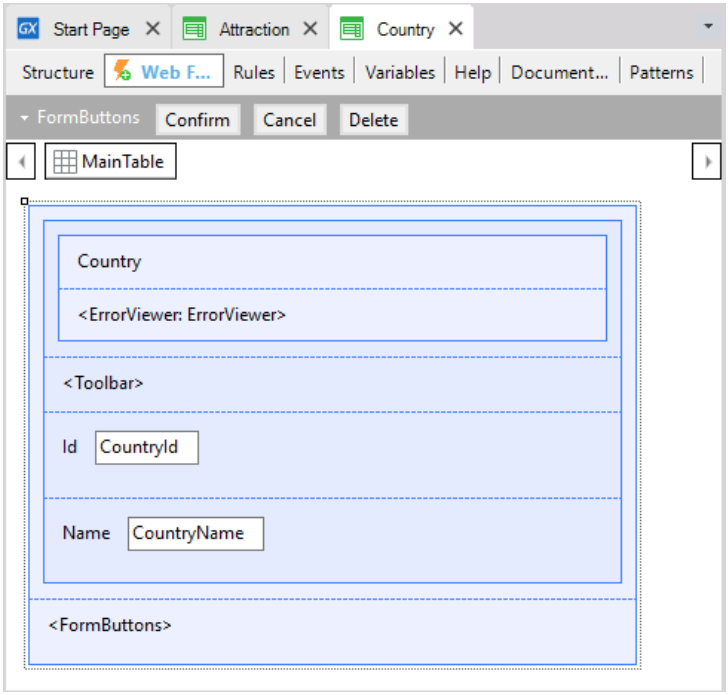
Agregando funcionalidad con patrones

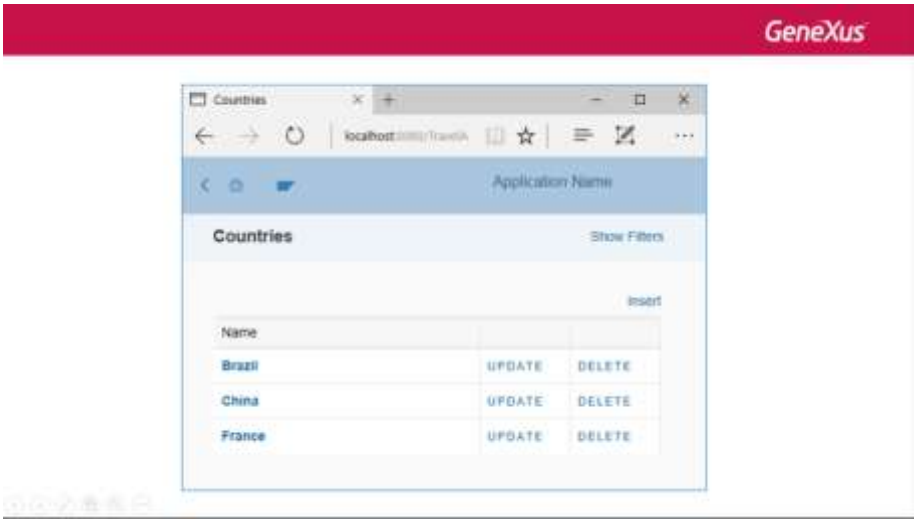
Los patrones nos permiten potenciar nuestra aplicación, agregando nuevas funcionalidades en forma muy sencilla.

Al aplicar un patrón GeneXus **crea por nosotros** todos los objetos necesarios para proveer el funcionamiento deseado, sin que tengamos que programarlos.

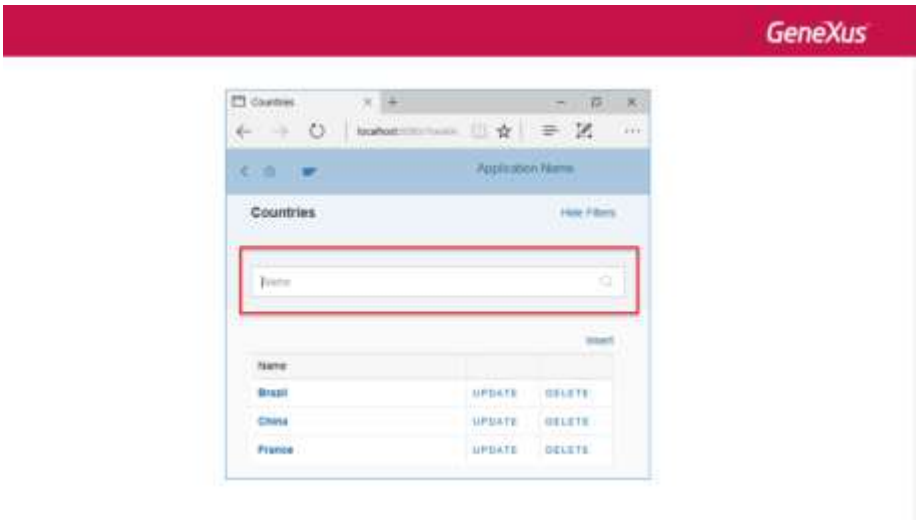
Por ejemplo, supongamos que si bien contamos con la transacción de países:



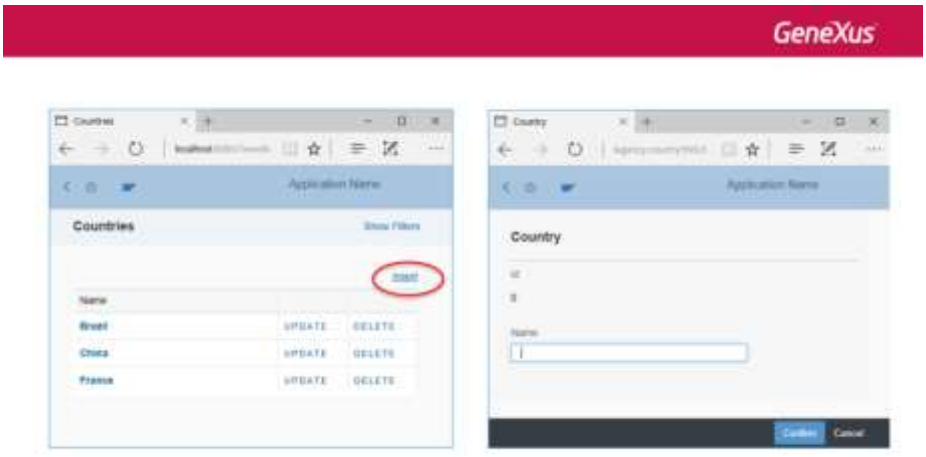
…para hacer ingresos, modificaciones y eliminaciones… quisiéramos contar también con una **página vistosa de consulta de los países** que nos presente todos los países existentes, siguiendo el estilo de diseño Fiori.



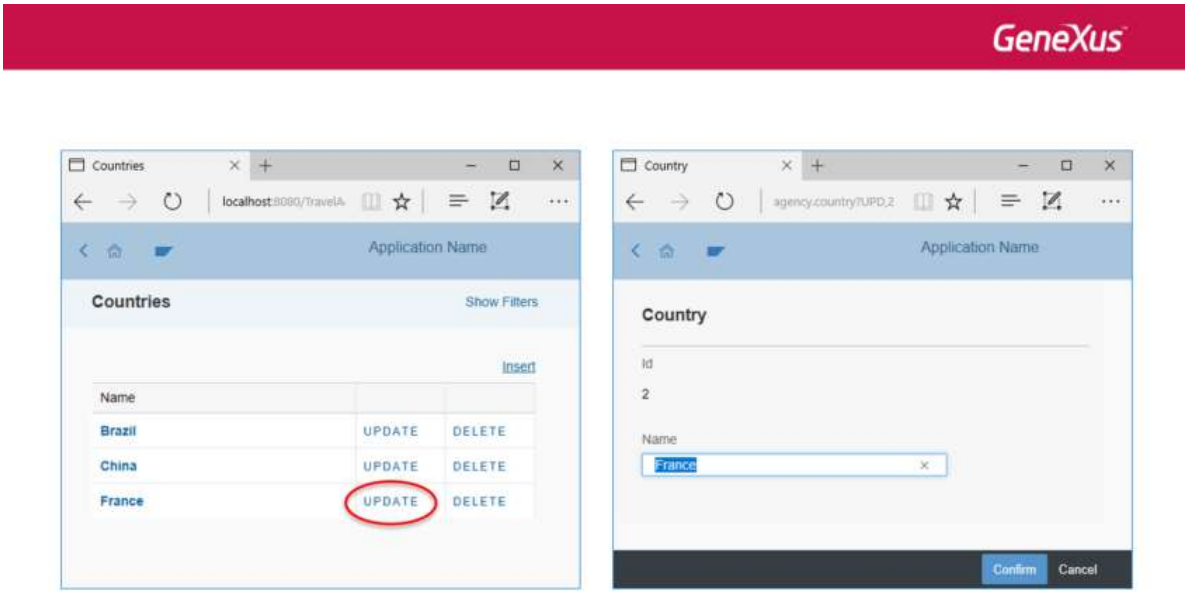
Así, que nos brinde la posibilidad de filtrar por nombre de país:



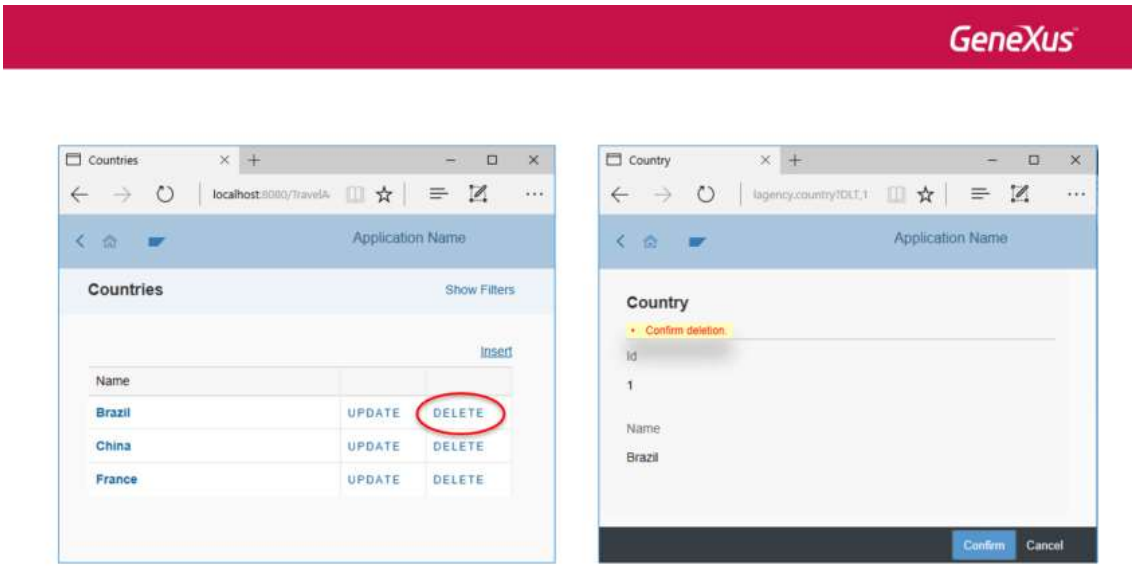
Y que ofrezca ir a la transacción de países si se desea insertar un país nuevo:



…modificar un país existente:



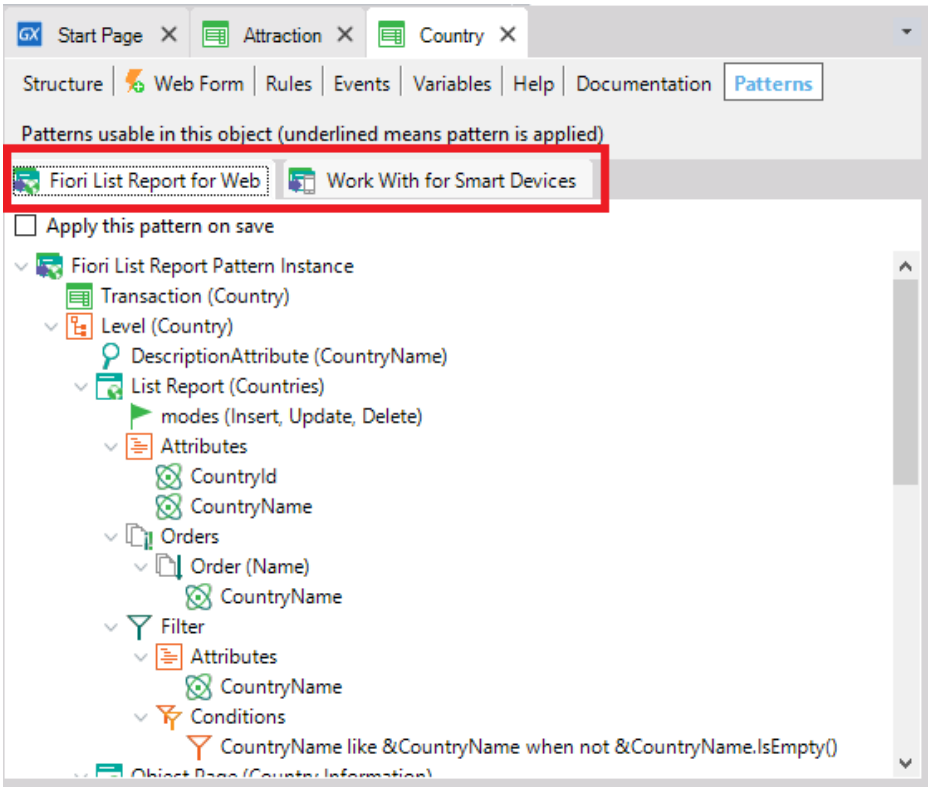
…o eliminarlo.



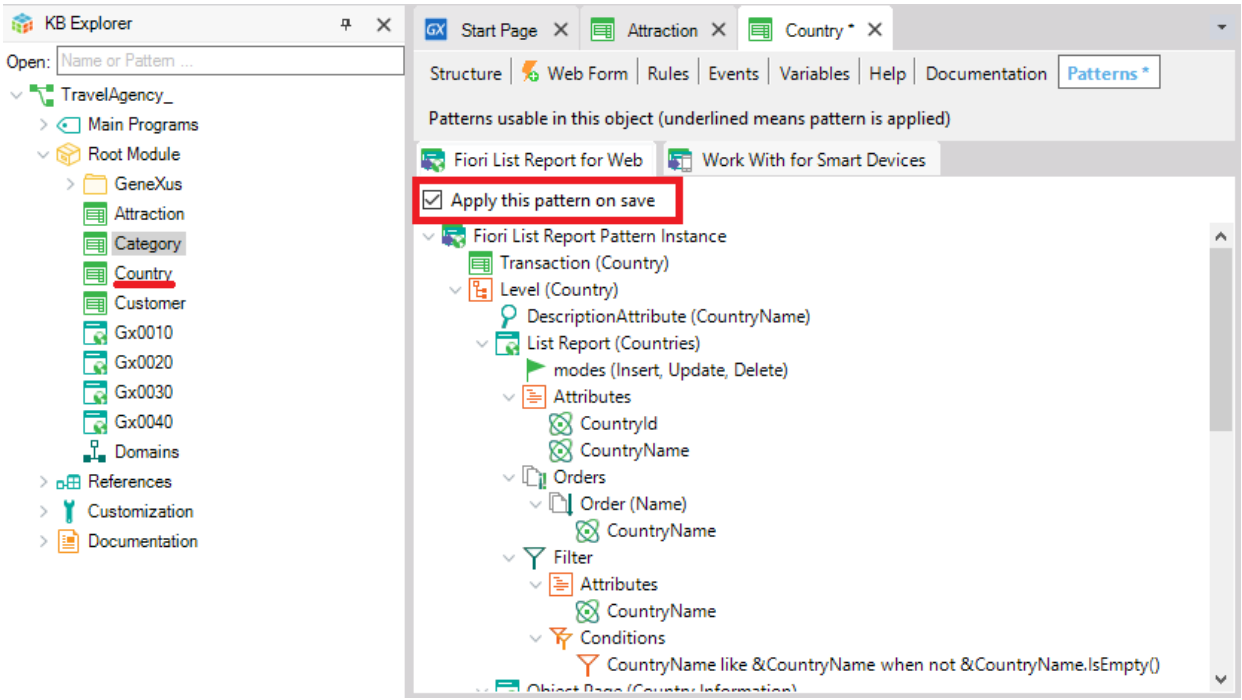
Para obtener esto muy rápidamente, contamos con el patrón “Fiori List Report for Web” y vamos a ver cómo aplicarlo a la transacción Country.

Lo tenemos disponible tanto para aplicaciones web como para aplicaciones para Smart Devices. Para ellas se llamará “Work with for Smart Devices” pero se tratará esencialmente del mismo patrón. Lo veremos luego.

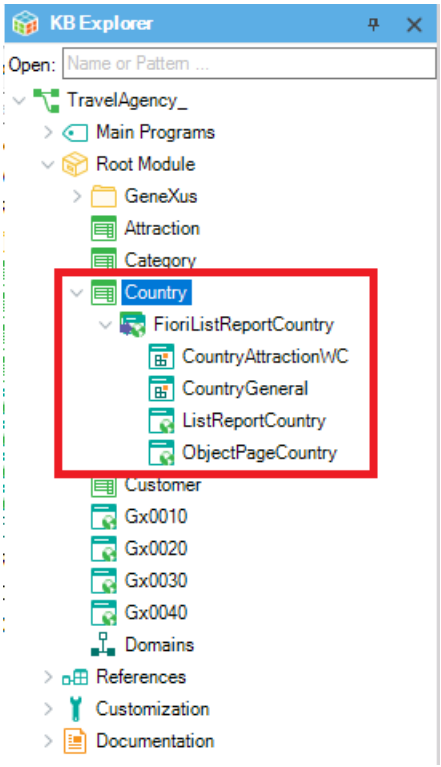
En la solapa Patterns de la transacción Country podemos ver ambos patrones.



Para obtener el patrón web, solamente necesitamos elegir esta solapa, y cliquear sobre “Apply this pattern on save” y grabar.

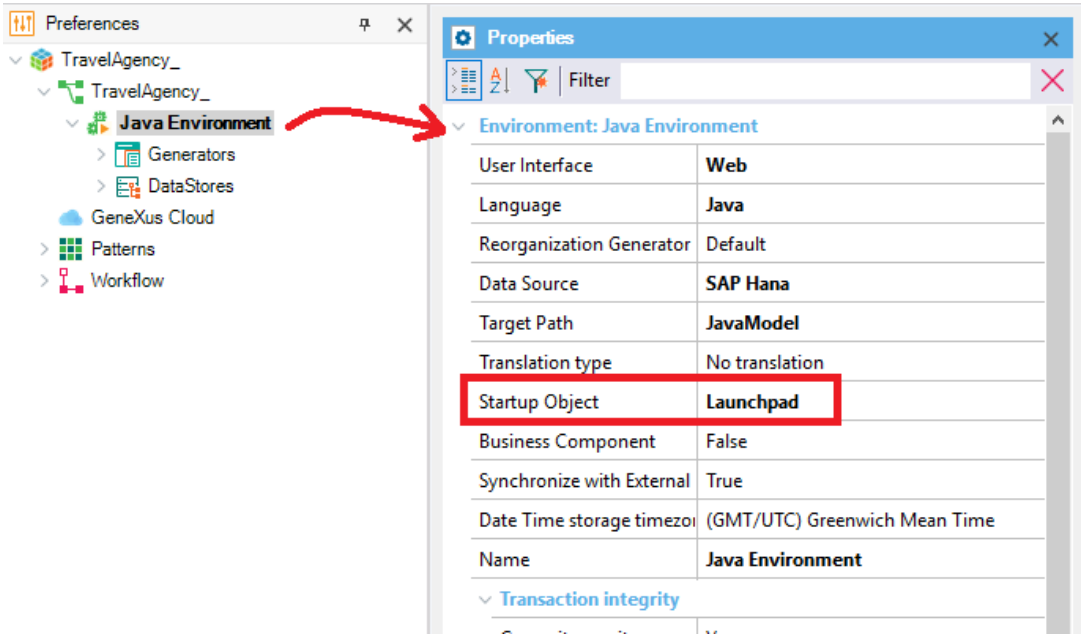


Ahora, si ubicamos la transacción Country aquí:



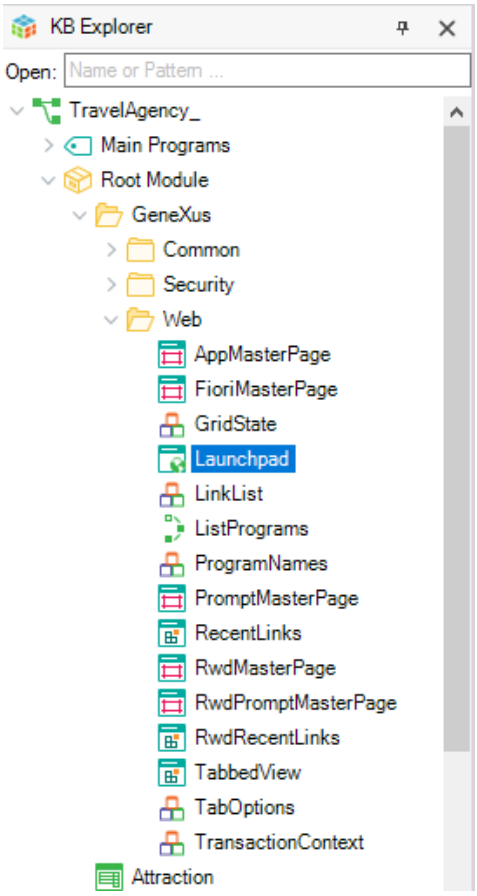
…vemos que bajo la transacción figuran varios objetos, que fueron creados por GeneXus tras haber aplicado el patrón. Estos serán los objetos que implementarán las pantallas web del patrón, que veremos a continuación.

Antes de presionar F5, observemos que otra vez GeneXus configuró como Startup object al objeto Launchpad:

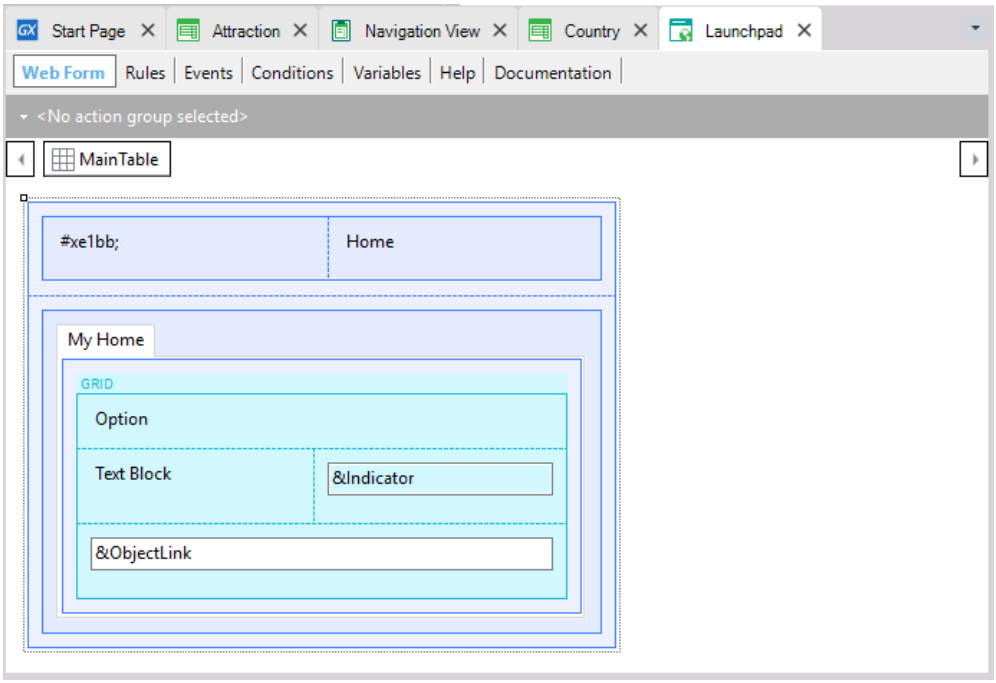


…por lo que ya no se compilará y ejecutará el Developer Menu sino este objeto especial, que contendrá todos los Fiori List Reports que apliquemos a transacciones de la base de conocimiento.

Ese objeto Launchpad es un objeto GeneXus creado automáticamente, que se encuentra aquí:



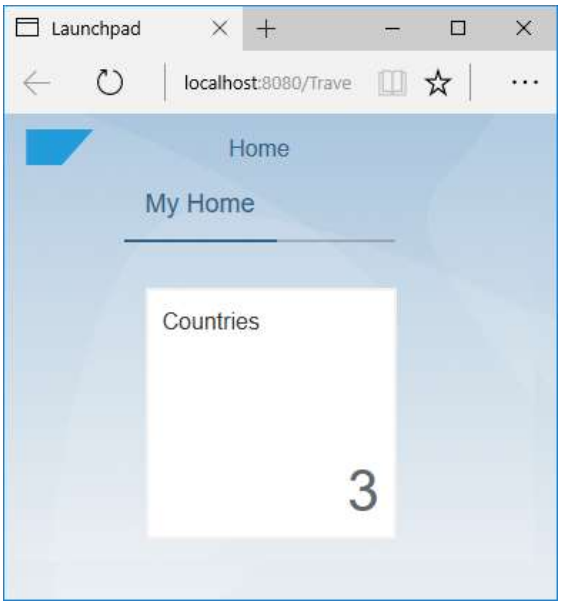
Es del tipo “Web Panel”, es decir, implementa una pantalla web:



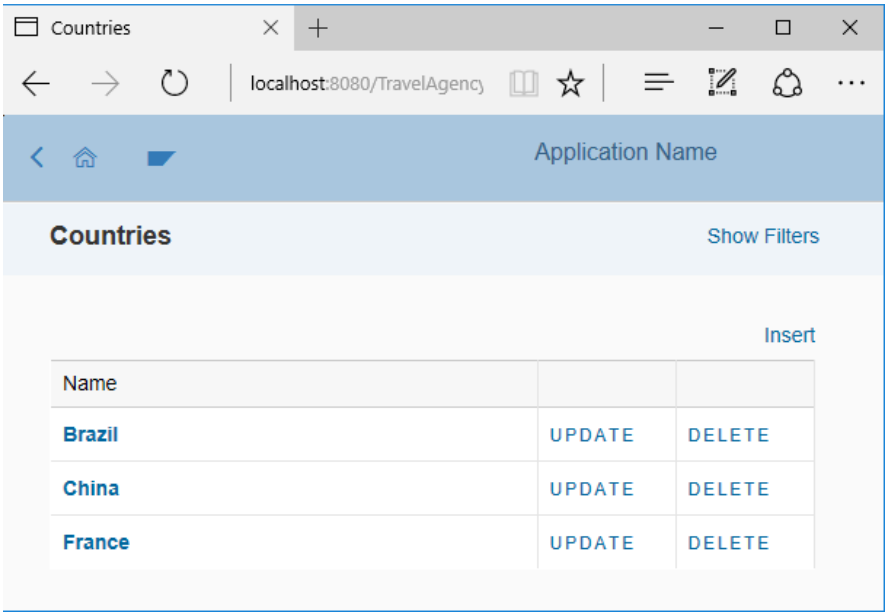
Nosotros también podremos crear paneles web a través de objetos Web panel, tal como lo hizo automáticamente GeneXus para este objeto. No lo veremos en este curso pero puede buscarlo en el Curso GeneXus 15 (no especializado en SAP). Vale para SAP exactamente lo mismo que allí se muestra.

Ahora presionemos F5 **para ver en ejecución todo lo que se ha generado automáticamente al aplicar el pattern**.

Se nos despliega en el navegador el Launchpad con diseño Fiori, ofreciéndonos una tarjeta con un link para trabajar con países. Y nos está indicando, además, la cantidad de países registrados:

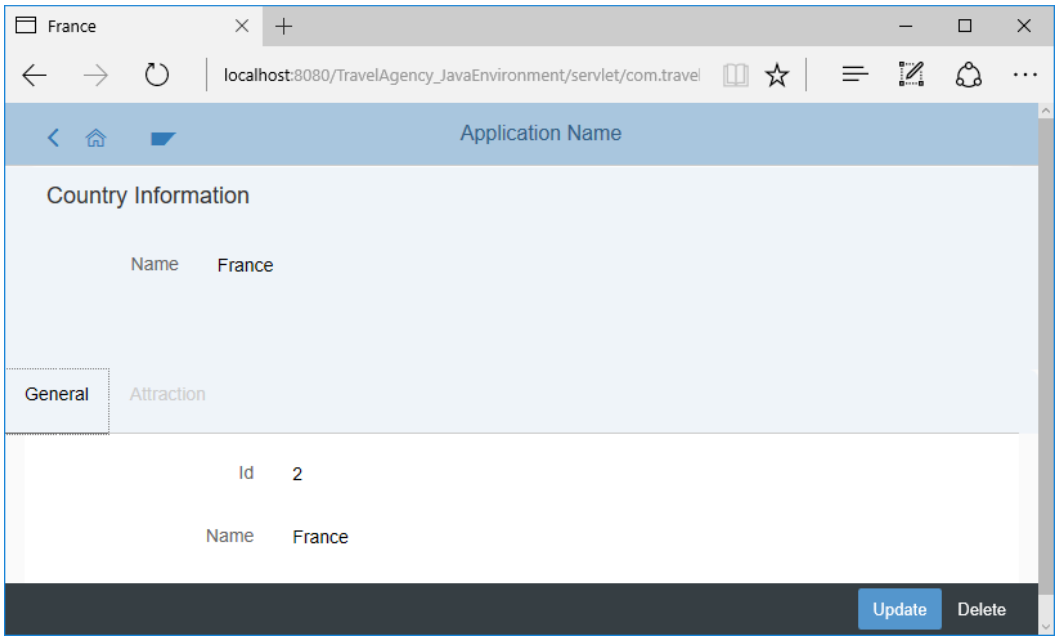


Aquí vemos el Listado de los tres países existentes:

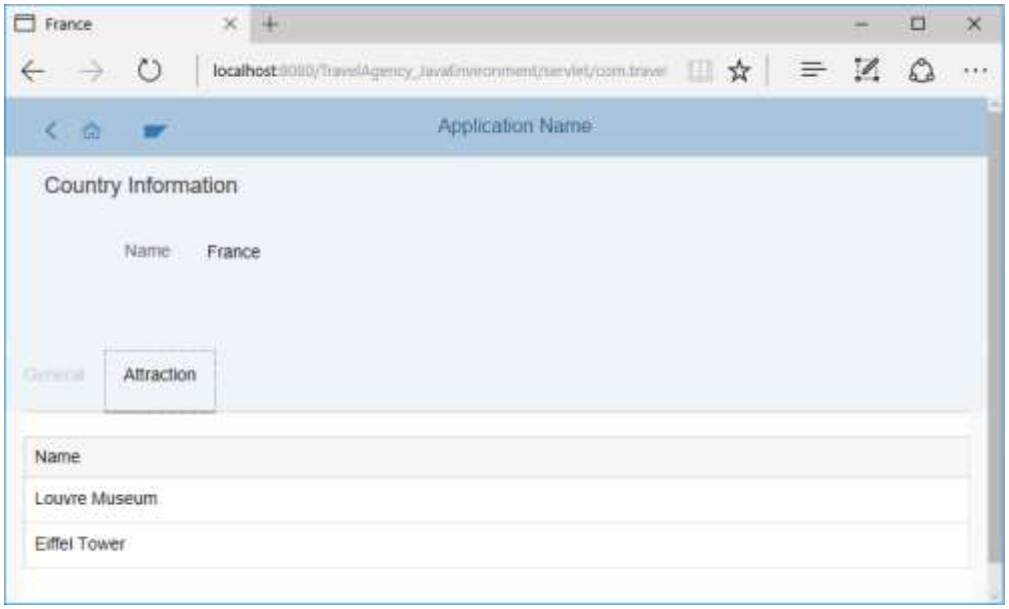


Sobre el que podemos realizar operaciones tales como:

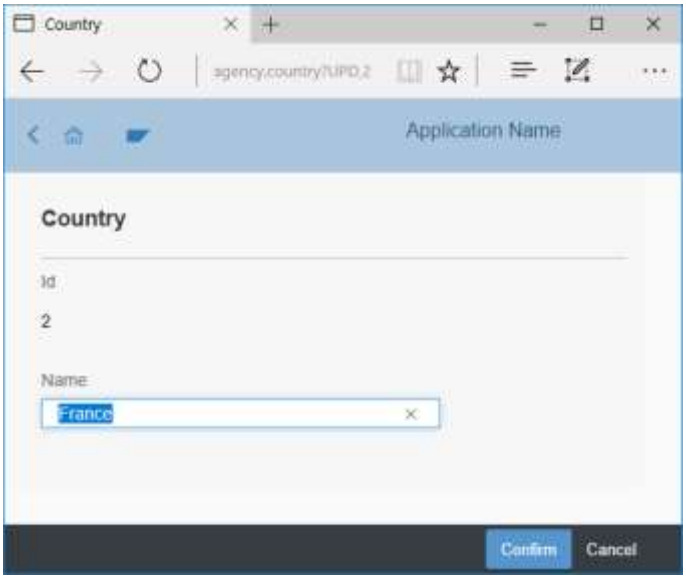
- Visualizar la información completa de un país, por ejemplo, Francia:



Donde como vemos, tenemos la pestaña General, que muestra la información general del país y presenta botones para Actualizar esa información o Eliminar el país, y luego una pestaña Attraction que va a mostrar todas las atracciones registradas para el país que estamos visualizando:



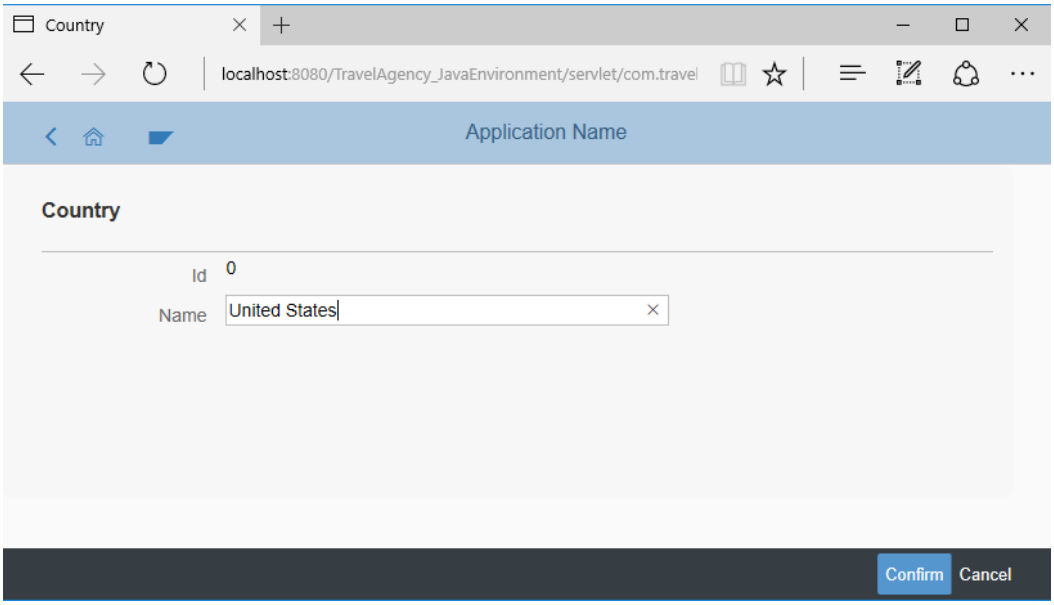
Tenemos también la posibilidad de directamente actualizar la información de un país:



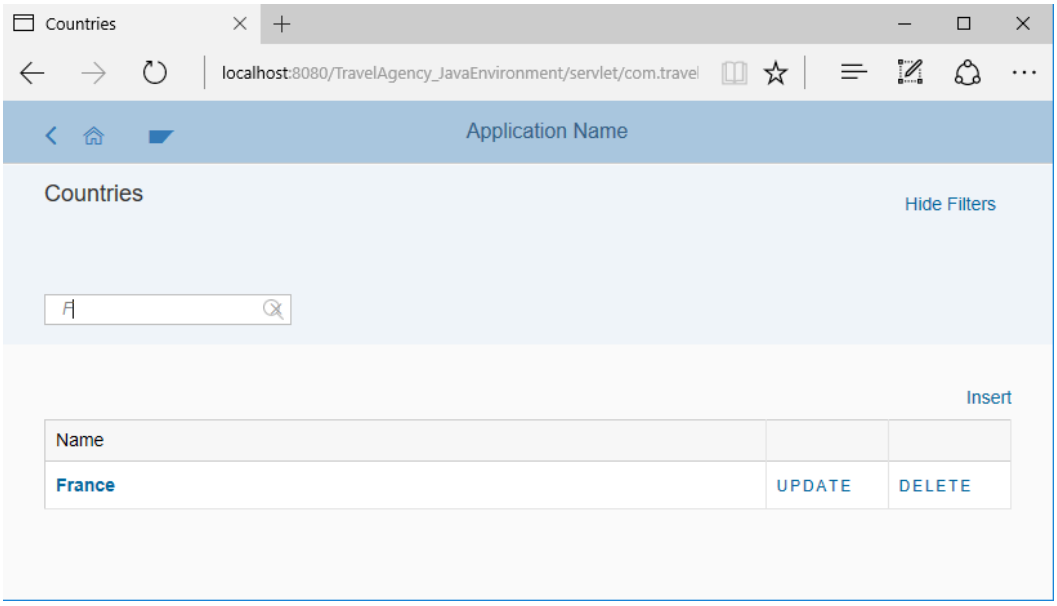
Esta pantalla es la de la transacción Country, donde han desaparecido los botones de navegación, porque estamos posicionados en el país elegido.

De la misma manera podemos eliminar el país. Vemos que como está llamando a la transacción en modo DELETE, y el país tiene atracciones relacionadas (las vimos recién: el museo Louvre y la torre Eiffel), no nos permite la eliminación. Si intentáramos con Brasil, que no tiene atracciones, sí podremos eliminar, simplemente Confirmando… Cancelamos, para no eliminar, y vemos que nos vuelve a llevar al listado de países.

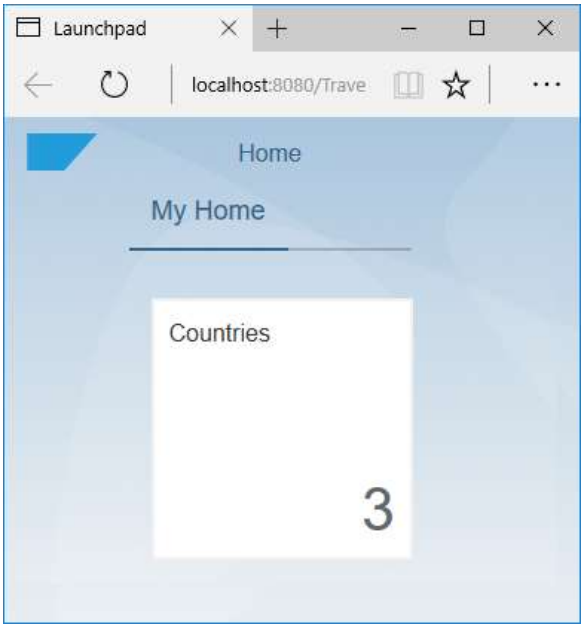
También podemos insertar un nuevo país. Aquí también se abre la transacción. Agreguemos United States:



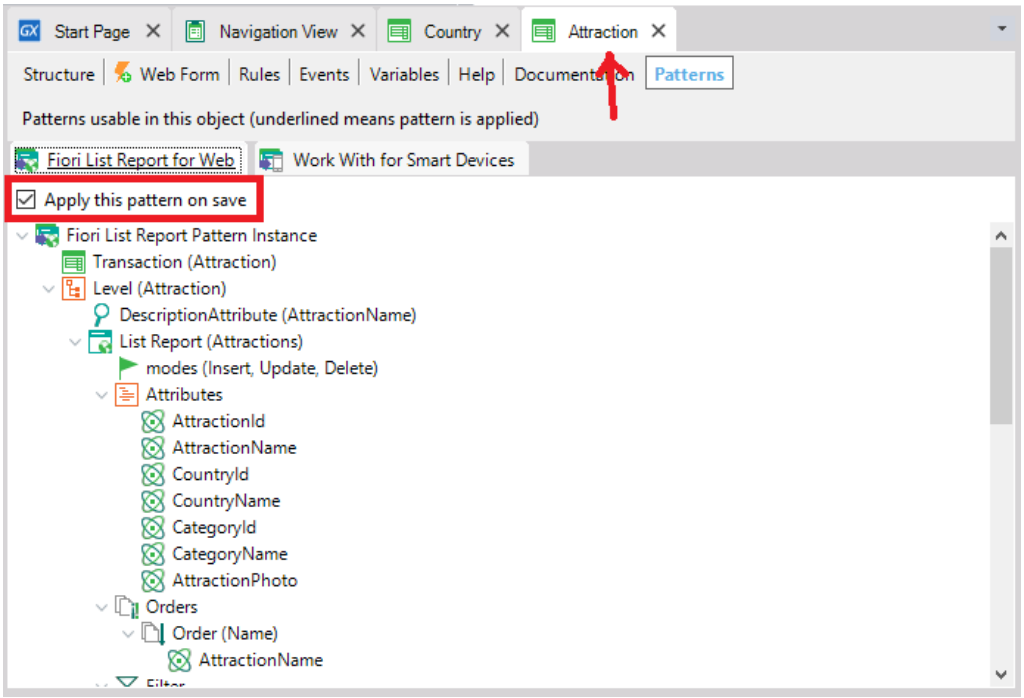
Podemos realizar filtros sobre los países mostrados, por nombre:



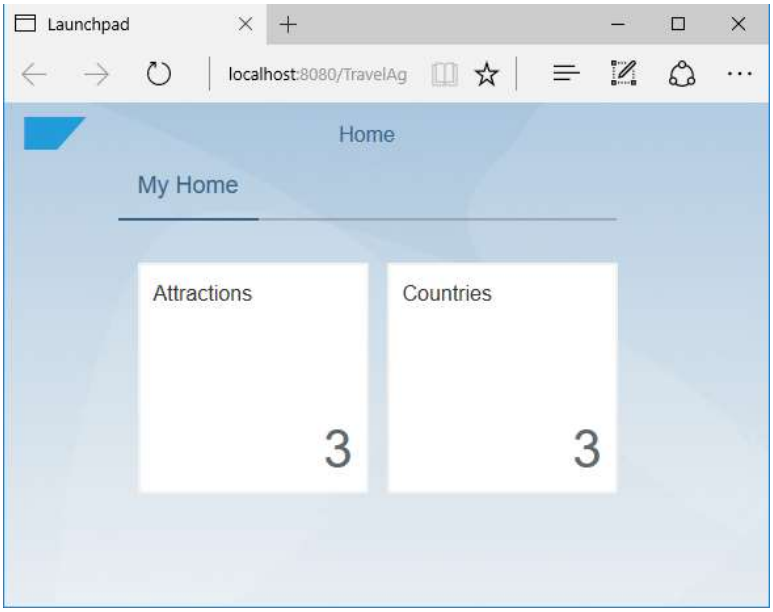
Aquí volvemos a la página Home, que es el launchpad:



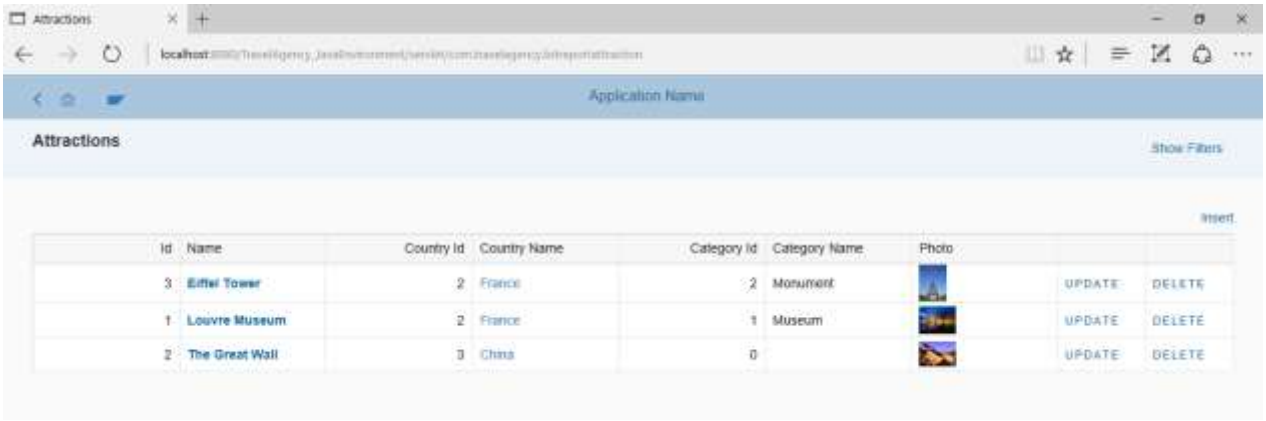
Ahora, bien, así de fácil fue aplicar el patrón a la transacción Country, y ahora lo haremos para la transacción Attraction.



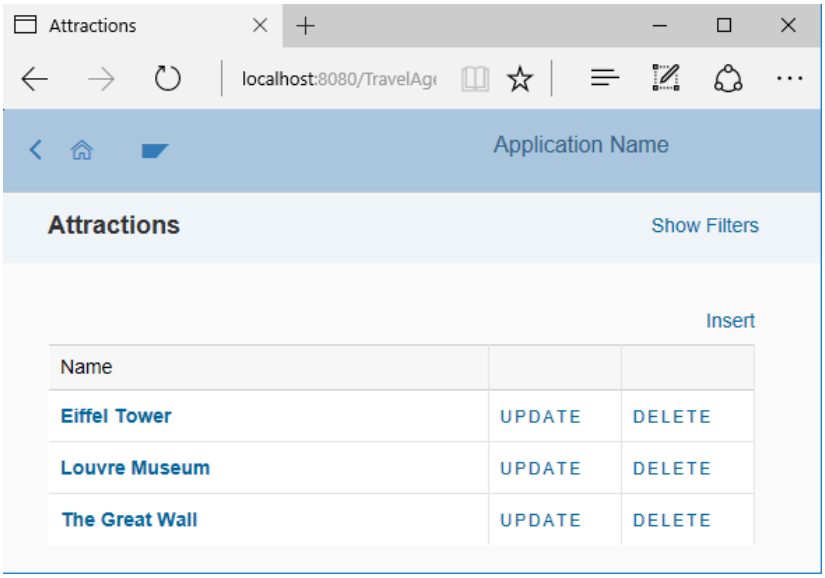
Pulsamos F5.



Vemos que al igual que en el caso anterior, se nos ofrece la tarjeta para “trabajar con atracciones” con la cantidad de atracciones registradas…

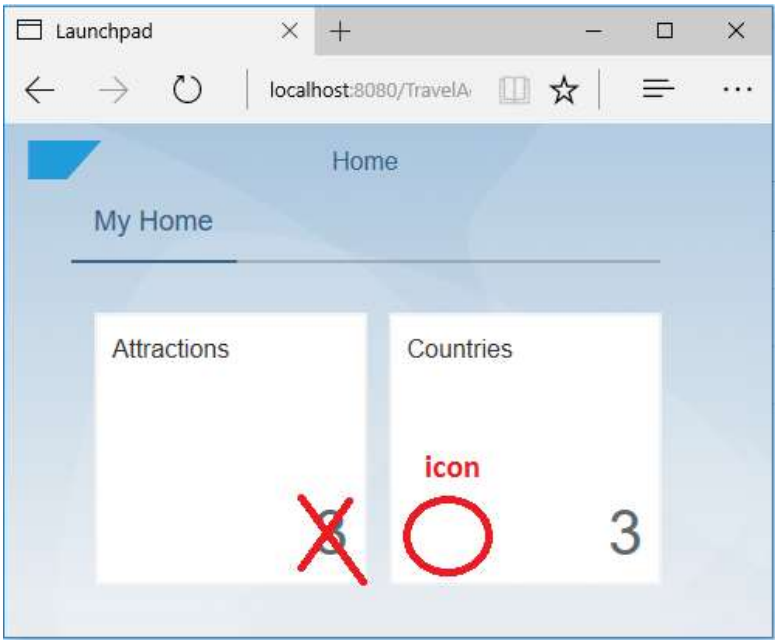


Vemos otra vez cómo, si modificamos las dimensiones de la pantalla, la información que se muestra se ajusta. En este caso, de mostrarse todos los atributos de Attraction se pasa a mostrar sólo el nombre:

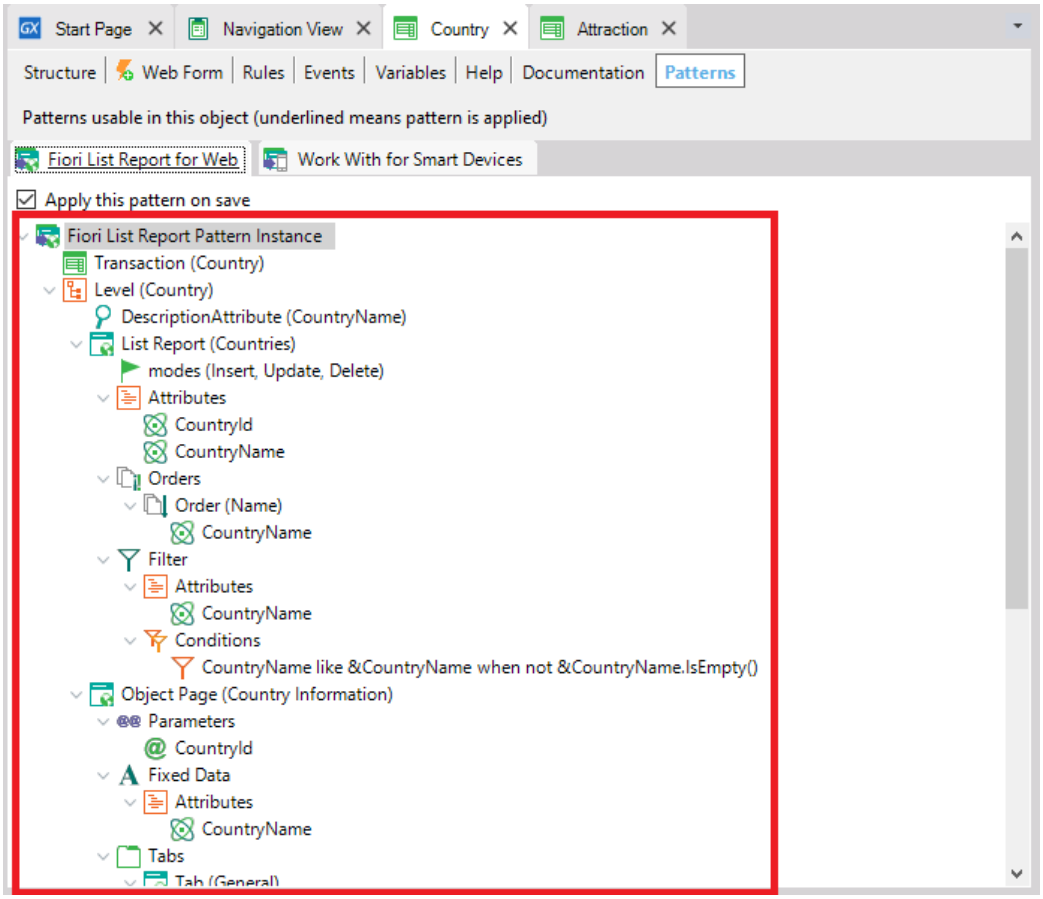


Contamos con el mismo estilo de consulta que vimos para “trabajar con países”…

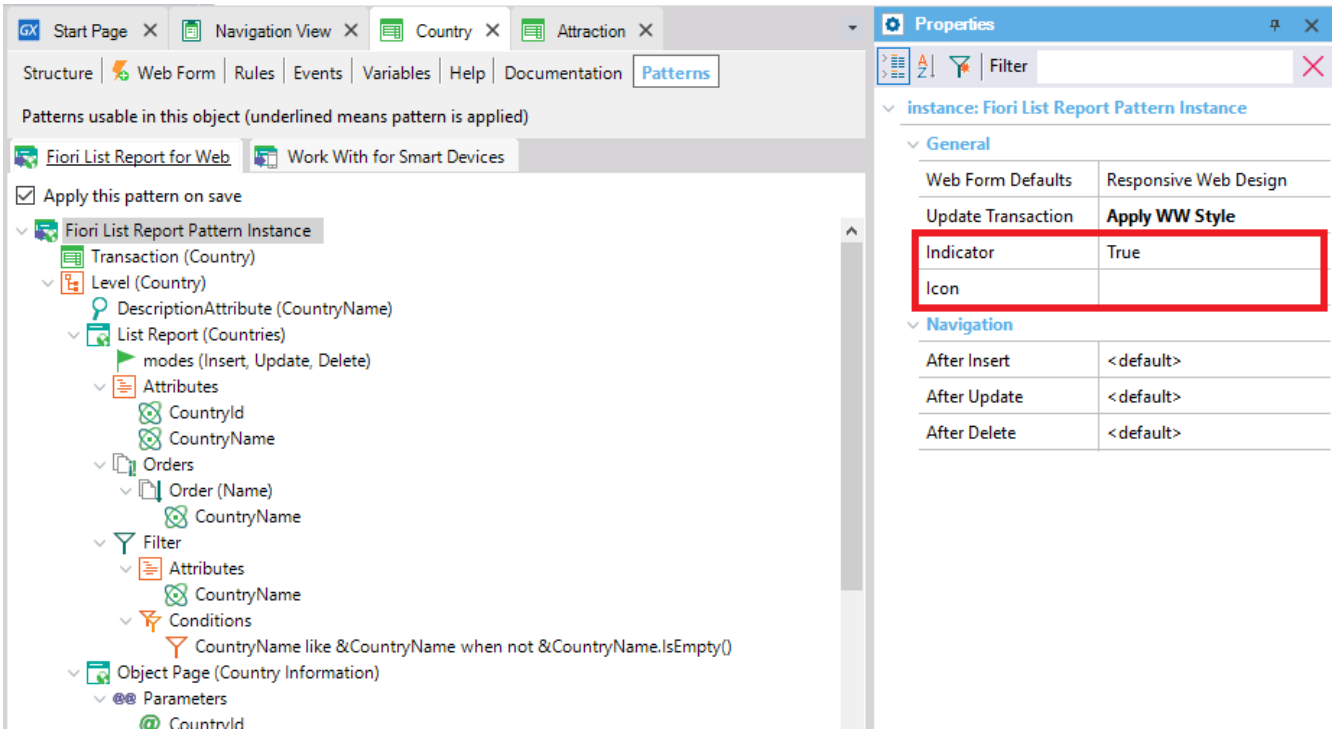
Lo que se muestra y el comportamiento de todas estas pantallas puede personalizarse. Por ejemplo, supongamos que no desea verse el indicador de cantidad de atracciones en el Launchpad, y en la tarjeta correspondiente a los países desea verse un ícono que los represente, aquí:



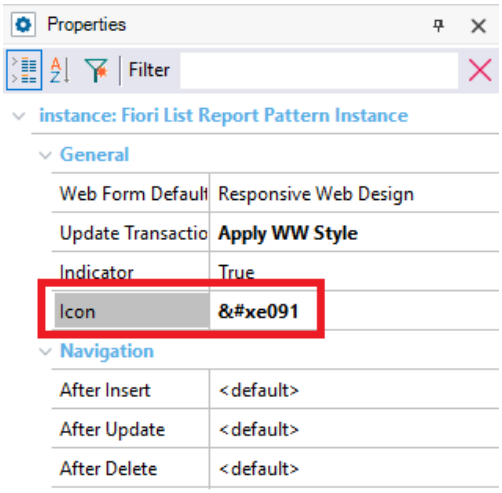
Hasta ahora solamente nos habíamos encargado de marcar “Apply this pattern on save” **sin detenernos a observar este árbol de valores configurables**:



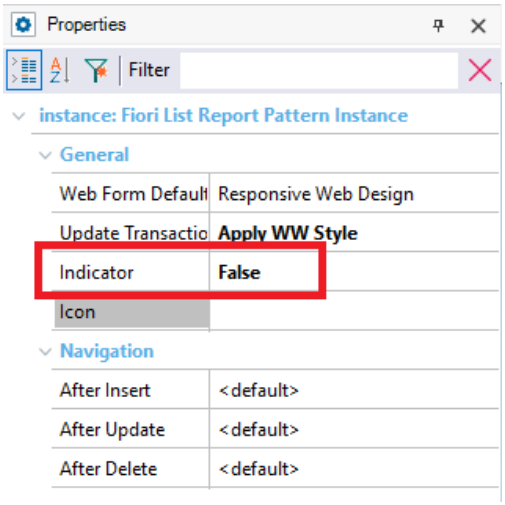
Es aquí donde está determinado todo lo que vemos en ejecución. Por ejemplo, posicionándonos en la raíz veremos entre las propiedades del pattern aplicado a esta transacción Country, estas dos:



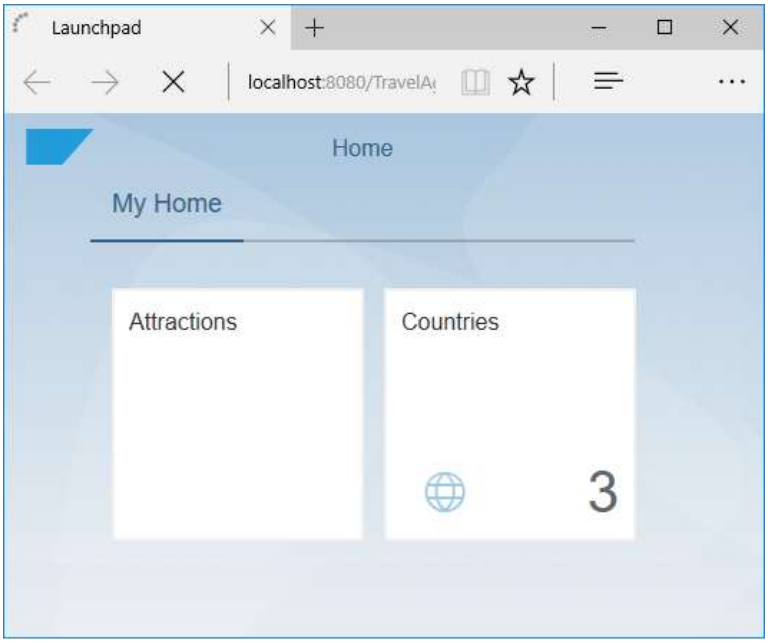
La primera es la que especifica si se mostrará en la tarjeta del launchpad la cantidad de países o no. Por defecto está en true y por eso veíamos ese número en ejecución. La segunda permite especificar un ícono. Aquí se especifica el código del ícono de acuerdo a las guidelines de diseño Fiori:



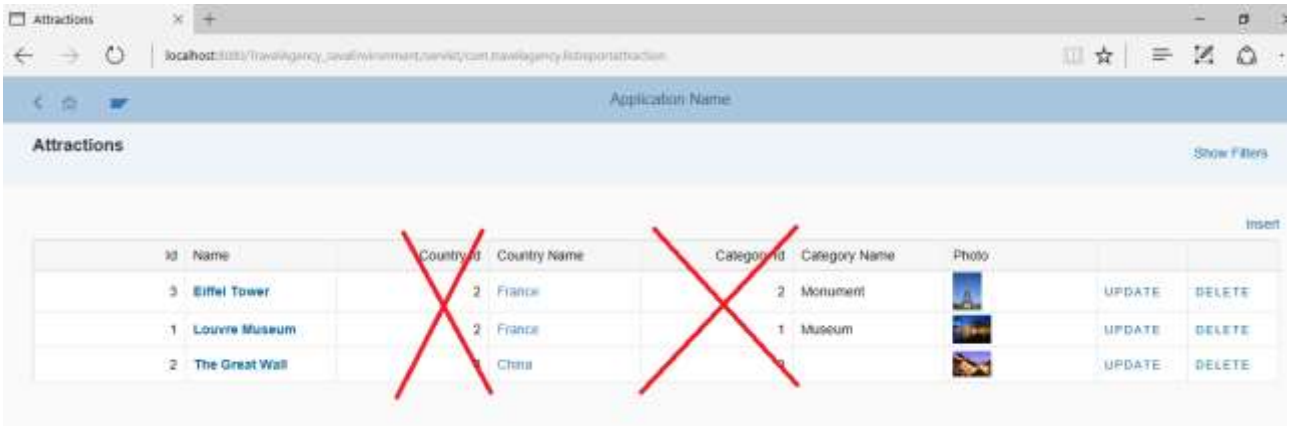
En el pattern de Attraction quitamos el número:



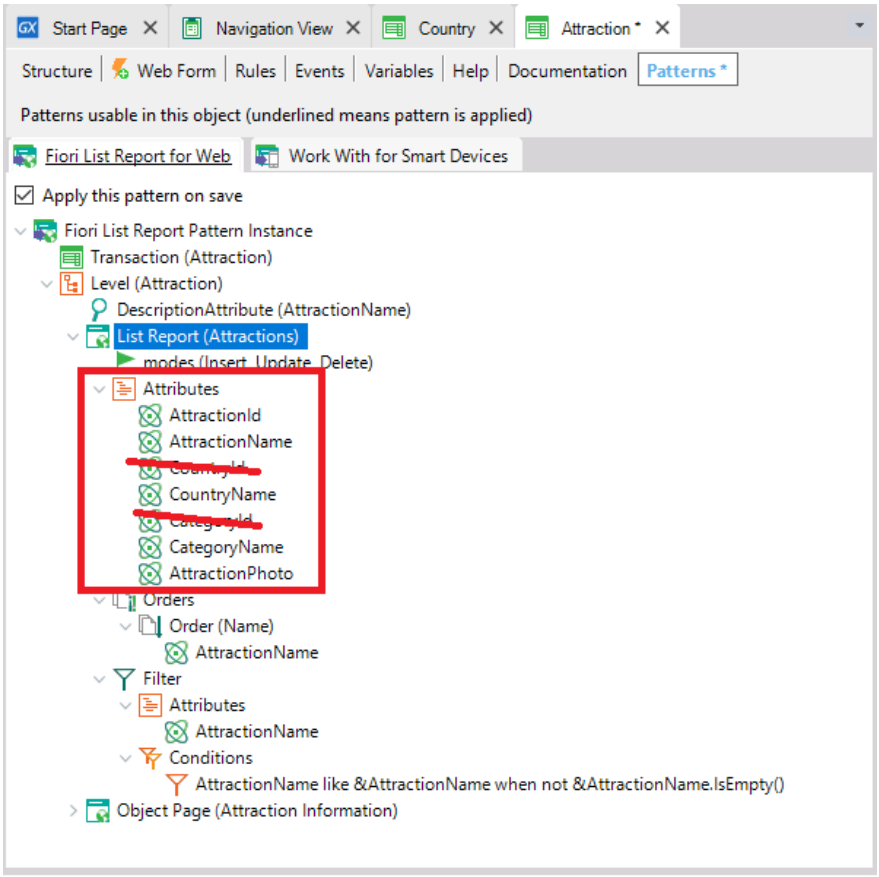
Presionemos F5…
Y así lo vemos en ejecución:



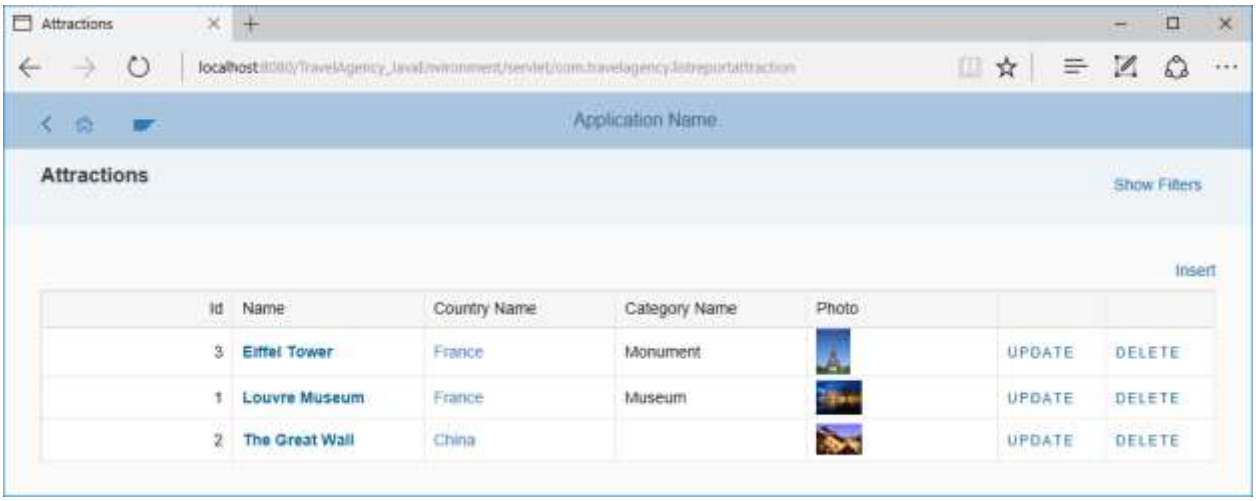
Por otro lado, supongamos que en el List Report de Attraction no queremos que se despliegue ni el identificador de país ni el de categoría:



Para ello, vamos al nodo List Report y bajo el subnodo Attributes vemos todos los atributos que están apareciendo en el form cuando el tamaño de pantalla lo permite. Eliminamos los que no deseamos ver nunca:

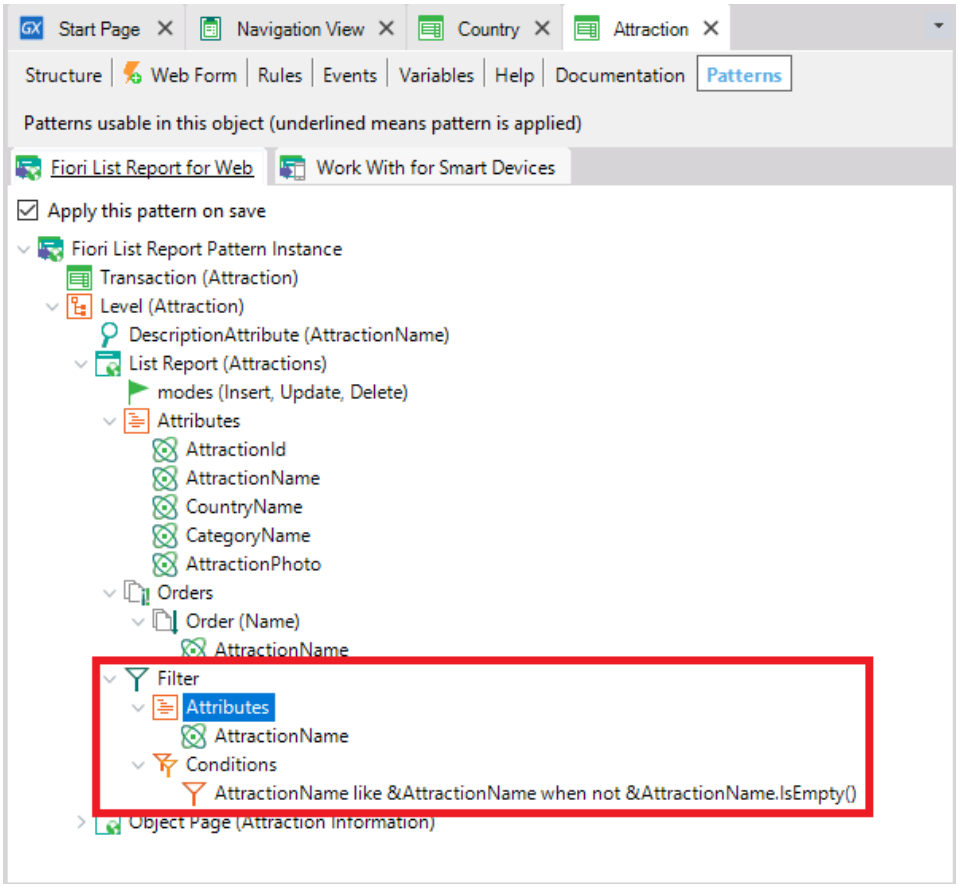


Y presionamos F5.

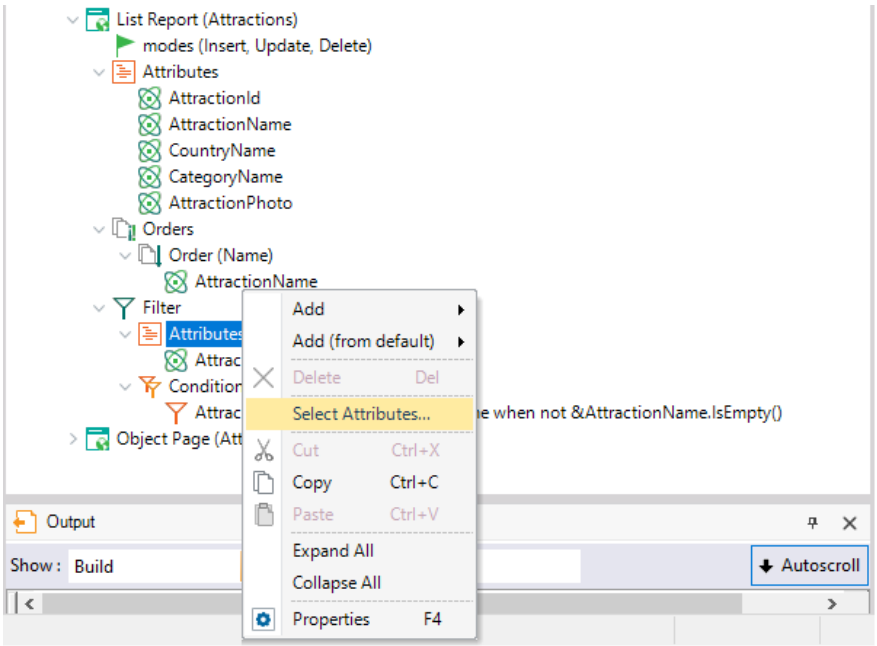


Ahora supongamos que queremos ofrecer al usuario no solo la posibilidad de filtrar lo que se muestra en este grid por nombre de atracción, sino también por país. Por ejemplo, queremos ver todas las atracciones de Francia.

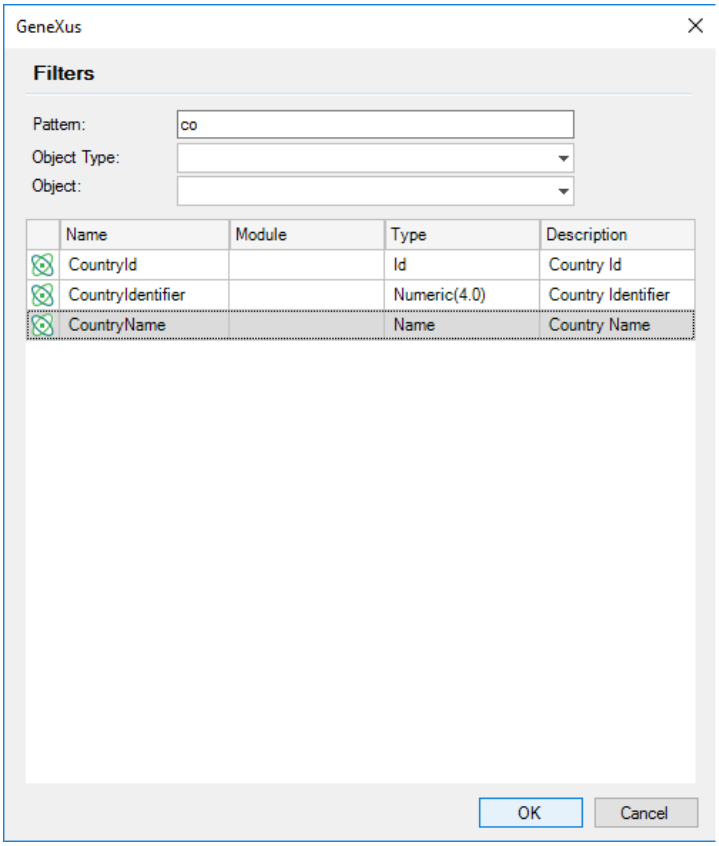
Para ello, vamos al nodo Filter:



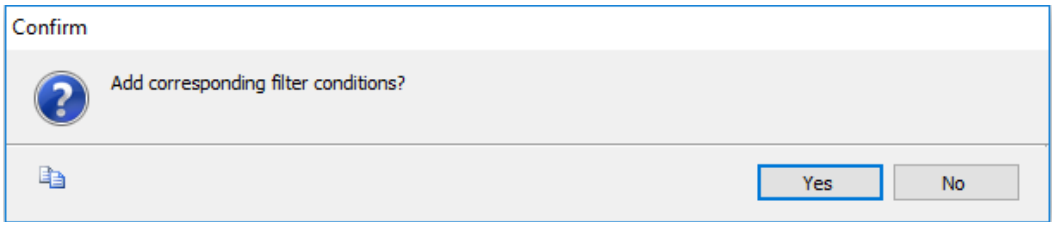
Y sobre el nodo Attributes elegimos Select Attributes:



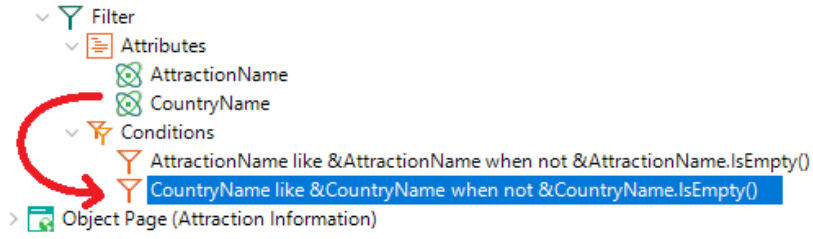
Y allí CountryName:



Se nos preguntará si queremos agregar las correspondientes condiciones de filtro, y diremos que sí, porque es justamente eso lo que queremos:

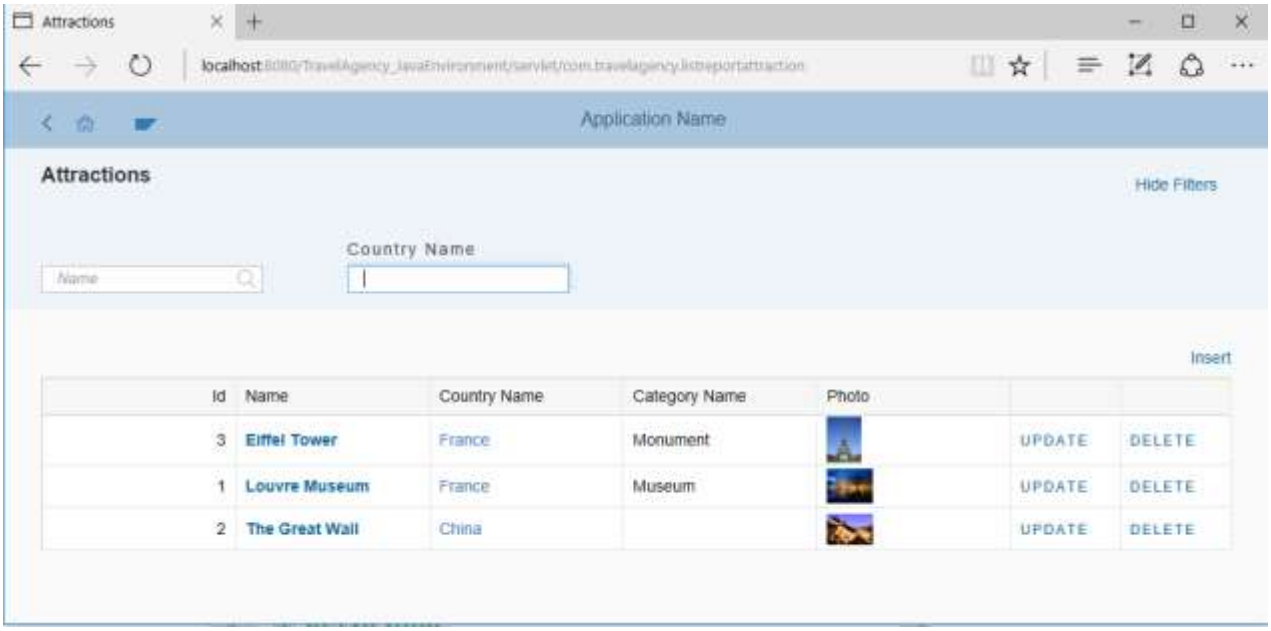


Y vemos cómo agregó un filtro por ese atributo:

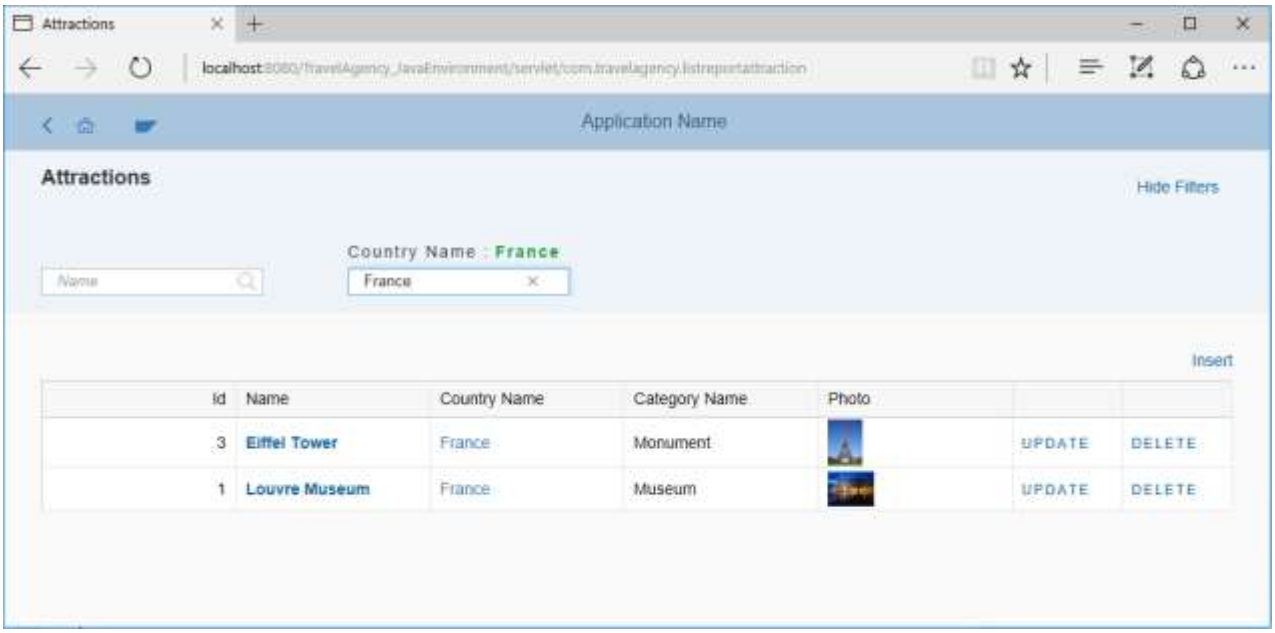


Presionemos F5 para verlo en ejecución.

Vemos cómo aparece aquí esta variable para filtrar por país:

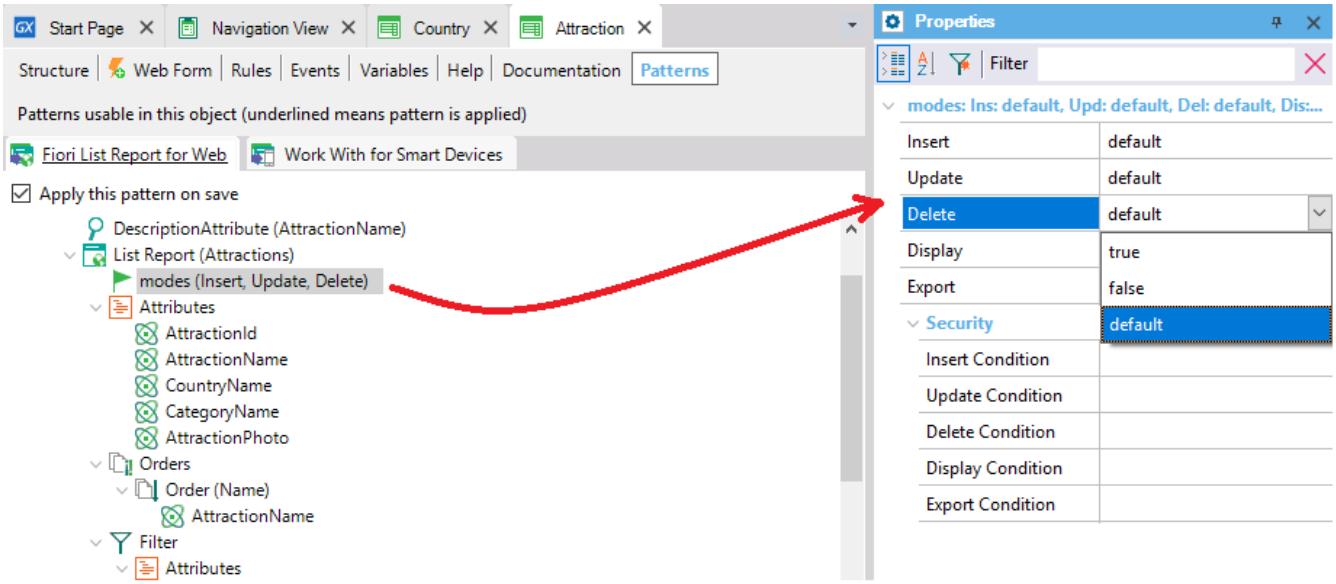


Por lo que si digitamos France:

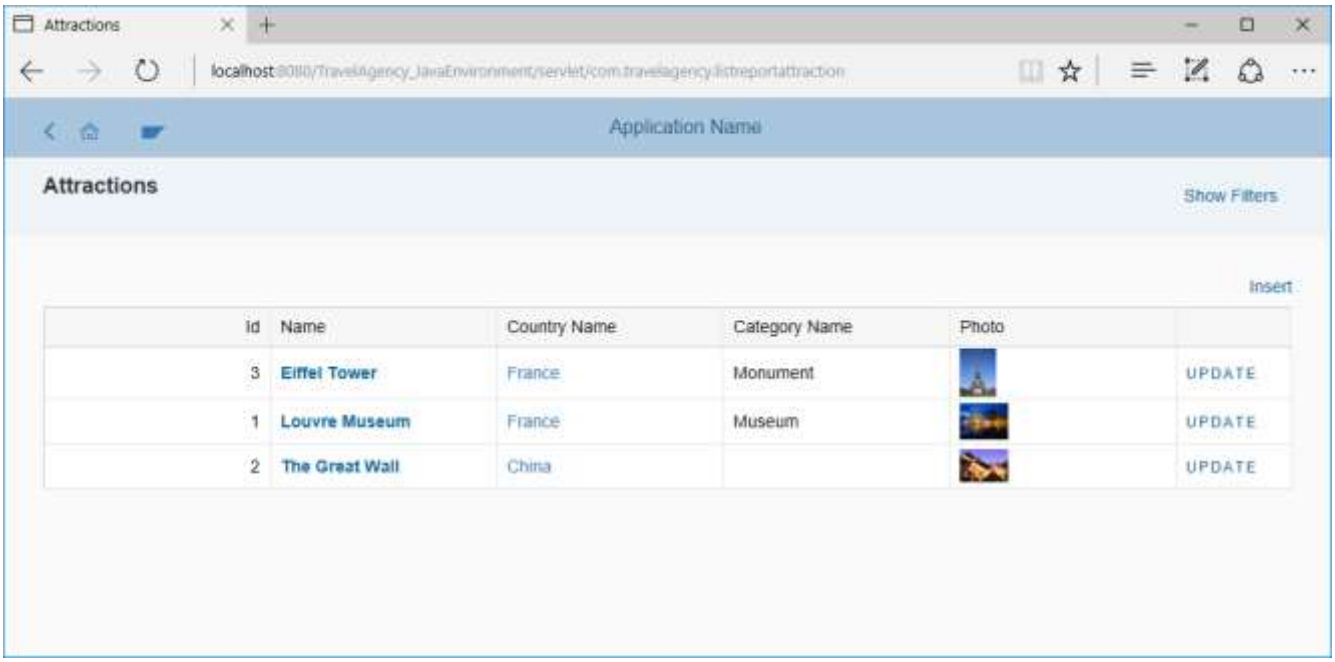


Aplica el filtro.

También podríamos querer deshabilitar alguna o todas las acciones Insert, Update y Delete para este panel. Para ello nos posicionaríamos en el nodo “modes”, y cambiaríamos el valor de la propiedad correspondiente. Estos valores “default” están especificados en un lugar centralizado, que luego veremos. Evidentemente el “default” de las propiedades Insert, Update y Delete está siendo true. Si quisiéramos deshabilitar la eliminación, configuraríamos “false” para el valor de esta propiedad:

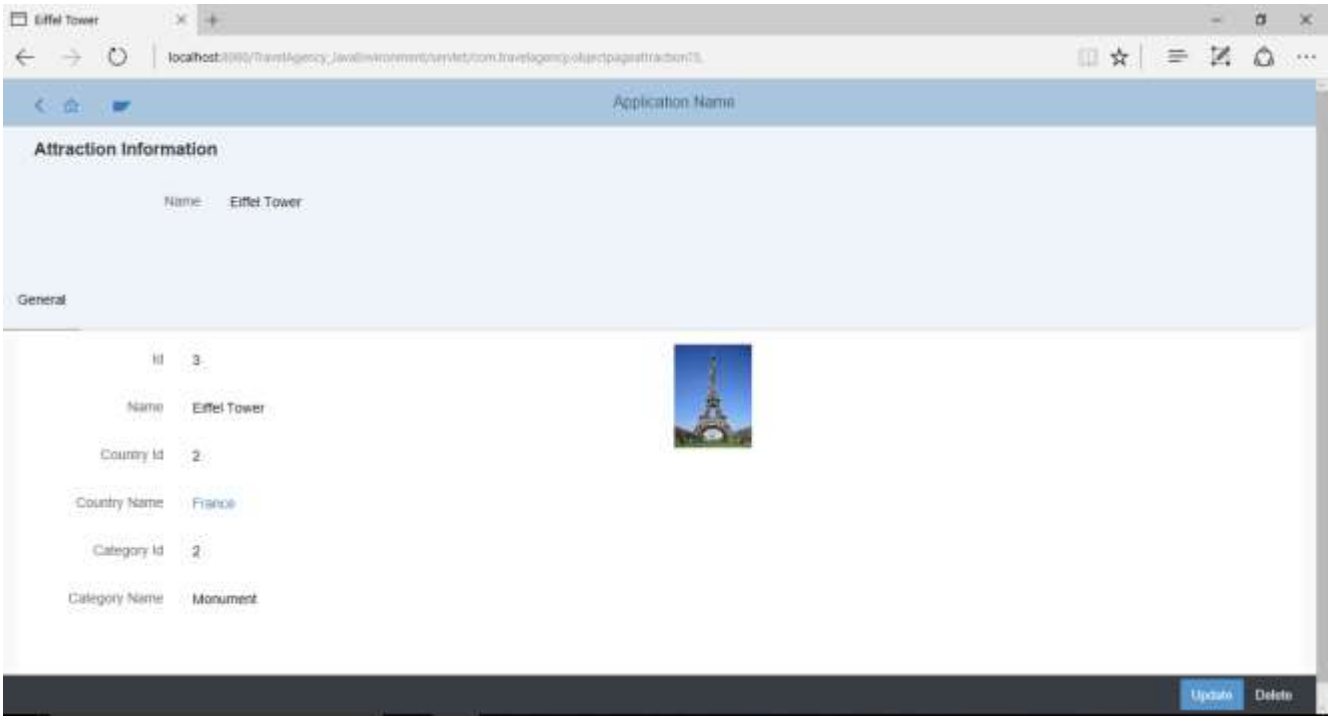


Y con ello se eliminará la columna del DELETE:

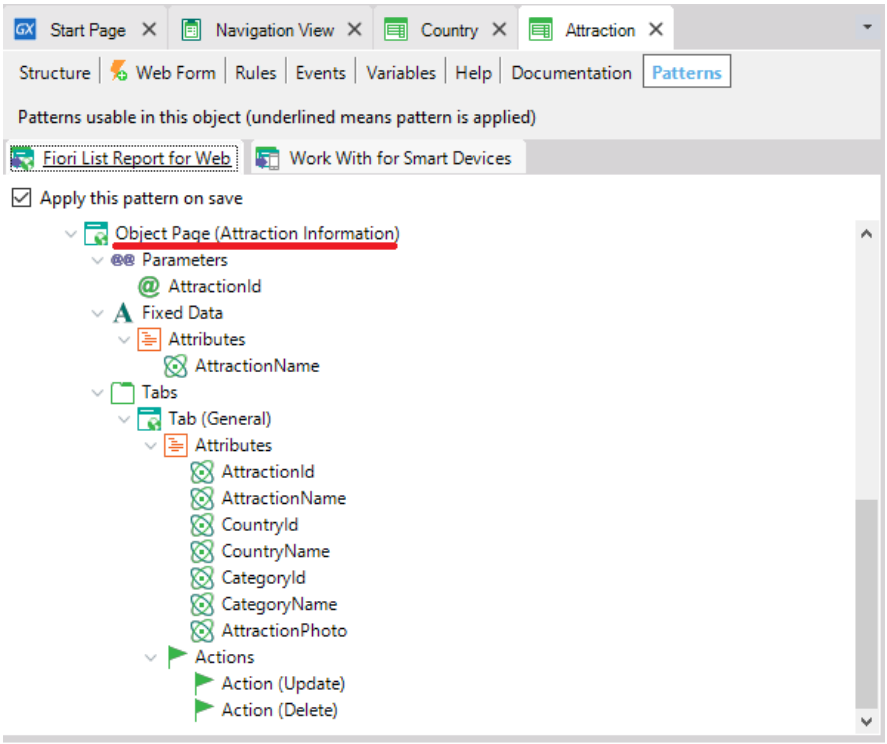


Tenemos más posibilidades de personalización que le invitamos a descubrir.

Aquí solamente vimos algunas para el panel del List Report de Attractions, pero también podemos personalizar el panel que nos despliega la información de la atracción elegida:

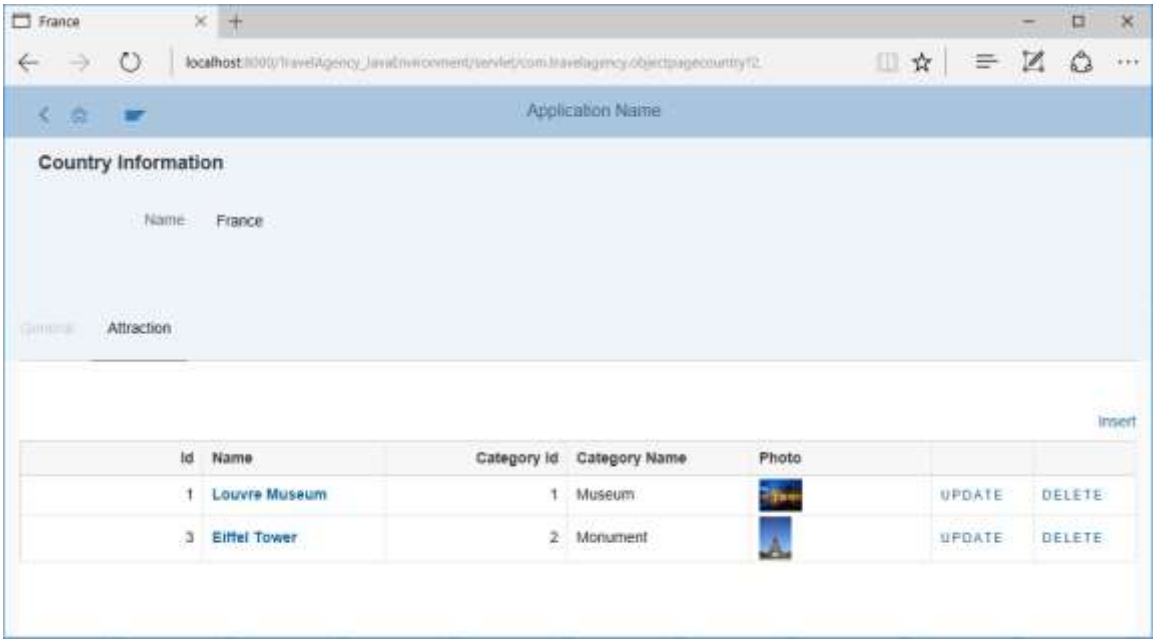


Las definiciones de este panel son las que se encuentran bajo este nodo:



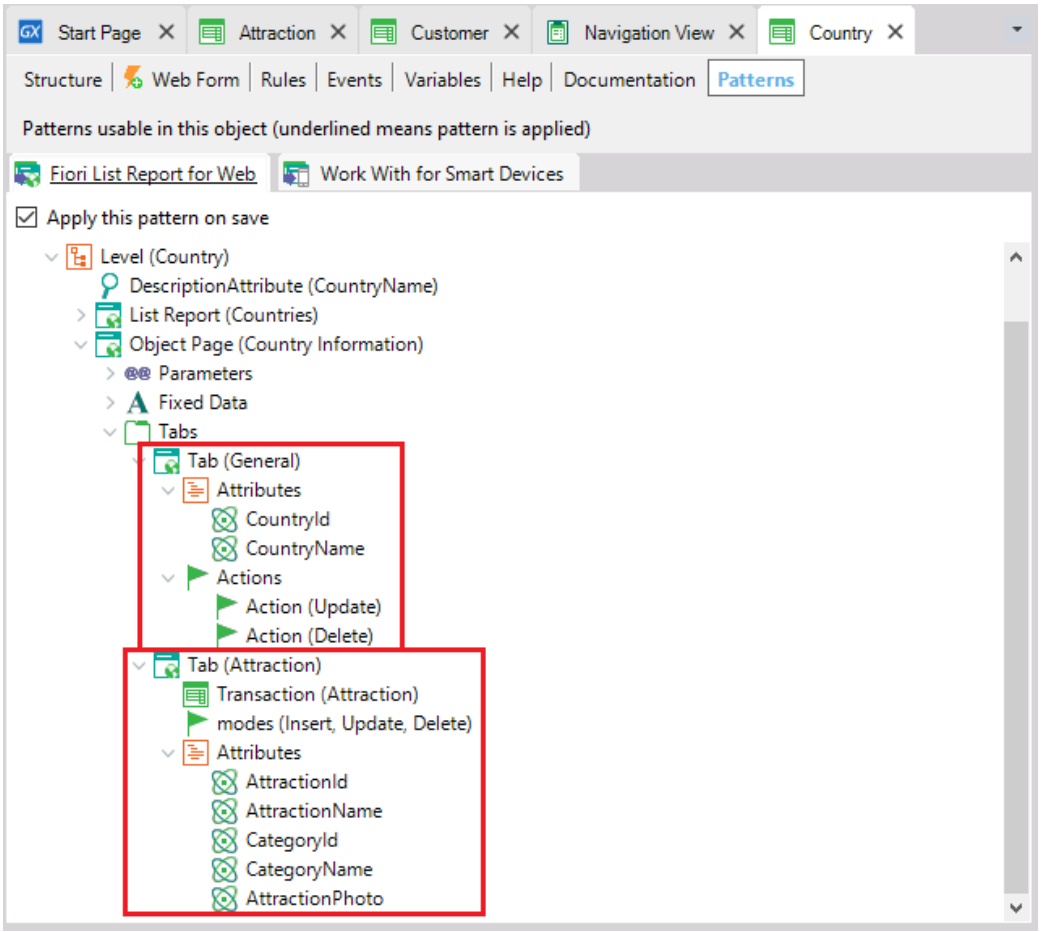
En este caso tenemos un único “tab”, el de nombre General, puesto que una atracción no tiene más información relacionada.

Pero observemos lo que pasa para un país:



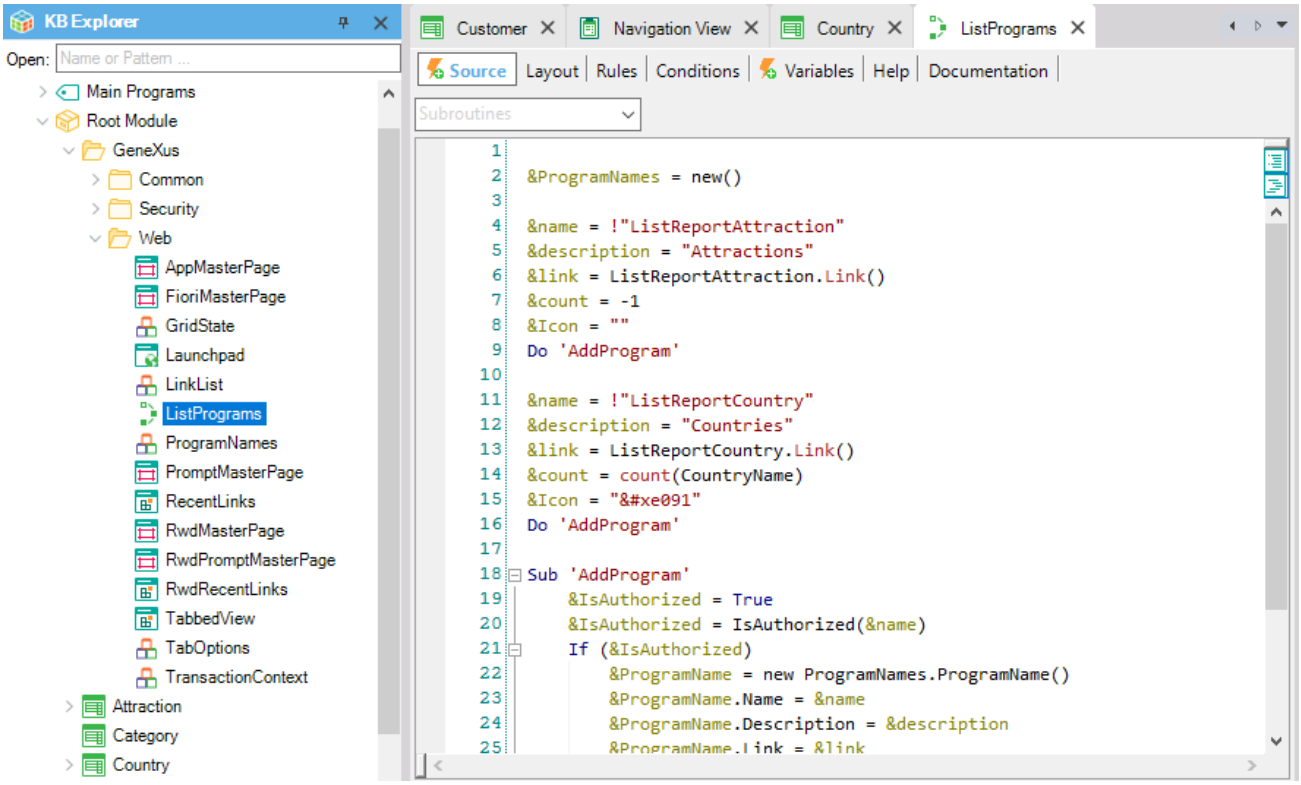
Tiene un “tab” para la información General del país, y otro “tab” para mostrar las atracciones relacionadas.

Si observamos en GeneXus, el ábol incluye ambos tabs:



Con esto hemos visto una muestra de lo que podemos conseguir con muy poco esfuerzo. A continuación veremos el patrón análogo, pero para la app nativa en ambiente móvil.

Pero antes de abandonar la aplicación web, mencionemos algo importante: el Launchpad está incluyendo únicamente tarjetas con los Fiori List Reports definidos en la base de conocimiento. Por ejemplo, en nuestro caso, no le hemos aplicado el patrón a la transacción Customer, por lo que no podemos invocarla desde el Launchpad. Existe un objeto predefinido en la base de conocimiento, el procedimiento de nombre ListPrograms, que es el responsable de obtener la colección de programas a ser incluidos en el Launchpad:

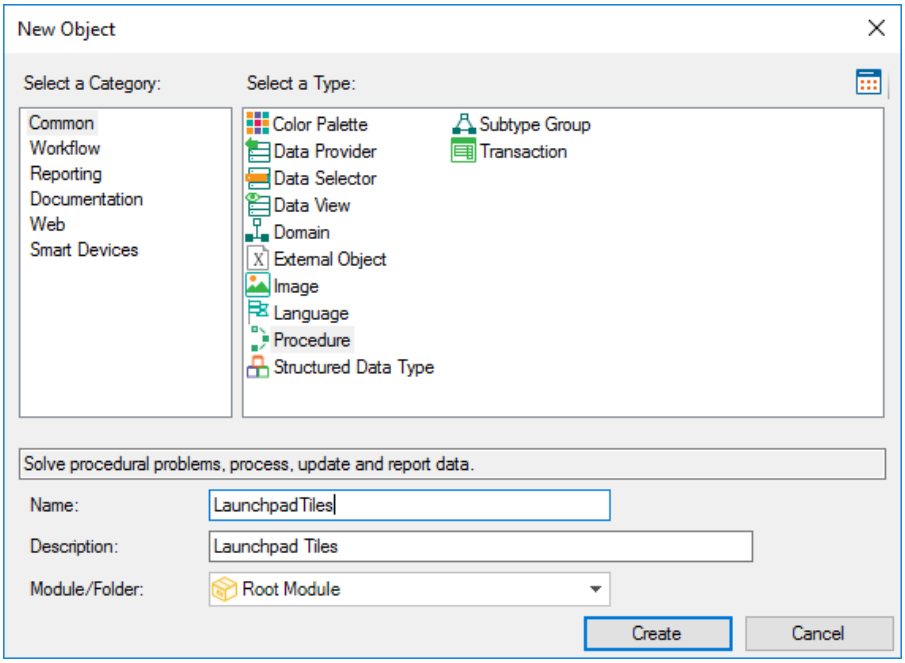


Cuando se aplica el patrón Fiori List Report for Web a una nueva transacción, este procedimiento se modifica, para incluir una tarjeta con el nuevo patrón.

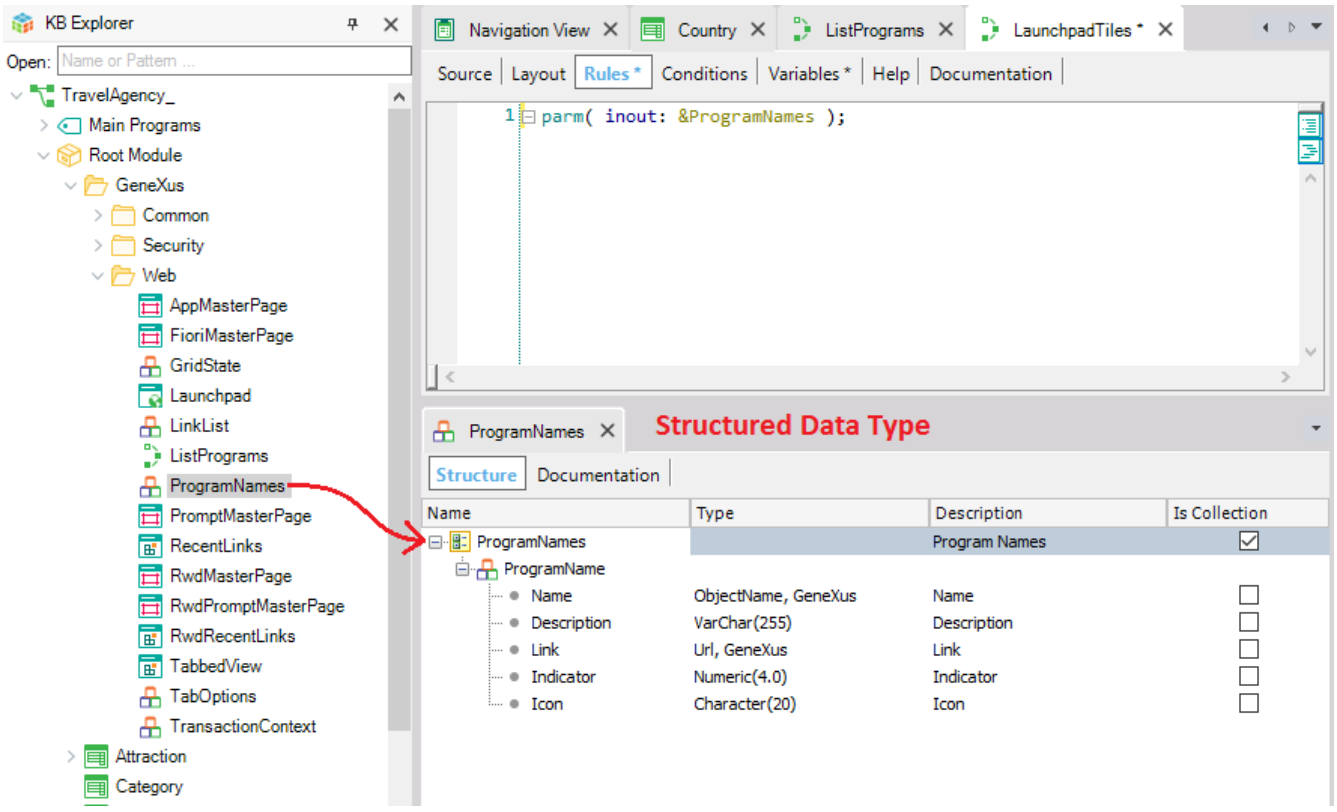
Pero como el launchpad será el objeto principal de la aplicación web, seguramente requerirá tarjetas para invocar a **otros** objetos, además de estos listados (por ejemplo a paneles web creados por el desarrollador). O incluso, se podrá desear que alguno de los Fiori List Reports no aparezca en el launchpad.

El desarrollador lo consigue fácilmente, creándose un objeto de tipo **procedimiento** que reciba por parámetro la lista de programas creada por el ListPrograms, y la manipule de acuerdo a las necesidades del launchpad. Por ejemplo, podrá borrar alguna de los programas (tarjetas) o insertar nuevos.

Agreguemos una tarjeta para invocar a la transacción Customer. Para ello nos creamos un proceidmiento:

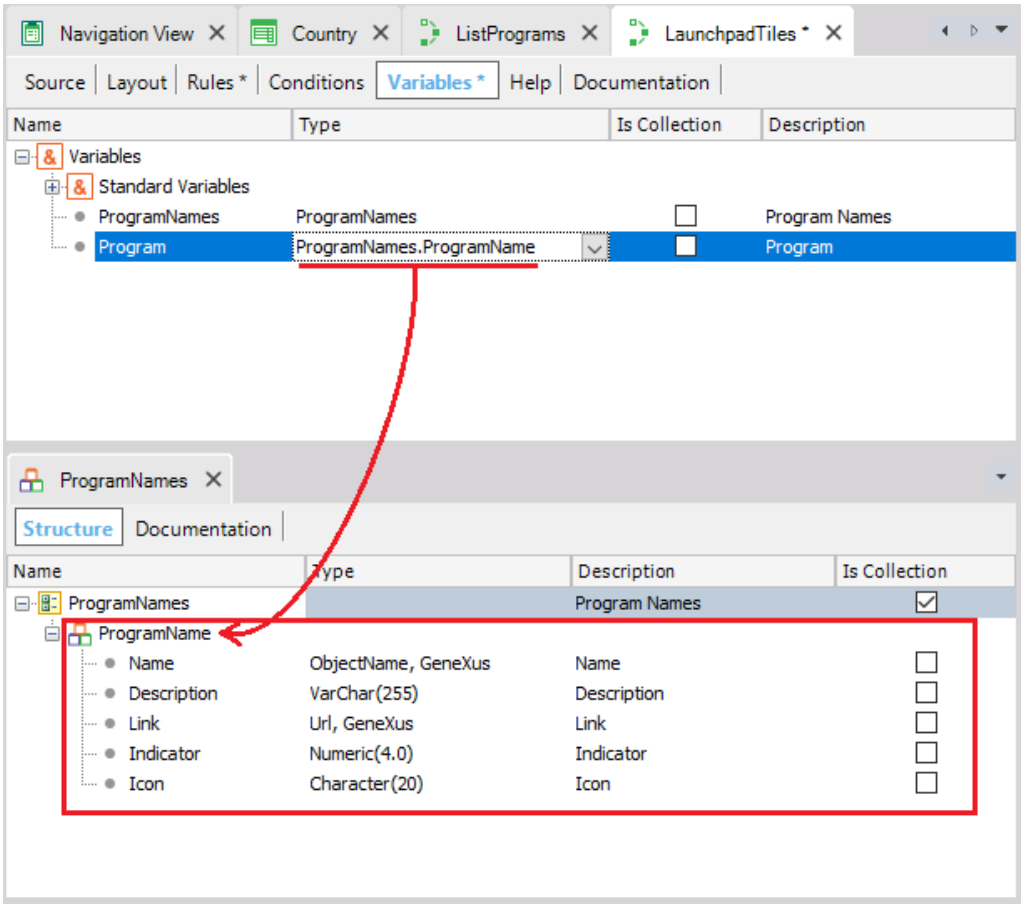


Definimos la regla que permite especificar los parámetros que recibe y/o devuelve el objeto: en nuestro caso, definiremos una variable de entrada/salida, que asumirá el tipo de datos estructurado predefinido ProgramNames (dado que la llamamos igual que el tipo de datos). Este tipo de datos es el que permite registrar la colección de programas a ser invocados:



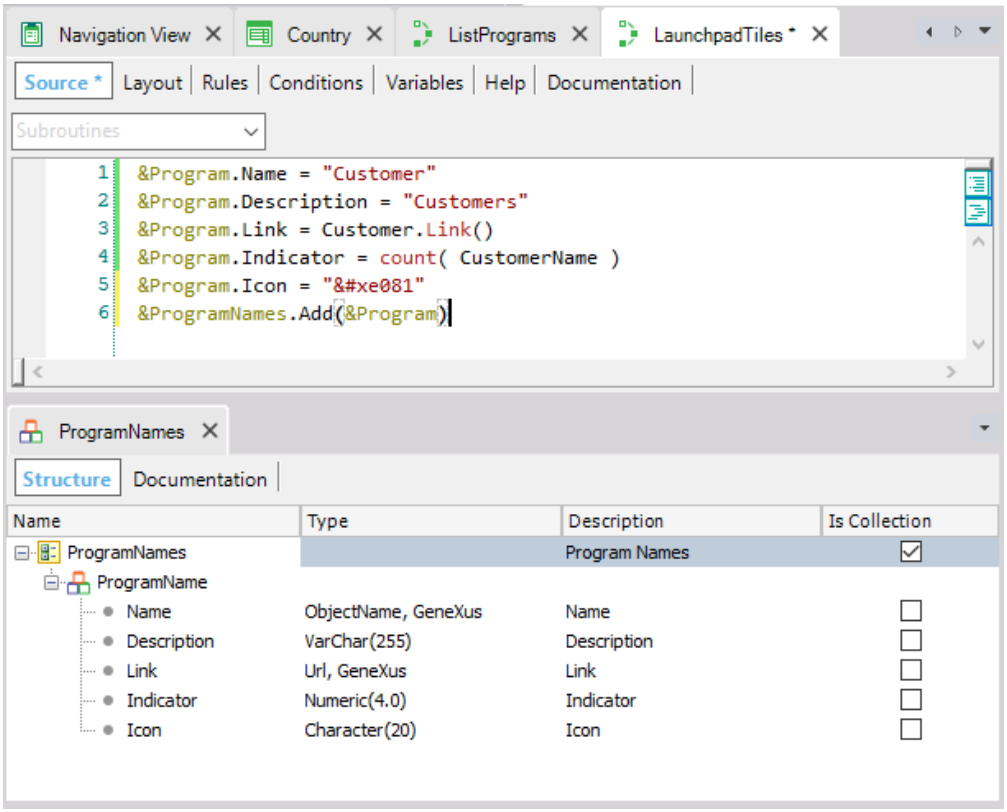
Vemos que es una colección de un tipo de datos estructurado simple, que corresponde a la info de una tarjeta.

Definimos también una variable interna al procedimiento de este tipo de datos:



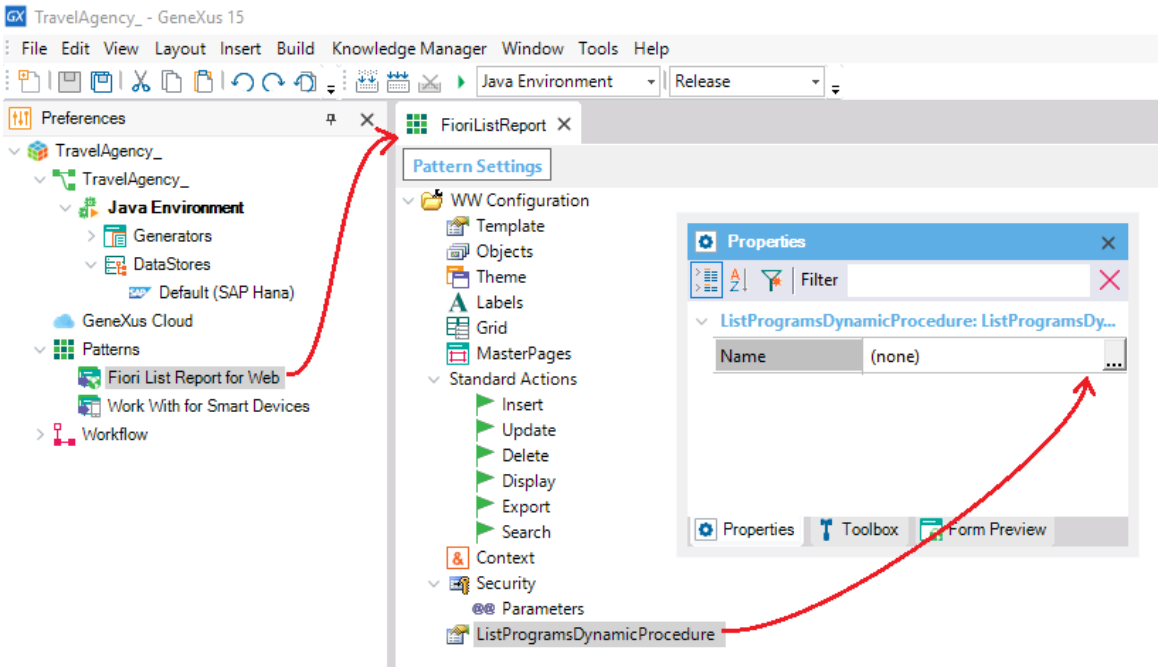
Y en el Source del procedimiento cargamos los elementos necesarios de la tarjeta que queremos agregar… (aquí, para el Indicador de cantidad de customers utilizaremos la misma fórmula Count que vimos antes, pero esta vez observemos que se trata de una fórmula local o inline, es decir, no está asociada a un atributo, y se dispara solamente en este procedimiento, en el momento preciso en que se ejecute esta sentencia)

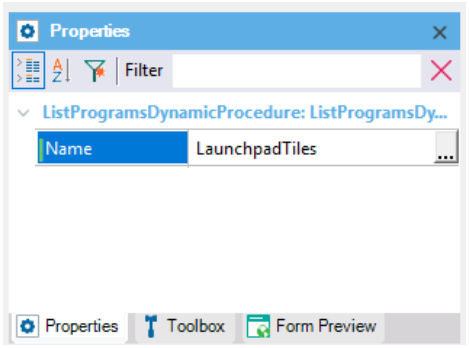
Luego agregamos esa “tarjeta” a la colección de ProgramNames:



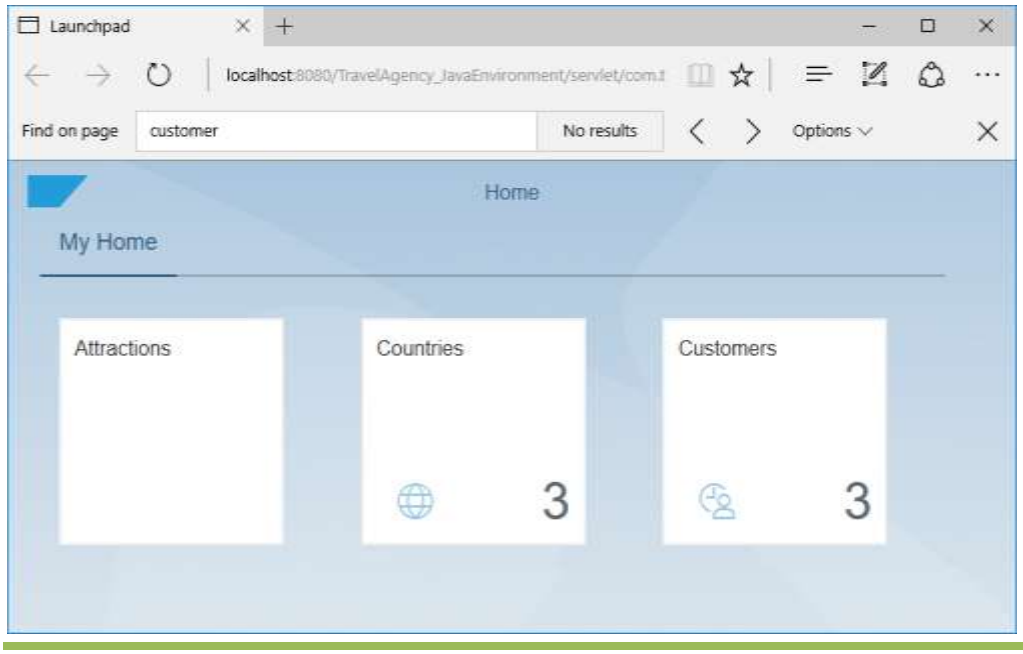
Y por último tenemos que especificar que cuando el Launchapad invoca al procedimiento predefinido ListPrograms que realiza la carga automática de esta colección, antes de terminar invoque a este procedimiento que acabamos de crear.

Eso lo logramos yendo a las Preferences, bajo Patterns, donde podemos ver las propiedades por defecto generales de todo pattern Fiori List aplicado a cualquier transacción (que luego pueden personalizarse en cada aplicación particular del pattern). Aquí es donde se definen los valores “default” que habíamos visto antes. Por ejemplo, que la acción de Insert esté habilitada. Entre las configuraciones posibles vemos el nodo ListProgramsDynamicProcedure. Aquí sustituimos el valor “(none)” por nuestro procedimiento:

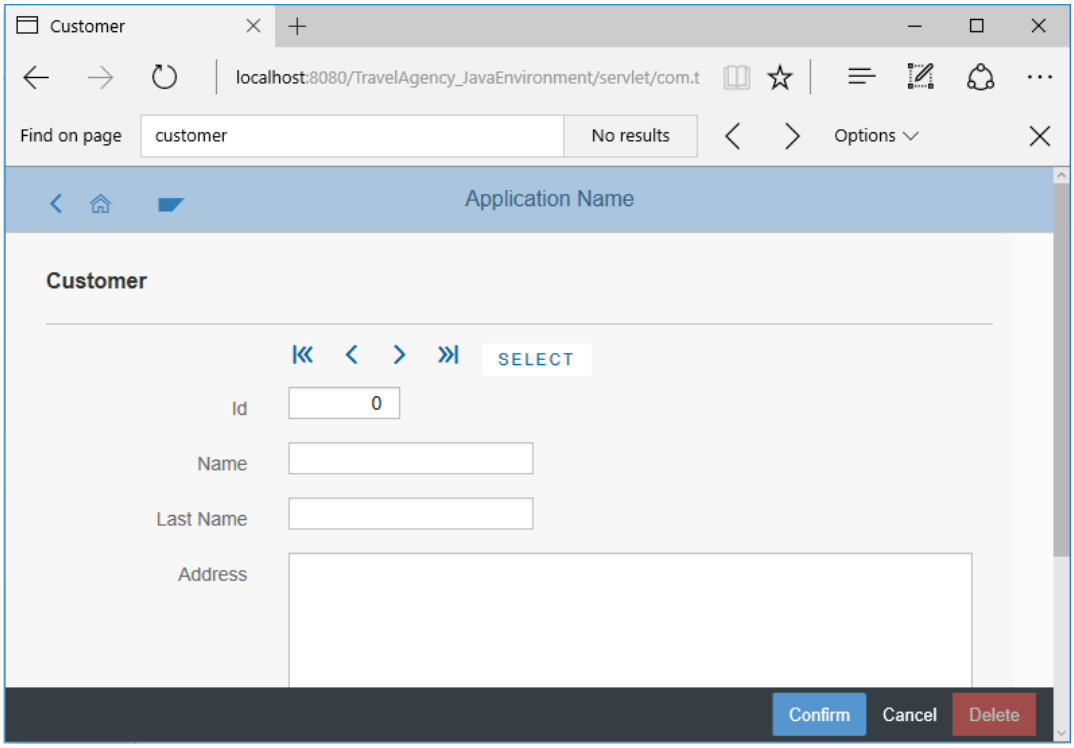

TravelAgency_ - GeneXus 15
File Edit View Layout Insert Build Knowledge Manager Window Tools Help
Java Environment
Release
Preferences
TravelAgency_
TravelAgency_
Java Environment
Generators
DataStores
Default (SAP Hana)
GeneXus Cloud
Patterns
Fiori List Report for Web
Work With for Smart Devices
Workflow
FioriListReport
Pattern Settings
WW Configuration
Template
Objects
Theme
Labels
Grid
MasterPages
Standard Actions
Insert
Update
Delete
Display
Export
Search
Context
Security
Parameters
ListProgramsDynamicProcedure
Properties
Filter
ListProgramsDynamicProcedure: ListProgramsDy...
Name
(none)
Properties
Toolbox
Form Preview


Properties
Filter
ListProgramsDynamicProcedure: ListProgramsDy...
Name
LaunchpadTiles
Properties
Toolbox
Form Preview

F5.


Launchpad
localhost:8080/TravelAgency_JavaEnvironment/servlet/com.t
Find on page
customer
No results
Options
Home
My Home
Attractions
Countries
3
Customers
3

Y vemos que la tarjeta Customers está invocando, efectivamente, a la transacción Customer:



En el próximo video veremos las consecuencias de aplicar el patrón para Smart Devices.