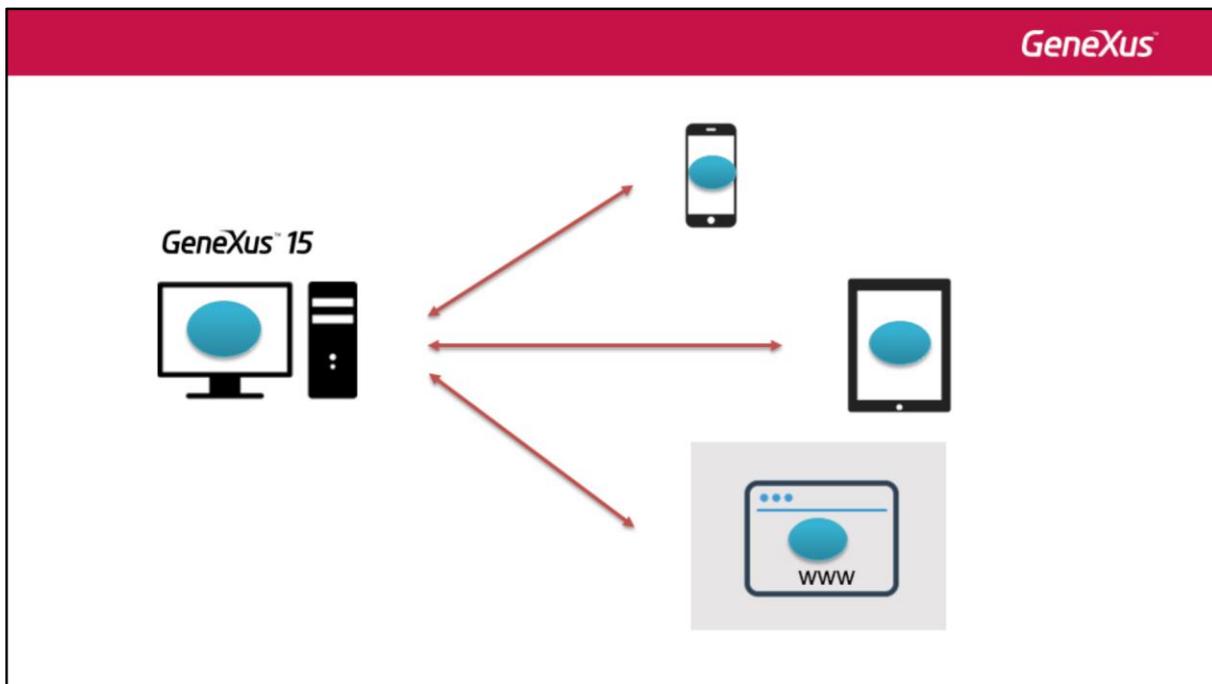


Smart Devices

GeneXus™ 15

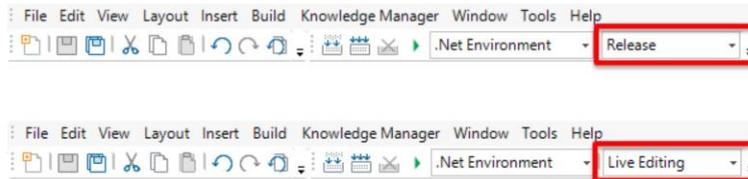
Live editing



Hasta GeneXus 15 cada vez que queríamos probar un cambio de diseño debíamos volver a especificar, generar y compilar la aplicación.

Pero en GeneXus 15 contamos con la herramienta de Live editing, que permite realizar cambios de diseño y algunos de comportamiento y ver instantáneamente esos cambios en la app en el dispositivo, o en el browser de la app web, sin tener que compilar nada, y sin que siquiera esos cambios deban ser grabados.

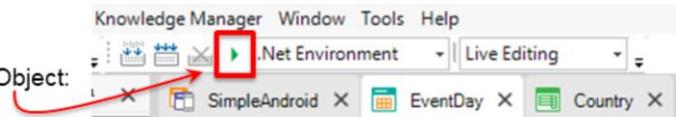
Step 1



Step 2

Run a SD main object

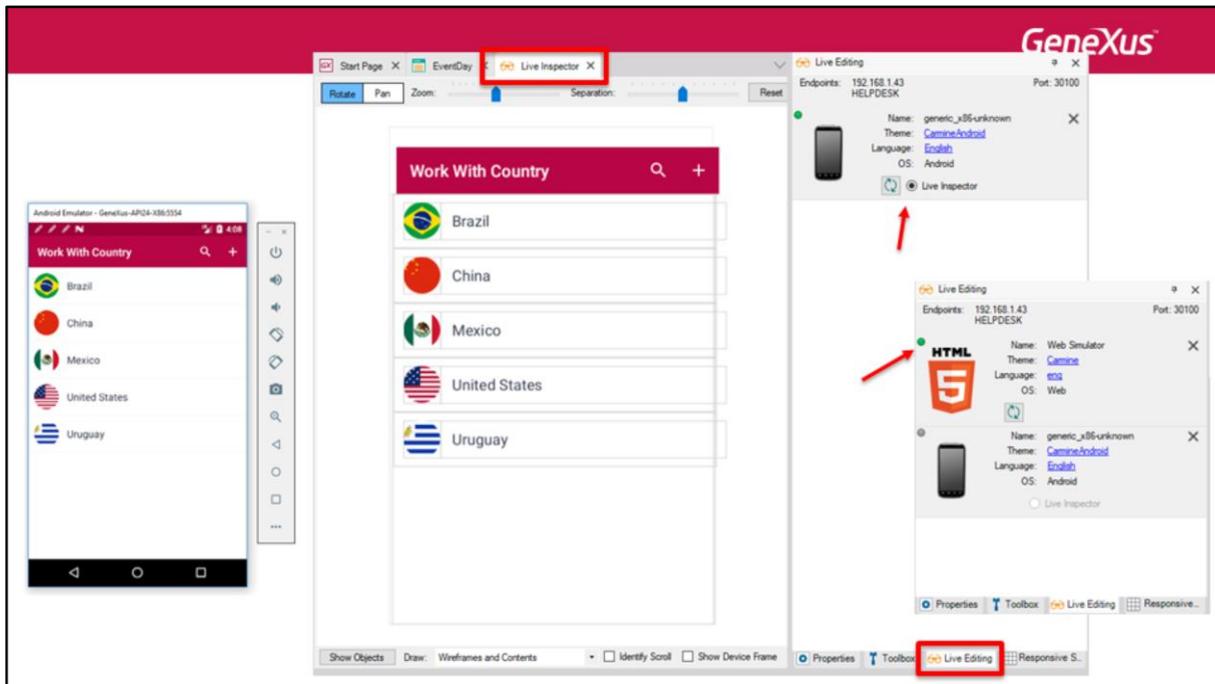
Having a SD main object as Startup Object:



Para usar el Live Editing se necesitan 2 pasos:

- primero cambiar el combo del valor Release al valor Live Editing
- Luego ejecutar la aplicación (asegurándonos de que se compile. Live editing no funciona con el KBN)

De esta manera el servidor de Live Editing quedará “escuchando” los cambios que hagamos a la aplicación y replicará estos cambios sobre la metadata a la que accederá el dispositivo móvil (o el emulador) y que se utiliza para dibujar la pantalla en el dispositivo y definir su comportamiento.



[demo] Lo mejor es ver esto en una pequeña demo. Aquí tenemos una KB bastante limpia que tiene solamente dos transacciones: Country y Speaker, con un Menu for Smart Devices, Event Day, que invoca a uno y otro Work With for Smart Devices.

Y acá lo estamos viendo en un emulador en ejecución. Por ejemplo, vemos el listado de países... Y aquí, el listado de oradores.

Bien, supongamos que vamos a querer probar cambios en el diseño de este Work With (de este List)... Por ejemplo agregarle el apellido del orador, agregar otros datos, modificar cómo estamos viendo la información; capaz que queramos agrandar la imagen para que ocupe más espacio de la pantalla; cambiarle el color... Y demás.

Entonces, para hacer todo eso, antes, hasta la Evolution 3, lo que teníamos que hacer era ir al Work With correspondiente, por ejemplo al de Speakers... Ir a la pantalla... Y hacer las modificaciones correspondientes en el layout y después volver a generar toda la aplicación de vuelta; volver a compilar todo para que otra vez se nos abra la aplicación en el emulador, en nuestro caso (si tuviéramos un dispositivo conectado a la computadora lógicamente sería en el propio dispositivo donde se abriría) y ahí recién podríamos probar los cambios y capaz que nos damos cuenta de que no era eso lo que queríamos y tenemos que volver a realizar el cambio en el layout, volver a compilar todo de vuelta, esperar todo ese tiempo de compilación, para recién ahí otra vez volver a ver el resultado.

Bueno, todo eso se nos aliviana muchísimo con Live Editing porque ya esos ciclos de Build no los vamos a necesitar más.

Entonces, decíamos, para habilitar Live Editing venimos a este combo de aquí, y cambiamos el valor de "Release" a "Live Editing".

No alcanza con hacer esto. Ahora lo que vamos a tener que hacer es volver a compilar la aplicación que queremos probar, con esta funcionalidad de Live Editing.

Si tuviéramos al Menu for Smart Devices EventDay como Startup Object, con hacer un F5 nos alcanzaría para asegurarnos de que se va a compilar. Como no lo tenemos como startup object lo que hacemos es Botón derecho / Run. Y ahí sí nos aseguramos de que se compile la aplicación.

Vamos a ver lo que pasa... que acá ya se habilitó una nueva ventana que no estaba: Live Editing.

Vemos que se actualizó lo que se mostraba en la ventana, y lo que vemos acá, en la ventana de Live Editing es una lista de todos los dispositivos con los que estamos prototipando. En este caso solamente uno.

Y que van a ser los dispositivos con los que el IDE va a estar conectado, listo para trabajar con Live Editing.

Para Live Editing usamos un canal bidireccional entre el dispositivo que está ejecutando la aplicación y el IDE, que funciona como server.

En el ejemplo, el IDE está escuchando... aquí (en la computadora llamada HELPDESK) y el dispositivo que ejecuta la aplicación es el que se llama de esta manera, que es el emulador.

Podemos ver el nombre del dispositivo, el theme que está utilizando, el lenguaje, el sistema operativo.

Por otro lado, vemos que tenemos esta opción acá, Live Inspector, que se abrió en esta ventana, aquí. ¿Qué es lo que estamos viendo aquí? (voy a abrirla un poco más, para que quede más claro). Bueno, en esta pantalla no estamos viendo nada (o estamos viendo muy poco) porque en verdad corresponde al Menu for Smart Devices, que es un tipo de objeto que no es inspeccionable por Live Editing. Pero observemos qué sucede si yo en el emulador, que está siendo escuchado, entonces, por Live Editing, ingreso por ejemplo al listado de países.

Vemos que automáticamente se refrescó en esta pantalla la información. Y ahora sí, me permite pasar por los distintos controles... presentes en el layout. Y no solo eso. Nos va a brindar información que ahora vamos a ver, pero además puedo, si con el mouse hago clic y dejo presionado y muevo el mouse, observemos que puedo desplazar, mover de alguna manera, mostrando las distintas capas de la aplicación. El layout entonces nos queda con la profundidad que tiene, qué controles están por debajo de qué controles.

¿Sí? Aquí vemos la application bar, por ejemplo. Y acá tenemos la tabla1, que contiene a los controles CountryFlag, CountryName...vemos entonces de qué controles se trata... Podemos volver a la posición plana, con un Reset. O moverlo para arriba o para abajo, como queramos. Incluso podemos hacer zoom si no vemos bien la información con ese tamaño, y demás. Voy a hacer un reset, y por ejemplo si hago clic sobre el país, Brazil, vemos el objeto al que pertenece ese control, el nombre del control, su clase... en este caso tiene una clase label, y me dice cuál es el nombre de esa clase, el tamaño y la posición en la pantalla.

Me paso a otro que es del mismo tipo, me muestra lo mismo. Si me paso a la imagen, vemos que me aparece CountryFlag. Si hago el clic sobre el Application Bar me dice que el objeto es application bar, el nombre del control, la clase, y así sucesivamente.

Y no solo eso, sino que yo además haciendo clic allí puedo o editar las propiedades de esas clases... O veamos cómo directamente en la ventana de propiedades ya me está mostrando la información del control, como para que yo pueda modificarlo directamente aquí, o incluso propiedades de su clase.

Observemos qué pasa si le cambiamos la clase al atributo CountryName, de AttributeTitle a AttributeTitleRed, por ejemplo. Vemos que automáticamente cambió el color, porque en el clase establecimos justamente que el Forecolor correspondía a este color de aquí.

Si vamos al dispositivo, vemos que aquí también cambió. Y esto lo hizo automáticamente, sin que tuviéramos que hacer nada más que cambiar la clase de control. Y esto además no se grabó; por tanto, si nos arrepentimos, alcanza con que cerremos el objeto sin grabarlo.

Algunas cosas las podremos hacer directamente desde aquí y para otras tendremos que ir al layout; por ejemplo si queremos agregar una columna entre la imagen del país y el nombre, para generar espacio en el medio. Para ello, venimos al list –al layout-, y nos posicionamos sobre la tabla, que contiene a estos dos elementos, que tiene la propiedad Columns Style, justamente indicando que tenemos dos columnas: una que ocupa 44 dips, y la otra que ocupa el 100% restante. Podemos

agregar una columna nueva: Insert Column After, entre la primera y la segunda, y lo que tenemos que hacer es configurar el tamaño de cada una de las columnas. Si dejamos esto así y vamos a Live Inspector vemos cómo se ajustó automáticamente el tamaño –vemos que está apareciendo la columna del medio aunque no tiene nada por tanto no vemos nada ahí, pero ya vemos cómo se ajustó el tamaño. En el emulador veremos exactamente el mismo ajuste.

Entonces ahora podemos jugar con el tamaño, por ejemplo decir que queremos que la segunda columna ocupe un tamaño fijo de 10dips y vemos cómo se autoajustó nuevamente... Y por ejemplo decirle que la primera queremos que ocupe ahora un tamaño relativo de 30%, por lo cual la última va a tener que ocupar un 70%, el 70% restante.

Y así vemos, entonces, cómo la información se va visualizando, en vivo, sin necesidad de ninguna compilación. ¿Qué cosas puedo ver en vivo y cuáles no va a depender de si estoy trabajando con elementos de la interfaz, con elementos ya cargados –por ejemplo acá tenemos un atributo, pero estos valores de aquí ya estaban cargados en el layout cuando empezamos a ejecutar la aplicación... O aquí mismo, si yo elijo Canadá vamos a ver el detalle de Canadá. Esta información evidentemente se va a buscar a la base de datos, pero a partir de que está cargada, yo con esta información puedo trabajar. Ahora, ¿qué pasaría si yo agregara acá algún atributo que no estaba inicialmente en la información que se estaba cargando? Es decir, si hiciera un insert de un atributo. Lo que va a suceder es que se va a mostrar vacío ese elemento, no se va a ver nada, porque, para tomar efecto la carga de ese atributo, se tiene que ejecutar... bueno, depende, o el evento Refresh o el evento Load. Va a depender de si estamos trabajando en la parte plana, o si por ejemplo estamos trabajando con los oradores, en este caso, de Canadá, que van a ser varios... Bueno, en ese caso será en el evento Load. Pero van a tener que ejecutarse esos eventos en el servidor para poder ir a buscar esa información. Pero el código de esos eventos no es inspeccionable por Live Editing. Aquí, cuando se muestra la info de Canadá es porque se están ejecutando los Data Providers rest, que estaban compilados con el último Run que hicimos. Si necesitáramos agregar un atributo a la información mostrada aquí, esos data providers que van a buscar la información a la base de datos van a tener que ser compilados nuevamente. Por tanto vamos a tener que volver a hacer un Run. Live Editing todo lo que puede hacer es ejecutar los Data Providers pre existentes, pero lógicamente, no puede modificarlos.

Algo que no dijimos, si hiciéramos F5 sobre el objeto Home web, por ejemplo, aquí mostramos cómo se verá la ventana de Live Editing cuando se abra el browser como consecuencia del proceso de Run. Con el punto verde se muestra cuál es el dispositivo que se está escuchando en ese momento.

Vemos que aquí pasó a escucharse el browser y dejó de escucharse el dispositivo, el emulador.

Además tenemos una opción Live Inspector, que al seleccionarla abrirá una ventana donde podremos visualizar la aplicación tal cual la vemos en el emulador, pero que, a diferencia del emulador, nos permite desgranar la información y ver las diferentes capas que componen el layout, los nombres y clases de cada control, etc., y realizar cambios directamente sobre ella y verlos impactados automáticamente.

The screenshot displays the GeneXus Live Editor interface. On the left, an Android emulator window titled "Android Emulator - GeneXus-AP04-X865554" shows a mobile app with a red header and two buttons labeled "Countries" and "Companies". A red arrow points to the "Countries" button. The central area is a wireframe view of the app, showing a red header bar and a white content area. The right side features a "Live Editing" panel with the following details:

- Endpoints: 192.168.1.43 HELPDESK Port: 30100
- Name: generic_x86-unknown
- Theme: [CaminoAndroid](#)
- Language: [English](#)
- OS: Android
- Buttons: [Live Inspector](#)

At the bottom, there is a toolbar with options: "Show Objects", "Draw: Wireframes and Contents", "Identify Scroll", "Show Device Frame", "Clip to De...", "Propert...", "Toolbox", "Live Ed.", and "Respon...".

Android Emulator - GeneXus-APD4-X865554

Work With Country

- Brazil
- China
- Mexico
- United States
- Uruguay

Work With Country

- Brazil
- China
- Mexico
- United States
- Uruguay

Live Editing

Endpoints: 192.168.1.43 HELPDISK Port: 30100

Name: generic_x86-unknown
Theme: CaminoAndroid
Language: English
OS: Android

Rotate Pan Zoom: Separation: Reset

Show Objects Draw: Wireframes and Contents Identify Scroll Show Device Frame Clip to De

Propert... Toolbox Live Ed... Respon...

The screenshot displays the Genexus IDE interface. On the left, an Android emulator window titled "Android Emulator - GeneXus-APD4-X865554" shows an application named "Work With Country" with a red header and a list of countries: Brazil, China, Mexico, United States, and Uruguay. The main workspace shows a wireframe of the application with a red header and a list of countries, each with a corresponding flag icon. The "Live Inspector" tool is active, showing a list of objects with their names and themes. The "Live Editing" panel on the right displays the endpoints: 192.168.1.43 and HELPDISK, and the port: 30100. The "Live Inspector" panel shows the selected object: generic_x86-unknown, Theme: CaminoAndroid, Language: English, and OS: Android. The bottom status bar includes options like "Show Objects", "Draw: Wireframes and Contents", "Identify Scroll", "Show Device Frame", "Clip to De", "Propert...", "Toolbox", "Live Ed.", and "Respon...".

The screenshot displays the GeneXus Live Editor interface. On the left, an Android emulator window titled "Android Emulator - GeneXus-APID4-X865554" shows an application named "Work With Country" with a list of countries: Brazil, China, Mexico, United States, and Uruguay. The main workspace shows a wireframe of the application with a search bar at the top and a list of country items below. A "Countryname" label points to the list items. On the right, the "Live Editing" panel shows endpoints: 192.168.1.43 HELPDISK, Port: 30100. It also displays device details: Name: generic_x86-unknown, Theme: CaminoAndroid, Language: English, OS: Android. The bottom toolbar includes options like "Show Objects", "Draw: Wireframes and Contents", "Identify Scroll", "Show Device Frame", "Clip to De...", "Propert...", "Toolbox", "Live Ed.", and "Respon...".

The screenshot displays the Genexus IDE interface. At the top, there are several tabs: 'Start Page', 'EventDay', 'Live Inspector', and 'WorkWithDevicesCountry'. Below the tabs is a toolbar with 'Rotate', 'Pan', 'Zoom', 'Separation', and 'Reset' buttons. The main workspace shows a mobile application design with a red header 'Work With Country' and a list of countries: Brazil, China, Mexico, United States, and Uruguay. A tooltip is visible over the Mexico entry, displaying the following information:

- Object: [WorkWithDevicesCountry](#)
- Control: [CountryName](#)
- Class: [AttributeTitle](#)
- ↳ Label Class: [TextBlockTitle](#)
- Size: 340 x 44 dip
- Position: x:54 y:220 dip

On the right side, the 'Properties' panel is open, showing the 'Class' tab. The selected class is 'Attribute/Variable: CountryName'. The properties are organized into sections:

- General**
 - Attribute: **CountryName**
 - Label Position: **None**
 - ReadOnly: **True**
- Link**
 - Autolink: **True**
- Appearance**
 - Class: **AttributeTitle**
 - Visible: **True**
 - Invisible Mode: **Keep Space**
 - Enabled: **True**
 - Format: **Text**
 - Invite Message: **Country Name**
- Virtual Keyboard**
 - Enter Event: **<Platform Default>**
- Control Info**
 - Control Type: **Edit**
 - Auto Grow: **False**
 - Input Type: **Values**

At the bottom of the IDE, there are checkboxes for 'Show Objects', 'Draw: Wireframes and Contents', 'Identify Scroll', 'Show Device Frame', and 'Clip to Device Frame'. The 'Clip to Device Frame' checkbox is checked. The bottom right corner shows icons for 'Propert...', 'Toolbox', 'Live Ed.', and 'Respon...'.

Work With Country

- Brazil
- China
- Mexico
- United States
- Uruguay

Object: WorkWithDevicesCountry
Control: Countryname
Class: AttributeTitle
Label Class: TextBlock
Size: 340 x 44 dip
Position: x:54 y:220 dip

Properties Panel:

- Attribute/Variable: CountryName
- Attribute: CountryName
- Label Position: None
- ReadOnly: True
- Link: Autolink: True
- Appearance: Class: AttributeTitle, Visible: True, Invisible Mod: Keep Space, Enabled: True, Format: Text, Invite Messa: Country Name
- Virtual Keyboard: Enter Event: <Platform Default>
- Control Info: Control Type: Edit, Auto Grow: False, Input Type: Values

Start Page X EventDay X Live Inspector X WorkWithDevicesCountry X

Rotate Pan Zoom: Separation: Reset

Android Emulator - Genexus-API24-X86-5534

Work With Country

Country Name Brazil

Country Name China

Country Name Mexico

Country Name United States

Country Name Uruguay

Work With Country

Country Name Brazil

Country Name China

Country Name Mexico

Country Name United States

Country Name Uruguay

Properties

General Class

Attribute/Variable: CountryName

Attribute	CountryName
Label Position	Left
Label Caption	Country Name
Readonly	True

Link

Autolink	True
----------	------

Appearance

Class	AttributeTitle
Visible	True
Invisible Mod	Keep Space
Enabled	True
Format	Text
Invite Messa	Country Name

Virtual Keyboard

Enter Event	<Platform Default>
-------------	--------------------

Control Info

Control Type	Edit
Auto Grow	False

Show Objects Draw: Wireframes and Contents Identify Scroll Show Device Frame Clip to Device Frame

Prop... Tool... Live E... Respo...

The screenshot displays the Genexus IDE interface. At the top, there are several tabs: 'Start Page', 'EventDay', 'Live Inspector', 'WorkWithDevicesCountry', and 'CarmineAndroid'. The 'CarmineAndroid' tab is active and highlighted with a red circle. Below the tabs is a toolbar with 'Rotate', 'Pan', 'Zoom', and 'Separation' options, along with a 'Reset' button. The main workspace is divided into three sections:

- Emulator:** On the left, an Android emulator window titled 'Android Emulator - Genexus-AP04-1365554' shows a mobile app with a red header 'Work With Country' and a list of countries: Brazil, China, Mexico, United States, and Uruguay.
- Design View:** In the center, a design view of the same app is shown, with a red header and a list of countries. The list items are represented by white boxes with rounded corners.
- Properties Panel:** On the right, the 'Properties' panel is open, showing the 'General' tab for the 'AttributeTitle' class. The 'Preview' section shows a red text block. The 'Colors & Font' section is expanded, showing the 'Class: AttributeTitle' with various properties. The 'Forecolor' property is highlighted with a red box and set to 'PrimaryDark'.

At the bottom of the IDE, there are several checkboxes: 'Show Objects', 'Draw: Wireframes and Contents', 'Identify Scroll', 'Show Device Frame', and 'Clip to Device Frame'. The bottom right corner contains icons for 'Prop...', 'Toolb...', 'Live E.', and 'Respo...'.

Supported changes

- Smart Devices
 - Layouts, Controls, Control Properties, Theme Class Properties
 - Client-side Events
 - Language / Translations
- Web
 - Theme Class Properties

If an attribute is inserted as control in a Layout?

Para finalizar, ¿qué tipo de cambios pueden verse “en vivo” en el dispositivo y la pantalla de Live Inspector?

En aplicaciones Smart Devices, cambios en layouts (filas, columnas de tablas), en propiedades de diseño de los controles, propiedades de las clases del theme de los controles, cambios en eventos que se ejecutan en el cliente –esos no los mencionamos pero también, si hacemos modificaciones en esos tipos de eventos, por ejemplo en un tap asociado a un control, etc., también ese cambio se va a ver reflejado automáticamente cuando estemos ejecutando la aplicación en el dispositivo y antes de grabar esos cambios incluso- y también cambios en los textos relacionados a la traducción.

Como dijimos en la demo, hay cambios tales como en los eventos en el server en Smart Devices (Start, Refresh, Load), o atributos que se agregan como campos a un layout, que no pueden verse en vivo con Live Editing. Esos cambios siguen requiriendo el ciclo normal de Build.

Para Web también existe la posibilidad de utilizar esta funcionalidad solo que está mucho más restringida que en Smart Devices. Y los únicos cambios que se pueden ver sin necesidad de recompilar son cambios en las propiedades de las clases de los temas.

User/Developer Experience