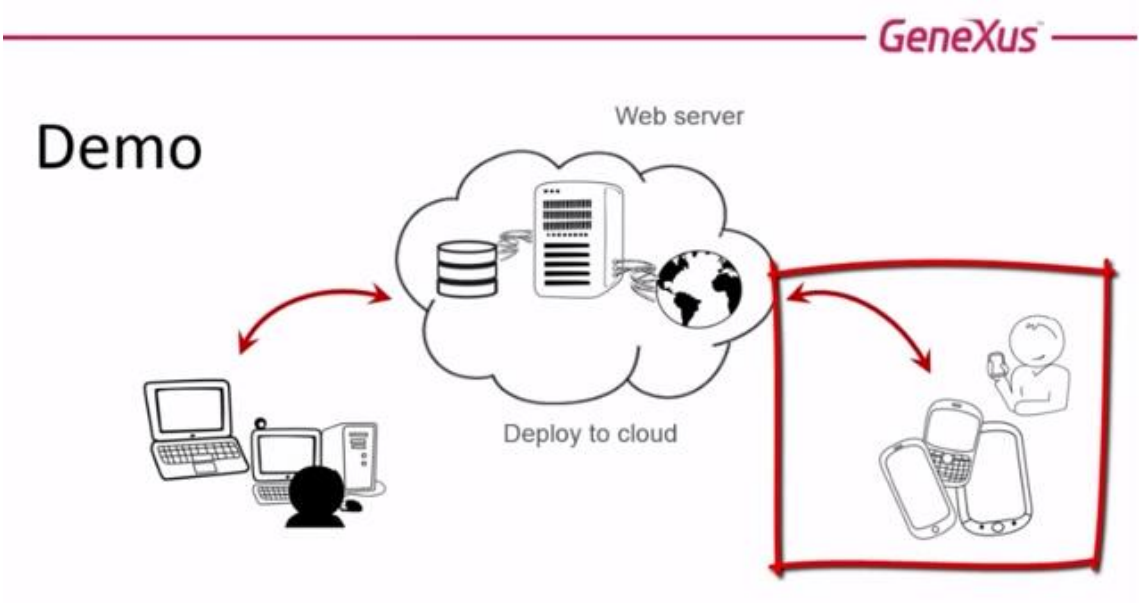
se tiene que haber ejecutado el evento Start y se tienen que, entonces, haber cargado las tablas con datos…





Esto es lo conocido por nosotros.

Ahora queremos empezar a desarrollar la aplicación para Smart Devices.



Es lo que haremos en el siguiente video.