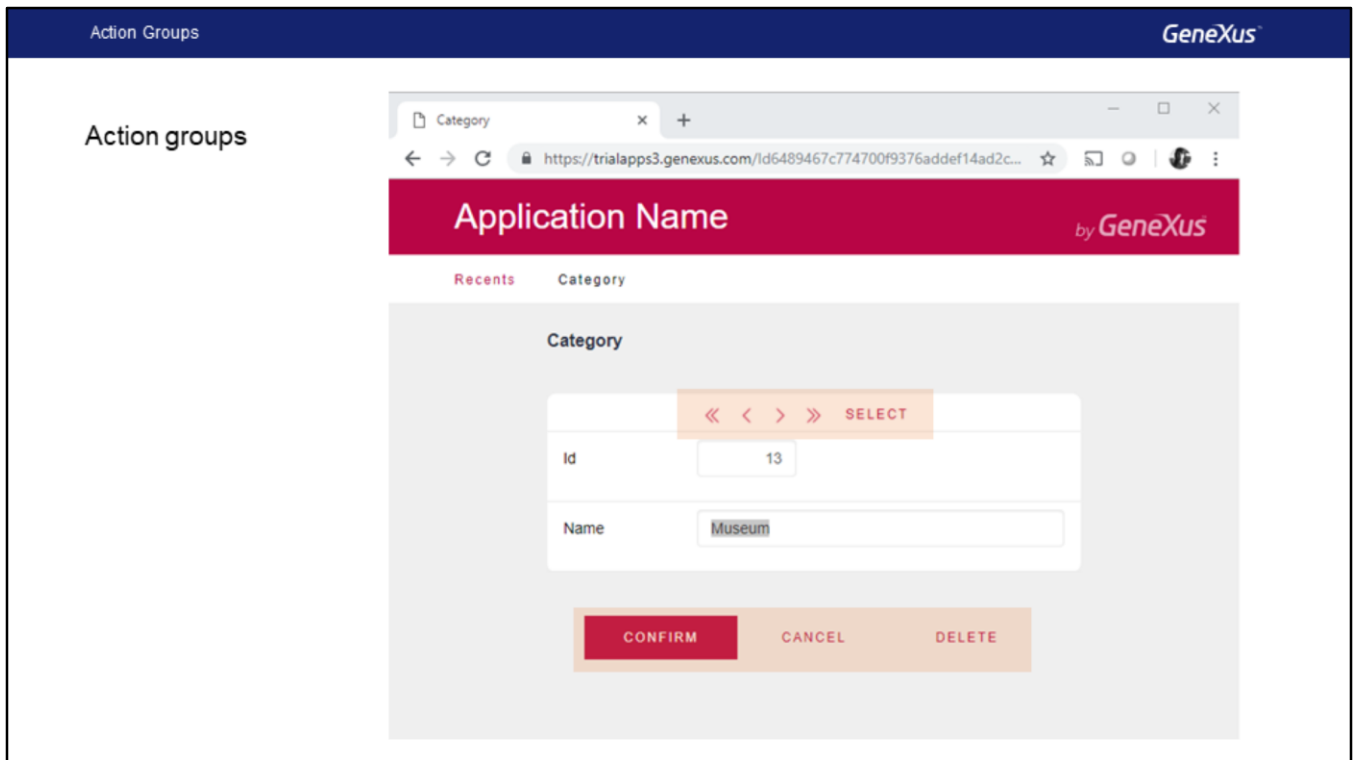


Menús y Master Pages

Pantallas interactivas

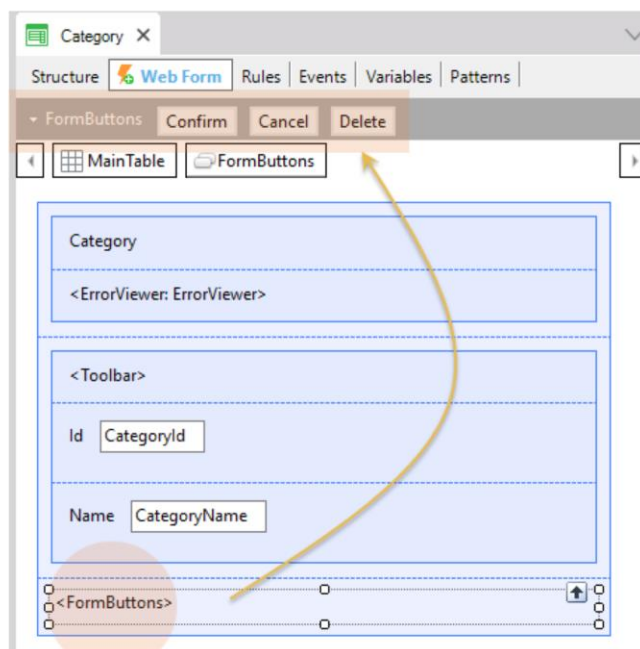
GeneXus 16



Si abrimos una transacción cualquiera en ejecución, por ejemplo Category, vemos en la pantalla los botones de navegación antes de los atributos, y abajo los botones para confirmar, cancelar o eliminar la categoría con la que estamos trabajando en cada oportunidad. ¿Dónde se configura esto en el objeto transacción?

Action groups

- Action group: elemento de la UI utilizado para agrupar controles (botones, imágenes, text blocks y variables)
- Ayuda a crear fácilmente una toolbar
- Permite insertar varios controles dentro de una misma celda de una Responsive Table. Será desplegada en una única fila

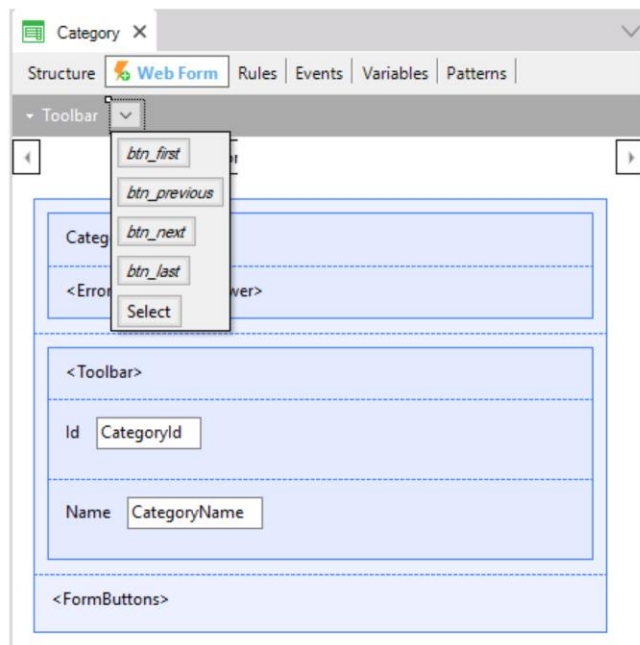


Si abrimos el Web form, vemos que además de los atributos que conforman la estructura de la transacción, GeneXus incorpora dos controles conocidos como Action Groups.

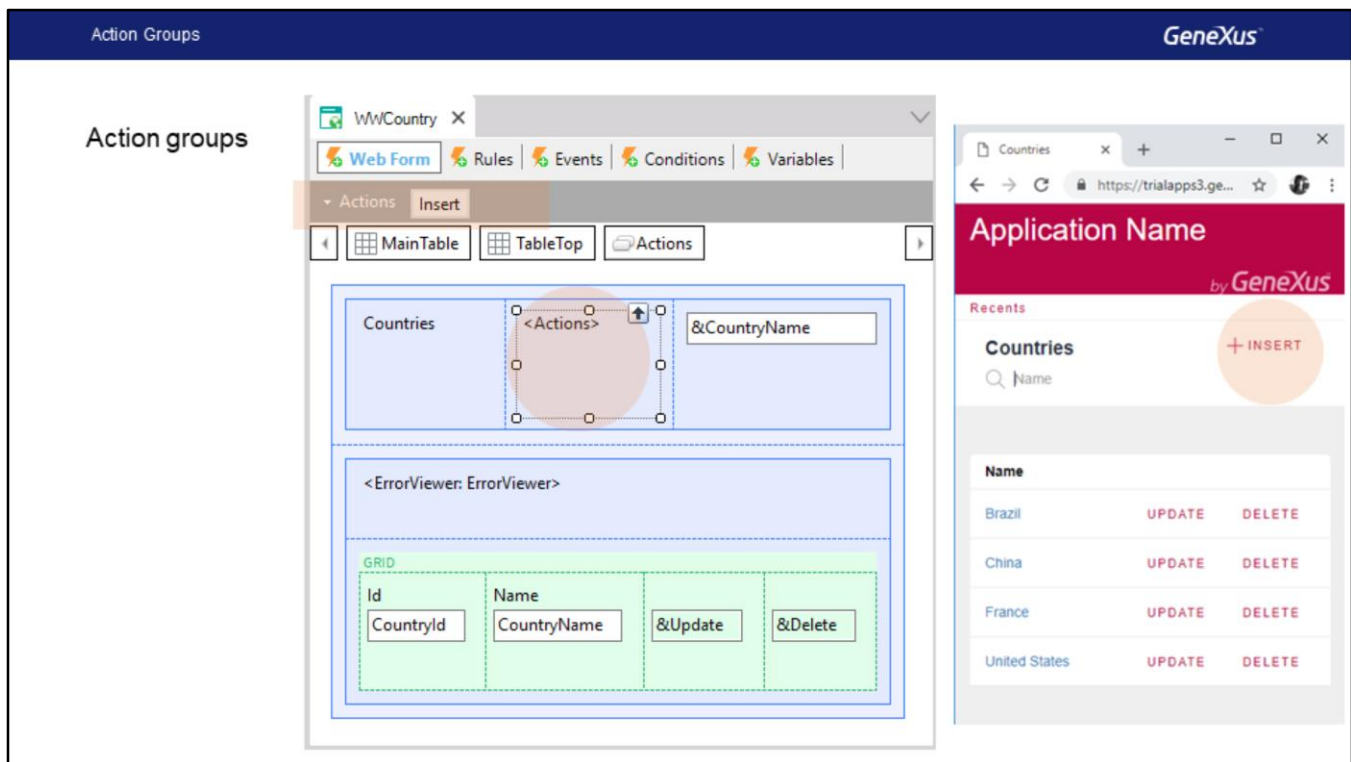
En la imagen estamos editando el control `<FormButtons>`, y vemos que al hacerlo aparecerá arriba del Web form, en la sección conocida como Action Bar, de qué controles estará compuesto ese grupo de acciones. Vemos que corresponderá a los botones para confirmar, cancelar o eliminar la categoría.

Action groups

- Action group puede contener entre otros controles otro control Action group.



Si ahora nos posicionamos sobre el action group control <Toolbar> en el form, vemos que se edita arriba, en la Action Bar, el grupo de acciones de nombre Toolbar. Y vemos que tiene un sub-action group, es decir, un action group dentro del otro action group. Ese action group contiene los botones para navegar entre los distintos registros de la transacción y un botón de Select, que como sabemos, llamará a la lista de selección de categorías, creada automáticamente por GeneXus.



También en el web panel Work With implementado por GeneXus al aplicar el pattern aparecerá un action group, el que en ejecución corresponde al botón de Insert.

Action groups


rs

Attractions

Q Name

+ INSERT ATTRICTIONS.PDF

Attractions PDF

Id	Name	Country Name	Category Na...	Photo	City Name	Trips	
25	Christ the Redemmer	Brazil	Monument		Rio de Janeiro	3	UPDATE DELETE
24	Eiffel Tower	France					
28	Forbidden city	China					

WvAttraction X

Web Form Rules Events Conditions Variables

Actions Insert Attractions PDF

MainTable TableTop Actions

Hide Filters Attractions

Ordered By

Name Country

<ErrorViewer: ErrorViewer>

ErrorViewer

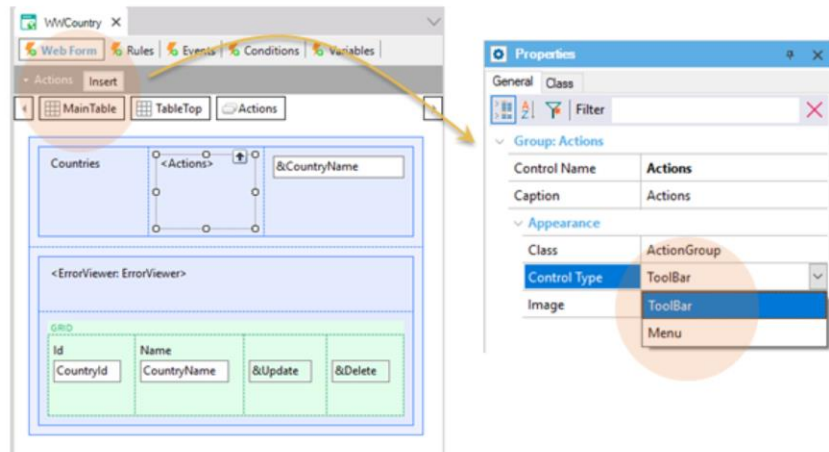
GRID

Id	Name	Country Name	Category Name	Photo	City Name	Trips	
&AttractionId	&AttractionName	&CountryName	&CategoryName		&CityName	&Trips	&Update &Delete

Cuando en la instancia del pattern Work With aplicado a la transacción Attraction habíamos agregado una acción para imprimir en pdf las atracciones, habíamos visto cómo GeneXus incorporaba en el objeto web panel generado esa acción al action group correspondiente.

Action groups

- Podemos crear nuestros propios action groups en web panels y transacciones y agregarlos en el form donde deseemos.
- Dos tipos de action groups:
 - ToolBar
 - Menu

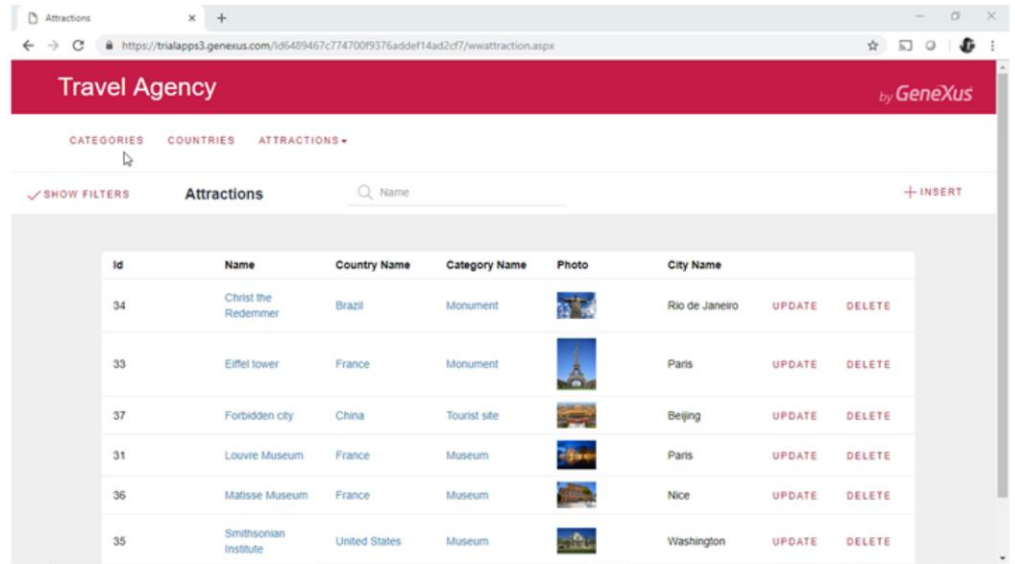


Los ejemplos de action groups vistos antes son del tipo ToolBar. Lo vemos al posicionarnos en la Action Bar, en sus propiedades.

A continuación implementaremos un menú de acciones.

Menú de navegación para el backend

- Backend vs Frontend
- Queremos un menú de navegación con las distintas opciones (aquí reducidas) para el backend.



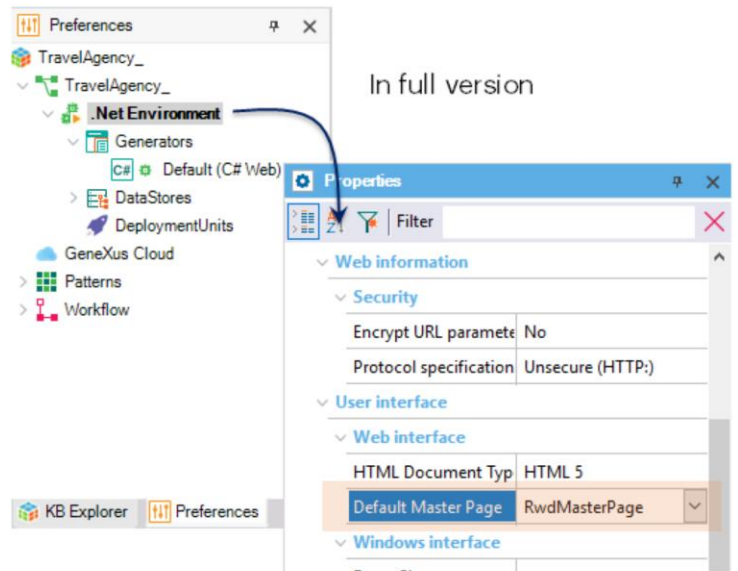
[DEMO: https://youtu.be/yiM-ss0_NSY]

Vamos a querer agregar un menú que aparezca horizontal cuando el tamaño de pantalla lo permite, y como el típico menú de hamburguesa (desplegable) cuando no (por ejemplo cuando la app web se ejecuta en phones). En la primera demo vemos a qué queremos llegar. Es una versión simplificada, donde no hemos colocado todos los ítems que nuestra app necesita (por ejemplo, no hemos agregado los panels que desarrollamos clases atrás).

Las apps en general tienen lo que se conoce como **frontend**, es decir, la aplicación que utilizarán los clientes, la app en sí misma, y un **backend**, que es una app web también, pero que será utilizada por quienes están encargados de mantener los datos de la aplicación, es decir, es una aplicación interna de la empresa. Nosotros hemos venido trabajando hasta el momento de manera desintegrada. No hemos prestado demasiada atención a estos aspectos. Empezaremos a integrarlos un poco en lo que sigue. La mayoría de lo que hemos hecho será para el backend. Por ejemplo, los trabajar con, serán utilizados por el personal de Travel Agency para cargar datos, y no por los clientes que quieren buscar excursiones, países para visitar, etc. Incluso los web panels y reportes que hemos implementado parecen más apropiados para el backend. No hay ninguna diferencia funcional entre backend y frontend, salvo por un tema de seguridad y permisos. Nosotros somos quienes determinamos qué se implementa en el backend y qué va para el frontend.

Menú de navegación para el backend

- Aplicamos el Work With for Web a todas las transacciones, para nuestro backend.
- Invocaremos a cada WW desde un ítem del menú
- Queremos que el menú se vea en todas las páginas web de nuestra app
- ¿Dónde lo insertamos? → Objeto MASTER PAGE
- ¿Cómo sabe cada objeto con qué Master Page ejecutarse?



Lo que haremos, entonces, será agregar un menú que se vea en todas las páginas del backend. Nosotros venimos probando con el Developer Menu, pero evidentemente esa no puede ser una página de entrada (home) de la aplicación. Empezaremos por aplicar el pattern Work With a las transacciones que no lo tenían aplicado.

El menú, como dijimos, será un tipo especial de action group. ¿En qué objeto web lo insertamos para que aparezca como menú en todas las páginas que visitemos dentro del backend?

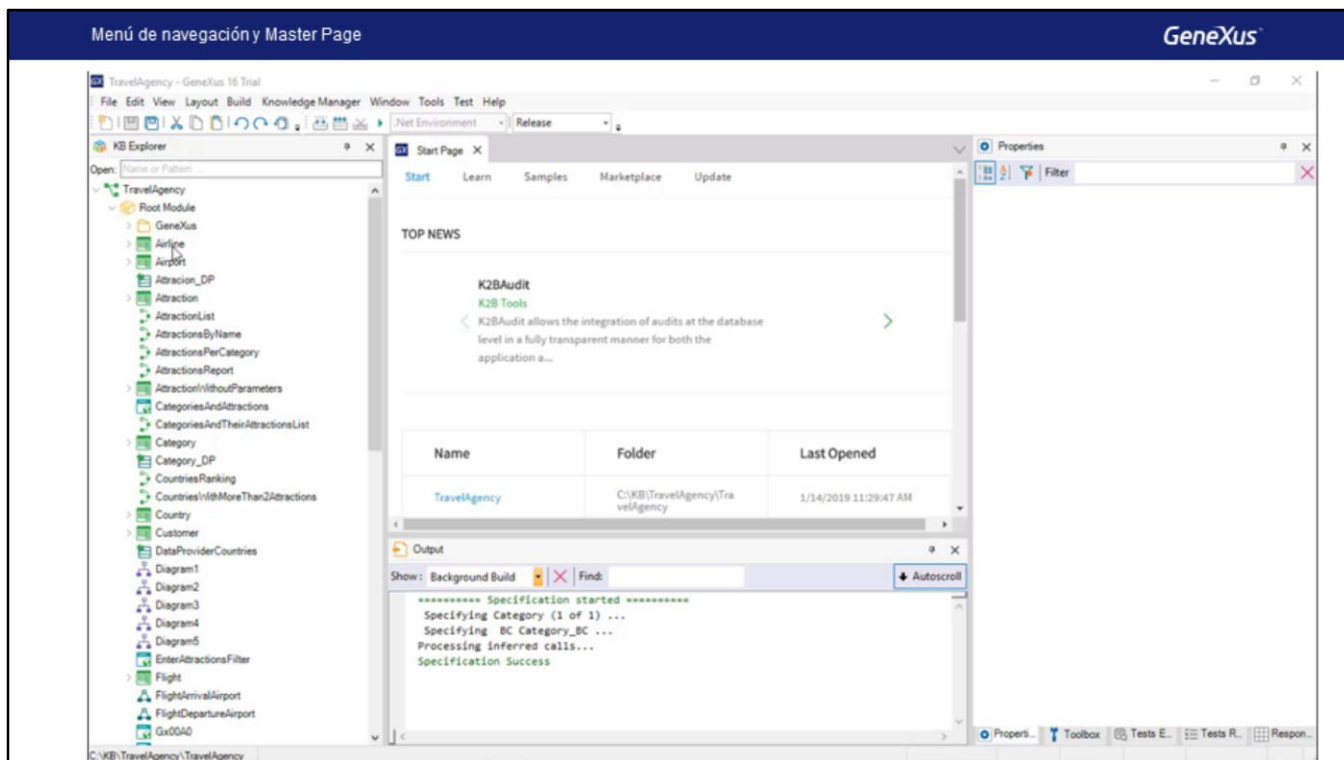
Ya habíamos hablado de las páginas maestras. Habíamos dicho que toda KB se creaba con estas páginas, que son el marco dentro del cual se cargan todas las páginas que se ejecutan.

Por default, cada Environment (más adelante hablaremos sobre los environments, por el momento pensemos que es la instancia de la app que estamos desarrollando) tiene definida una Master Page asociada a todos sus objetos web (incluyendo los que creamos de cero).

En la demo que sigue veremos, por ejemplo que la transacción Attraction es de tipo Web Page y vemos que tiene como Master Page la de nombre RwdMasterPage. Vienen otras Master Pages predefinidas con la KB, para otras cosas que aquí no veremos.

Así, si observamos el objeto WWAttraction creado por el pattern, también vemos que tiene definida esa Master Page como su página maestra. Lo mismo vemos si visualizamos las propiedades de un web panel que creamos nosotros. ¿Dónde se encuentra esa Master Page? Dentro de la carpeta GeneXus, subcarpeta Web

Si estuviéramos utilizando la versión full de GeneXus en lugar de la Trial, podríamos acceder a la pestaña Preferences, donde podríamos cambiar esa Master Page default de todas nuestras páginas web, por otra. De todos modos, siempre puede, para un objeto particular, cambiársele la Master Page por otra cualquiera definida en la KB. En la versión Trial no podemos modificar la Master Page default. Será siempre la que se ve en la imagen de arriba.



[DEMO: <https://youtu.be/eCuD6QQsnyU>]

En la demo estamos viendo cómo absolutamente todas las páginas, tanto las del work with, como las de las transacciones e incluso la de los web panels que nosotros mismos creamos se están ejecutando dentro de ese marco, donde, como podemos ver, toda la barra de arriba se mantiene invariable. Querremos que el menú esté en esa zona.

Abrimos la master page RwdMasterPage, y modificamos, primeramente el nombre de la app, para que ahora diga Travel Agency. Podríamos colocar una imagen allí, a modo de ícono de la app, y todo lo que deseemos.

También vemos que aparece un control de tipo Component, ya mencionado en el curso. Ese componente se crea invocando a otro objeto web panel que es el encargado de mostrar los objetos visitados. Es lo que en ejecución veíamos como Recents.

Y luego, lo más importante, es el control <ContentPlaceholder>. Este control solamente puede estar dentro de un **web panel de tipo Master Page**. Allí es donde se cargará cada página invocada. Lo demás es el marco que esa página tendrá.

Eliminamos el <component> porque no queremos la ruta de los links más visitados, y en su lugar es donde colocaremos nuestro menú.

Para ello vamos a la Toolbox y arrastramos el control New Action Group a su lugar. Se nos abre una ventanita para darle nombre al action group que estamos creando. Le llamamos NavMenu. Podría

ser un menú dinámico, pero aquí no lo veremos.

¿Dónde definimos los ítems del menú? Es arriba, en la Action Bar que se muestra en gris sobre el web form. Hacemos botón derecho allí e insertamos lo que queramos, por ejemplo un Text Block. Modificamos sus propiedades cambiándole el nombre al control, pues luego tendremos que nombrarlo en los eventos, y el Caption, porque será lo que queremos que diga en ejecución.

Hacemos lo mismo por cada opción que queramos incluir.

Luego vamos a los eventos, y como primera cosa comentamos o eliminamos dentro del evento Refresh la carga del componente que habíamos eliminado. Si no, nos mostrará un error al querer grabar, pues ese componente ya no existe más en el form.

Y luego en el evento Start lo que hacemos es definir los links que queremos para cada una de nuestras opciones del menú. Esto se hace a través de la propiedad Link del control TextBlock, y luego invocando al objeto deseado con el método link. Aquí no entraremos en las diferencias entre invocar con o sin Link.

Al presionar F5 y abrir cualquier objeto web vemos que aparece el menú. Aquí solo incluimos como ítems las atracciones, categorías y países, por simplicidad. Pero esto no es un menú de navegación, es una ToolBar. No está apareciendo como queríamos. No muestra el menú de hamburguesa cuando el tamaño de la pantalla es pequeño. Es que no le cambiamos su Control Type por Menu.

Al hacerlo y presionar F5, lo vemos como queríamos. De hecho, si utilizando chrome presionamos F12, podemos hacer variar la pantalla para ver cómo se vería no solo en desktops, sino también en dispositivos móviles. Es una buena manera de probar la responsividad. Y así vemos cómo en un iphone, por ejemplo, sí está apareciendo el menú de hamburguesa.

Vimos

- Propiedad **Master Page** de cada objeto web
- Objeto Master Page
 - Control ContentPlaceholder en su web form
 - Cómo insertar un action group
 - Cómo hacer que sea de tipo menú

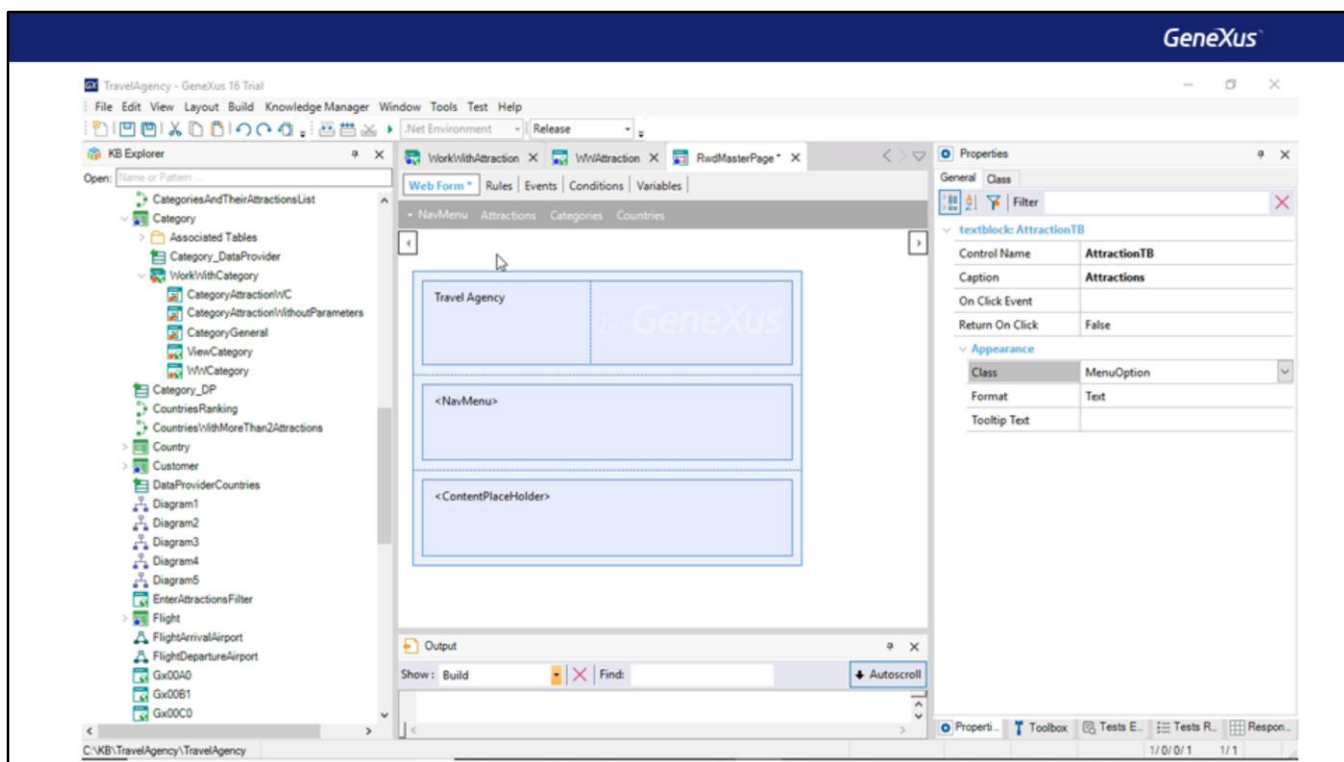


Menú con sub action group

- Es posible anidar action groups dentro de otros
- Para cambiar el diseño o algo del comportamiento de los controles, los asociamos a clases, que quedan definidas dentro de un objeto Theme de la KB (aquí solo lo introducimos; se verá específicamente más adelante)



Vea la demo en la siguiente página.



[DEMO: <https://youtu.be/zPFotMPpw6Q>]

Para agregar un action group a otro, simplemente hacemos botón derecho / Insert Group. Crearemos un submenú para mostrar cosas de atracciones turísticas: tanto el Work With como un par de los reportes que teníamos implementados en nuestra KB. También podrían ser web panels.

Allí dentro arrastramos el text block que teníamos en el padre, Attractions, y agregamos dos nuevos text blocks, para invocar a cada reporte. Le tenemos que colocar un Caption apropiado al Submenú, porque será lo que se verá antes de desplegarlo. Le ponemos como Caption, Attractions.

‘Vemos en ejecución que cuando el menú se despliega horizontal –no como hamburguesa- el submenú se está solapando con el contenido de abajo. Es decir, su color de fondo (background color) es transparente. Si quisiéramos que fuera blanco, para que esto no suceda, ¿dónde lo modificamos?

Si en GeneXus editamos las propiedades del action group que implementa el menú (lo mismo veríamos si editamos las del submenú), encontramos que bajo el grupo Appearance aparece una propiedad class, por defecto la de nombre ActionGroup. ¿Qué es una clase?

Ya dijimos cuando vimos más sobre web panels que la relación entre un control del form y su clase es análoga a la relación entre un tipo de datos de un control atributo o variable y su dominio. Es decir, la clase centraliza un montón de propiedades, como para poder cambiarlas en un lugar solo y apliquen automáticamente a todos los controles que la tengan por clase.

Vemos que GeneXus nos ofrece un camino corto para poder editar las propiedades de la clase del control que estamos editando. Es el tab Class, al lado del General. Allí cambiamos la propiedad

Background Color por White.

Pero al hacerlo vemos que se nos abre automáticamente un objeto de nombre Carmine. Es un objeto de tipo Theme, y es el theme predefinido de la KB. Es que al cambiar a White el valor anterior, no lo hicimos solamente para el control action group que estábamos editando, sino para cualquier otro action group que tenga a la de nombre ActionGroup por clase. El objeto theme es el que almacena las propiedades de todas las clases posibles de la KB. Vemos que también está la clase ActionGroupItem que es la que permite definir el diseño y comportamiento de los ítems del menú. Así, si observamos, vemos que el color rojo que veíamos en ejecución en las opciones del menú proviene de la propiedad Forecolor de esta clase.



Videos

training.genexus.com

Documentation

wiki.genexus.com

Certifications

training.genexus.com/certifications