

Grillas horizontales, Grillas flexibles y Control Flex

GeneXus™

Cuando vimos la aplicación de la agencia de viajes terminada al principio de este curso, podíamos ver las atracciones desplazándose en sentido horizontal como si fuera un carrusel.

En este video veremos cómo configurar un grid para que adopte ese comportamiento, así como también otros tipos de grids y controles flexibles que tenemos disponibles para desarrollar nuestra aplicación.

Smart grid

Comencemos con la grilla estándar, a la que podemos configurar un desplazamiento horizontal.

Configurando el desplazamiento horizontal en un grid

Layout | Rules | Events | Conditions | Variables | Documentation

Application Bar


MainTable | Grid1

Country &CountryId

Name From &AttractionNameFrom

Name To &AttractionNameTo

GRID

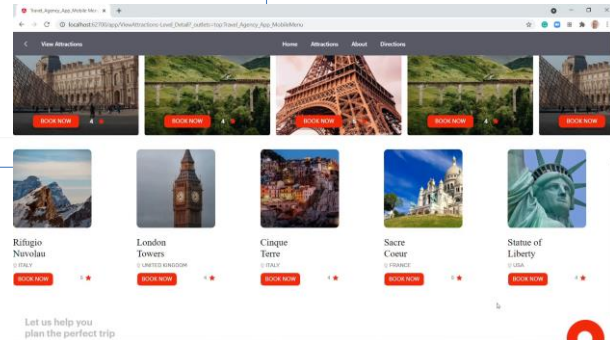
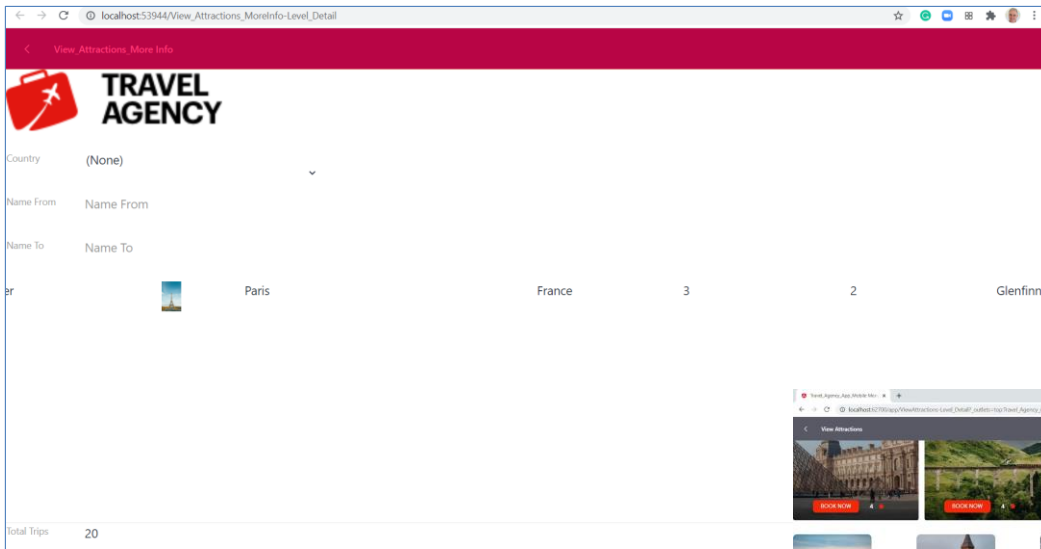
AttractionId	AttractionName		CityName	CountryId	CountryName	&Trips
--------------	----------------	---	----------	-----------	-------------	--------

Total Trips &TotalTrips

Control Info	
Control Type	Grid
Auto Grow	False
Scroll Direction	Horizontal
Snap To Grid	False
Items Layout Mode	Single
Appearance	
Class	Grid
Visible	True
Invisible Mode	Keep Space
Enabled	True
Rows	<default>
Empty Grid Background Image	(none)
Empty Grid Background Class	Image
Empty Grid Text	
Empty Grid Text Class	TextBlock

Desde el upgrade 11 de la versión GeneXus 16, el control grid pasó a ser un Smart Grid, por lo que disponemos de propiedades que nos permiten cambiar la apariencia de cómo se va a mostrar la información, de una forma más flexible.

Si lo único que queremos es que el desplazamiento de los registros sea en dirección horizontal, podemos setear la propiedad Scroll Direction del grid en el valor Horizontal y si ejecutamos...



Vemos que ahora el grid ocupa una sola fila y pinchando con el mouse podemos desplazarnos hacia la derecha y luego podemos volver hacia la izquierda.

Si volvemos a la aplicación terminada, a la sección de la pantalla donde se recorren todas las atracciones disponibles, vemos que además del desplazamiento horizontal, se pueden ver varias atracciones a la misma vez, por lo que debemos ver cómo hacer para configurar esto.

Pero además, para cada atracción los datos de la atracción están dispuestos en forma diferente a lo que logramos en nuestra ejecución.

The screenshot shows the GeneXus IDE interface. On the left, the 'Layout' tab is active, displaying a design with a grid containing various controls like text boxes and a dropdown menu. A 'Rows Style' dialog box is open in the center, showing a table of row heights and unit settings. On the right, the 'General Class' properties panel is visible, showing settings for the 'Grid1Table' control, including appearance and scroll behavior.

Row	Height	Unit
1	pd	Percentage
2	pd	Device
3	400dip	Independent Pixel
4	pd	Platform Default
5	pd	
6	pd	
7	pd	

Control Name	Grid1Table
Layout Name	
Appearance	
Columns Style	100%
Rows Style	pd;pd;400dip;pd;pd;t ...
Width	100%
Height	pd
Auto Grow	True
Class	Table
Background	(none)
Visible	True
Invisible Mode	Keep Space
Enabled	True
Scroll Behavior	
Scroll Factor	1
Zoom Factor	0
Scroll Attachment	Parent
Layout Behavior	
Expand Bounds	Background Only
Expand Bounds Directions	Top, Left, Bottom, Right

Es decir que si comparamos el diseño de columnas lado a lado de nuestro grid, con la forma en que se ven los datos en la aplicación terminada, no solamente debemos modificar cuántos ítems entran en cada fila o columna, sino que debemos cambiar el diseño mismo del grid.

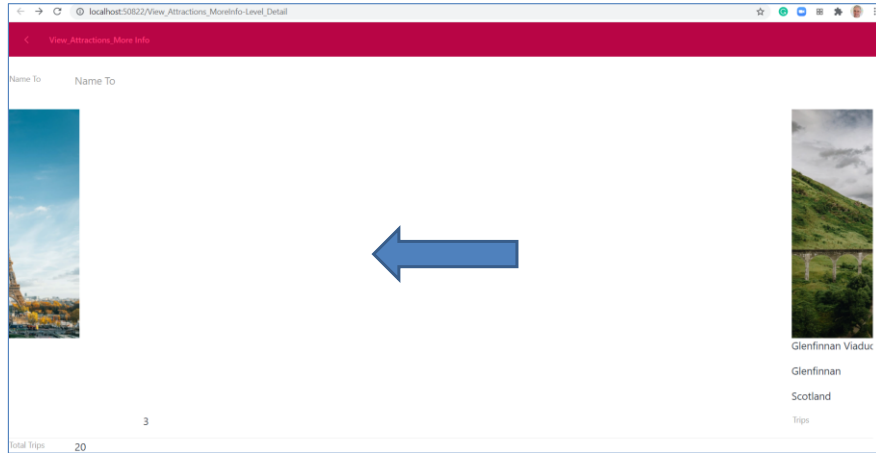
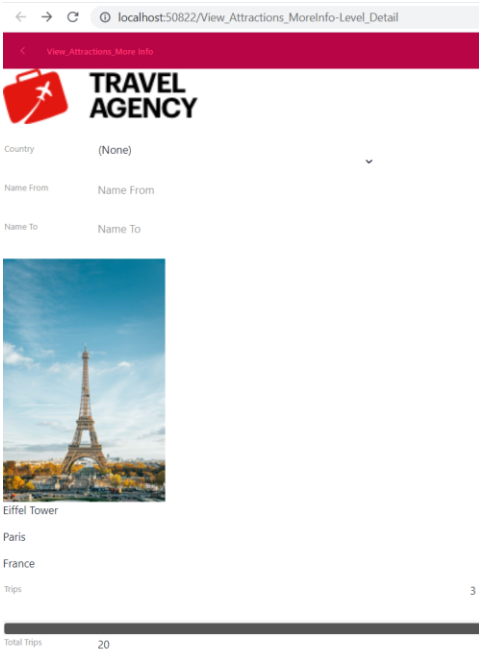
Para eso, vamos al atributo CountryId y lo movemos debajo de AttractionId y lo mismo hacemos con el resto de los atributos de las columnas, lo ponemos todos hacia abajo. Luego seleccionamos a los atributos AttractionId y CountryId, le ponemos visible en False y la propiedad Invisible mode en el valor Collapse space.

Vamos a ajustar las propiedades de la tabla que está dentro del grid. Para eso seleccionemos al atributo AttractionId y luego seleccionemos la tabla Grid1Table. Ajustamos la propiedad Rows Style y le damos 400 dips a la tercer fila que es la de la foto.

Como queremos desplazarnos horizontalmente, seleccionamos el grid y ponemos la propiedad Scdroll Direction en Horizontal.

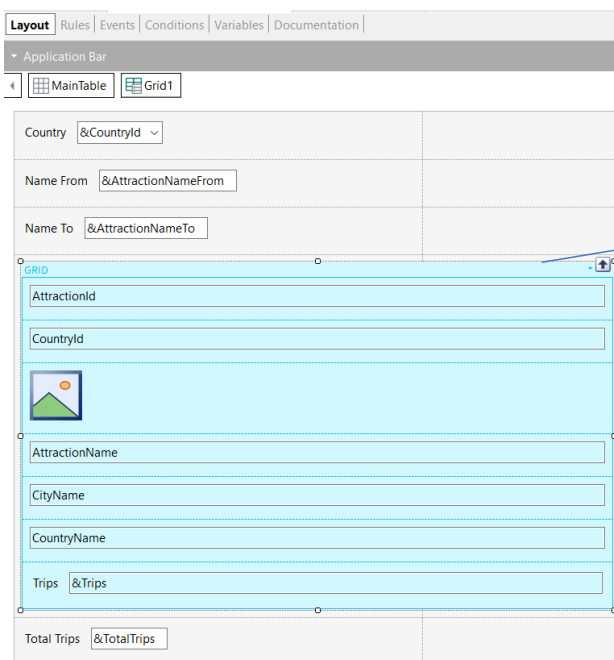
Y ejecutamos.

Nuevo diseño y desplazamiento horizontal



Vemos que ahora la presentación de los datos es más parecida a lo que queríamos, pero aún tenemos el problema que vemos únicamente un registro por vez en la pantalla.

Propiedad: Items Layout Mode



Control Info	
Control Type	Grid
Auto Grow	False
Scroll Direction	Horizontal
Snap To Grid	False
Items Layout Mode	Single
Appearance	
Class	Multiple by Quantity
Visible	Staggered by Quantity
Invisible Mode	Multiple by Size
Enabled	True
Rows	<default>
Empty Grid Background Image	(none)
Empty Grid Background Class	Image
Empty Grid Text	
Empty Grid Text Class	TextBlock

Además de la propiedad Scroll Direction del grid, disponemos de la propiedad Items Layout Mode que nos permite cambiar la forma en que aparecen los datos del grid.

Hay varios valores disponibles:

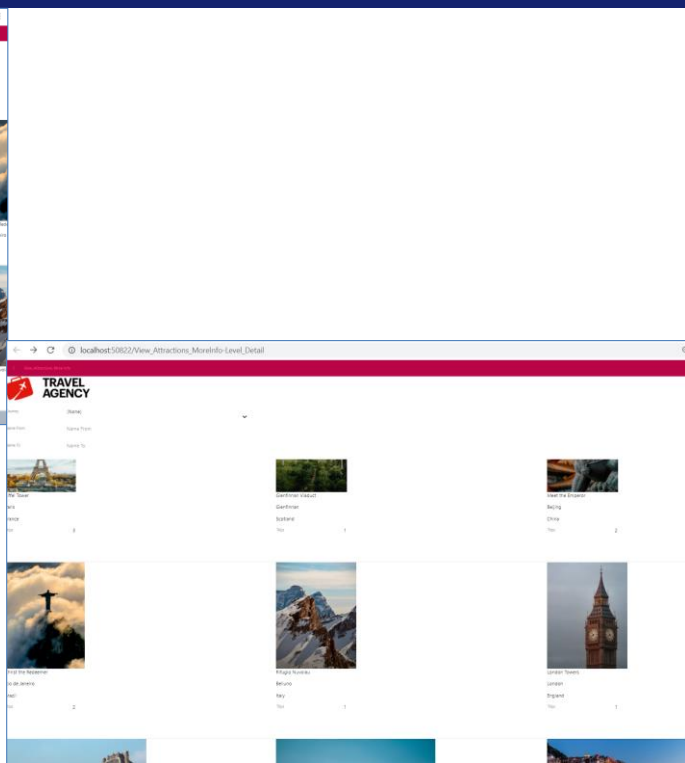
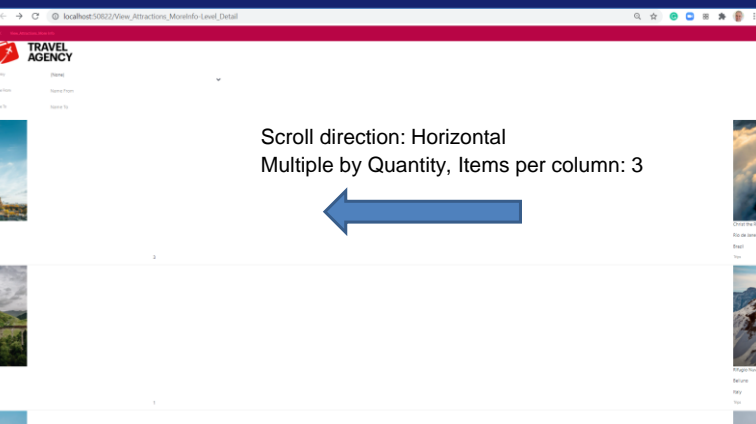
Single – Es el valor por defecto y hace que la grilla muestre solamente un elemento por fila.

Multiple by Quantity – Este valor indica que se mostrará un número fijo de elementos en cada fila / columna del grid. Se determinará mediante las propiedades Items Per Row si la dirección de scroll es vertical o Items Per Column, si la dirección de scroll es horizontal.

Staggered by Quantity – La grilla se muestra con un diseño escalonado, donde cada elemento puede tener un tamaño diferente. La cantidad de elementos por fila o columna se determinan mediante las propiedades Items Per Row o Items Per Column, dependiendo de la dirección de desplazamiento del grid.

Multiple by Size - El número de elementos que se muestran en cada fila / columna se determinará según el tamaño de cada elemento. Para lograrlo, se utilizan los valores de las propiedades Maximum Width, Minimum Width si la dirección de scroll es vertical y Maximum Height y Minimum Height si la dirección de scroll es horizontal.

Vamos a asignar el valor Multiple by Quantity y como tenemos la dirección de scroll en horizontal, nos aparece la propiedad Items Per Column. Le ponemos el valor 3 y ejecutamos.

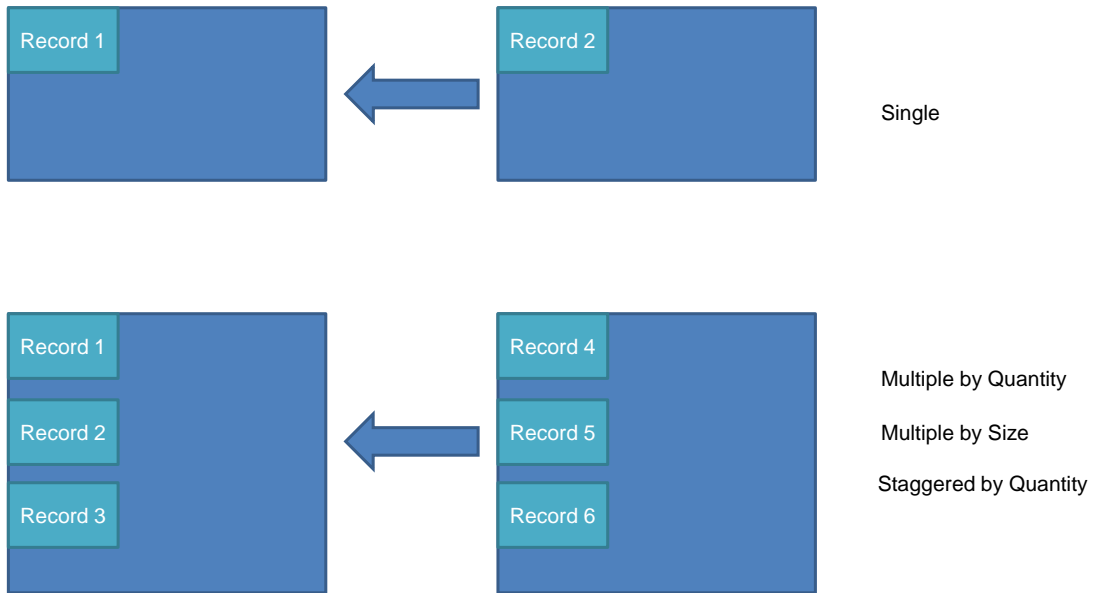


Vemos que vemos una única columna en la página y en cada columna 3 atracciones que las vemos si nos desplazamos hacia abajo. Si paginamos hacia la derecha aparecen las siguientes 3 atracciones, que aparecen hacia abajo.

Esto no es lo que queremos, vamos a cambiar el scroll a vertical y dejamos las propiedades con los mismos valores.

Ahora vemos en cada fila a 3 atracciones y para ver las siguientes nos debemos mover hacia abajo.

Propiedad: Items Layout Mode, Scroll direction = horizontal

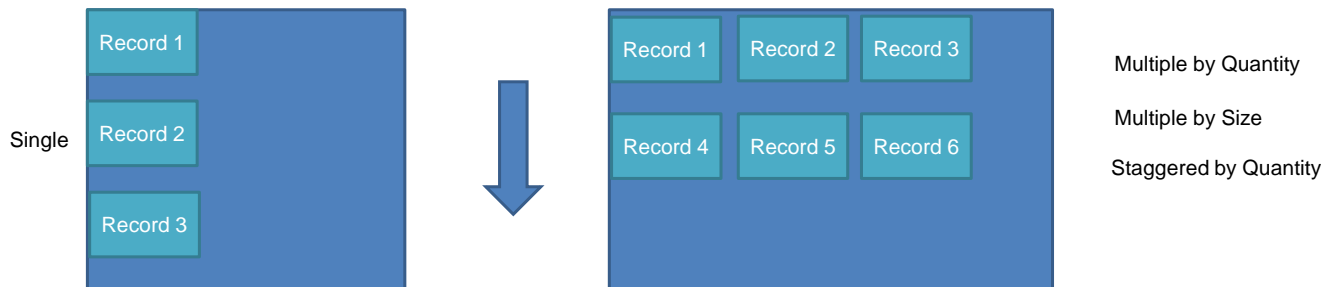


En resumen, en el grid estándar (smart grid), si el desplazamiento es horizontal como en nuestro ejemplo, podemos definir cuántos registros queremos ver por columna, es decir dentro de la misma columna hacia abajo.

Pero siempre estaremos viendo una columna a la vez.

La cantidad de registros por columna que se ven hacia abajo, lo podemos definir como un valor fijo, o que dependa de la altura (mínima y máxima) que le queramos dar a cada item. Si definimos que lo haremos con un valor fijo tenemos la posibilidad que se vea escalonado o no, dependiendo de si elegimos Staggered by Quantity, o Multiple by Quantity, respectivamente.

Propiedad: Items Layout Mode, Scroll direction = vertical



Si el desplazamiento es vertical, podemos configurar cuántos registros queremos ver en cada fila.

Es decir que la fila siguiente volverá a traer el mismo grupo de registros. Y esto también lo podemos definir como un valor fijo (escalonado o no), o que dependa del ancho (mínimo y máximo) que le queramos dar a cada ítem.

O sea que no lograremos con el Smart Grid el comportamiento que necesitamos, de poder ver todas las atracciones que quepan en el ancho disponible de la pantalla y que podamos desplazarnos hacia la derecha si hay todavía más que no podemos ver.

Horizontal Grid

Ya vimos varias opciones del grid estándar para desplegar información. Veremos ahora otro control denominado Horizontal Grid.

Horizontal grid

The screenshot shows the GeneXus IDE interface. On the left, a form is displayed with a horizontal grid control. The grid contains the following fields: AttractionId, CountryId, AttractionName, CityName, CountryName, and Trips. The grid is highlighted with a blue border. On the right, the Properties window is open, showing the 'Control Info' section. The 'Control Type' is set to 'Horizontal Grid'. The 'Columns Per Page Portrait' property is set to 5, and the 'Rows Per Page Portrait' property is set to 1. The 'Appearance' section shows the 'Class' property set to 'Grid'.

Control Info	
Control Type	Horizontal Grid
Auto Grow	False
Paged	True
Show Page Controller	True
Page Controller Class	PageController
Columns Per Page Portrait	5
Rows Per Page Portrait	1
Columns Per Page Landscape	1
Rows Per Page Landscape	1

Appearance	
Class	Grid
Visible	True
Invisible Mode	Keep Space
Enabled	True
Rows	<default>
Empty Grid Background Image	(none)
Empty Grid Background Class	Image
Empty Grid Text	
Empty Grid Text Class	TextBlock

Si vamos a la propiedad Control type, vemos que hay varios controles que podemos usar para que la información se despliegue. Elegimos el valor Horizontal Grid.

Al asignar este valor, se habilitarán una serie de propiedades mediante las cuales podremos configurar cómo se verá el grid. Por ejemplo, la propiedad Show Page Controller determina si se verá o no el paginador y la propiedad Page Controller Class nos permitirá definir la clase en la que está basada este paginador.

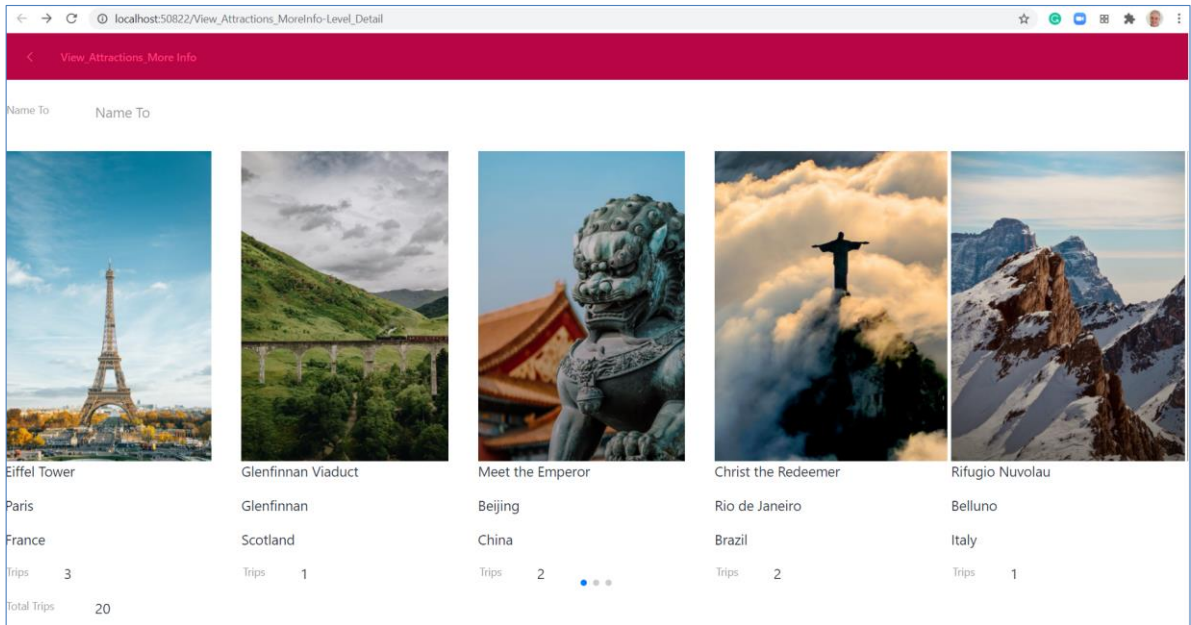
Esta clase es como un template al que le podemos cambiar propiedades y todos los controles que se basen en la clase, heredarán dichos valores.

Sin embargo, GeneXus 17 nos provee una forma mucho más flexible, escalable y potente de hacer esto que es a través de un Design System Object, que estudiaremos más adelante.

Volviendo a las propiedades que nos permiten personalizar nuestra grilla horizontal, vemos que podemos definir cuántas columnas queremos ver y a la vez cuántas filas queremos en cada página. Vamos a poner la propiedad Columns Per Page Portrait en el valor 5 y dejamos la propiedad Rows Per Page Portrait en el valor 1 porque queremos ver una sola fila por página.

Ejecutamos.

Usando un control Horizontal Grid



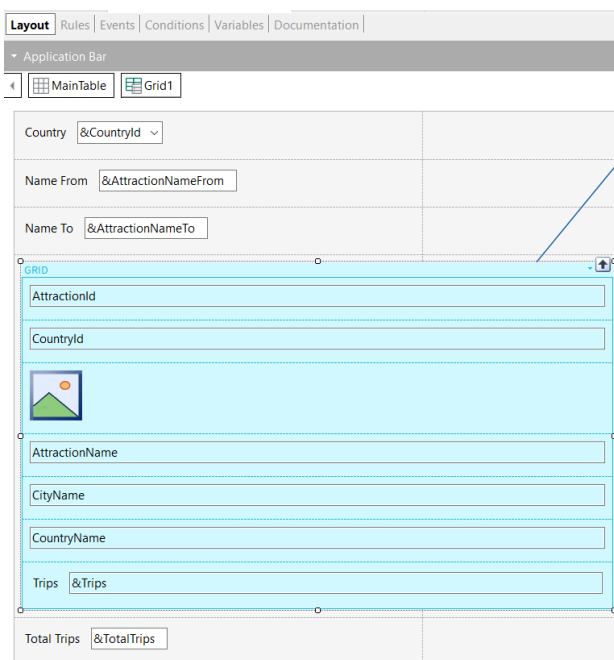
Vemos que nos aproximamos mucho a la forma en que queríamos verlo. Ahora podemos paginar hacia la derecha y veremos las atracciones ocupando toda la página, 5 atracciones a la vez.

Si hubiéramos cambiado también la propiedad `Rows Per Page Portrait`, podríamos ver varias filas de atracciones, pero no nos interesa en nuestro caso.

Observemos debajo de la página al control de paginado, que nos dice en qué página estamos.

Flex Grid y Flex Control

Ya hemos logrado un resultado aceptable para nuestra aplicación.
Veremos ahora otra grilla aún más flexible y un contenedor flexible que
permite organizar contenido aún desconociendo su tamaño.



Control Info

Control Type	Flex Grid
Auto Grow	False
Flex Direction	Row
Flex Wrap	No Wrap
Justify Content	Flex Start
Align Items	Stretch
Align Content	Stretch

Appearance

Class	Grid
Visible	True
Invisible Mode	Keep Space
Enabled	True
Rows	<default>
Empty Grid Background Image	(none)
Empty Grid Background Class	Image
Empty Grid Text	
Empty Grid Text Class	TextBlock

Para este ejemplo primero hacemos un Save As del panel View_Attractions_MoreInfo y le ponemos de nombre View_Attractions_MoreInfo 2.

Para usar la grilla flexible, vamos a la propiedad Control Type del grid y elegimos Flex Grid.
Flex Grid es un tipo de grilla FreeStyle, donde podemos cargar tantos elementos como queramos, de forma dinámica.

La forma en que aparece la información depende de lo que establezcamos en las cinco propiedades que vemos en pantalla: Flex Direction, Flex Wrap, Justify Content, Align Items y Align Content.

Control Info

Control Type	Flex Grid
Auto Grow	False
Flex Direction	Row
Flex Wrap	No Wrap
Justify Content	Flex Start
Align Items	Stretch
Align Content	Stretch

Flex Direction



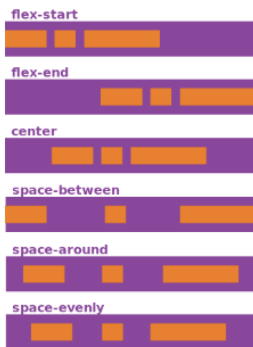
Source: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

Flex Wrap



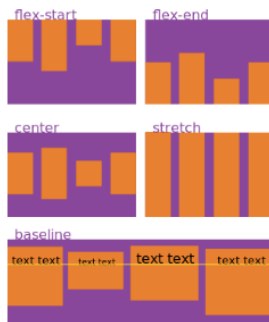
Source: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

Justify Content



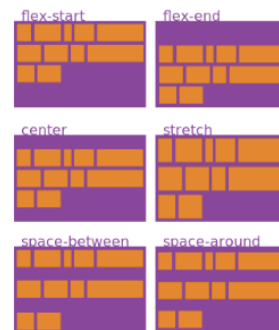
Source: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

Align Items



Source: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

Align Content



Source: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

Flex Direction: Especifica la dirección de los elementos, que se pueden colocar horizontalmente (Rows o Inverse Rows) o verticalmente (Column o Inverse Column)

Flex Wrap: Establece cómo se van a acomodar los elementos cuando ocupan más que el largo de una fila y no pueden mostrarse todos. El valor Wrap los acomoda en la siguiente línea.

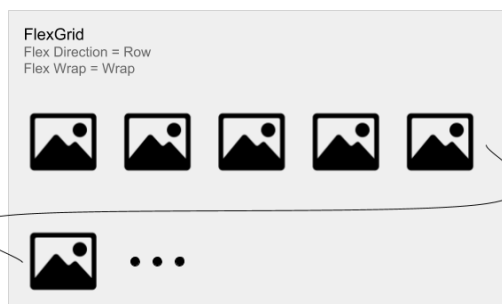
Justify Content: Define la alineación de los elementos en las filas (alineación horizontal si Flex Direction se especifica como Row y alineación vertical si Flex Direction se especifica como Column). Si la alineación es horizontal, nos permite establecer cómo se va a alinear el contenido, si de izquierda a derecha, de derecha a izquierda, centrado, que se ajuste la separación para ocupar todo el ancho disponible o si se agregan espacios entre los elementos.

Align Items: Define la alineación de los elementos en las filas (alineación vertical si Flex Direction se especifica como una fila y alineación horizontal si Flex Direction se especifica como una columna)

Align Content: Nos permite alinear la fila cuando hay espacio adicional en el contenedor.

Estas propiedades también aplican al Flex Control, que es un contenedor flexible que veremos en un momento.

Ejemplo de uso del Flex grid



Layout | Rules | Events | Conditions | Variables | Documentation

Application Bar

MainTable | Grid1

Country: &CountryId

Name From: &AttractionNameFrom

Name To: &AttractionNameTo

GRID

- AttractionId
- CountryId
-
- AttractionName
- CityName
- CountryName
- Trips: &Trips

Total Trips: &TotalTrips

Control Info	
Control Type	Flex Grid
Auto Grow	True
Flex Direction	Row
Flex Wrap	Wrap
Justify Content	Flex Start
Align Items	Stretch
Align Content	Stretch

Supongamos que queremos ver una galería de fotos de las atracciones, con los datos de cada atracción debajo de la foto. Queremos que las fotos fluyan horizontalmente y que automáticamente pasen a una nueva fila para evitar el desplazamiento horizontal.

Como vimos antes, algunas fotos están en formato apaisado y otras en formato vertical, por lo que sus dimensiones se desconocen en el momento del diseño.

Para lograr esto usaremos el Flex Grid, con la propiedad Flex Direction en el valor "Row" y Flex Wrap en el valor "Wrap". Dejamos el resto de las propiedades por defecto y vamos a ejecutar.

Ejemplo de uso del Flex Grid

The screenshot shows a web application interface for a travel agency. At the top, there is a navigation bar with the text "View Attractions, More Info" and the "TRAVEL AGENCY" logo. Below the logo, there are form fields for "Country" (set to "(None)"), "Name From", and "Name To". A horizontal flex grid table is displayed below these fields, showing a row of images. To the right of the application, two configuration panels are visible:

Table: Grid1Table

Control Name	Grid1Table
Layout Name	
Appearance	
Columns Style	400dip
Rows Style	pd;pd;400dip;pd;pd;pd;pd

Control Info

Control Type	Flex Grid
Auto Grow	True
Flex Direction	Row
Flex Wrap	Wrap
Justify Content	Space Around
Align Items	Stretch
Align Content	Stretch

At the bottom of the application, there is a footer with the text "Total Trips 20".

Vemos que intentó hacer un wrap pero como había espacio suficiente, no llegó a hacerlo.

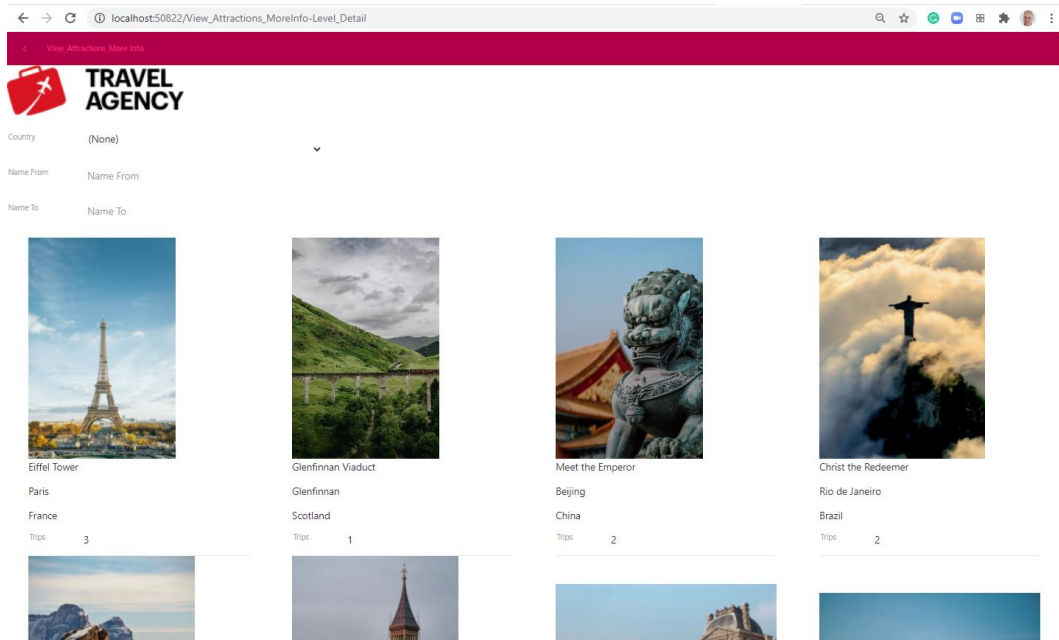
También vemos que no se dejó lugar suficiente para ver los datos de las atracciones con claridad.

Para arreglar esto, vamos a darle un valor de 400 dips a la columnas de la tabla del grid, la Grid1Table, de forma que ocupe más espacio y el Flex Grid deba hacer el wrap.

También cambiamos la propiedad Justify Content del Flex grid con el valor Space Around para que agregue espacios entre las atracciones y no queden pegadas unas a otras.

Ejecutamos.

Ejemplo de uso del Flex Grid



Vemos que ahora cada columna ocupa más espacio y sí se hace el wrap, mostrando las atracciones hacia abajo automáticamente una vez que se ocupa el espacio disponible en la fila.

También vemos que la separación entre atracciones es la adecuada y que la información se muestra bien.

Supongamos que deseáramos que la información del nombre de atracción, ciudad, país y total de viajes salga a la derecha de la foto y no abajo.

Si queremos cambiar el layout de cada item del grid, debemos usar un Flex Control.

Flex Control

Layout * Rules Events Conditions Variables Documentation

Application Bar

MainTable Grid1 Grid1Table Table1 AttractionId

Country &CountryId

Name From &AttractionNameFrom

Name To &AttractionNameTo

GRID

AttractionId

CountryId

AttractionName

CityName

CountryName

Trips &Trips

Total Trips &TotalTrips

Flex: Table1

Control Name	Table1
Appearance	
Class	Table
Background	(none)
Visible	True
Invisible Mode	Keep Space
Enabled	True
Scroll Behavior	
Scroll Factor	1
Zoom Factor	0
Scroll Attachment	Parent
Layout Behavior	
Expand Bounds	Background Only
Expand Bounds Direction	Top, Left, Bottom, Right
Flex Direction	Column
Flex Wrap	No Wrap
Justify Content	Flex Start
Align Items	Stretch
Adjust Container Size	False

Control Info

Control Type	Flex Grid
Auto Grow	True
Flex Direction	Row
Flex Wrap	Wrap
Justify Content	Space Around
Align Items	Stretch
Align Content	Stretch

Table: Grid1Table

Control Name	Grid1Table
Appearance	
Columns Style	400dip;100%
Rows Style	400dip
Width	100%
Height	400dip
Auto Grow	True

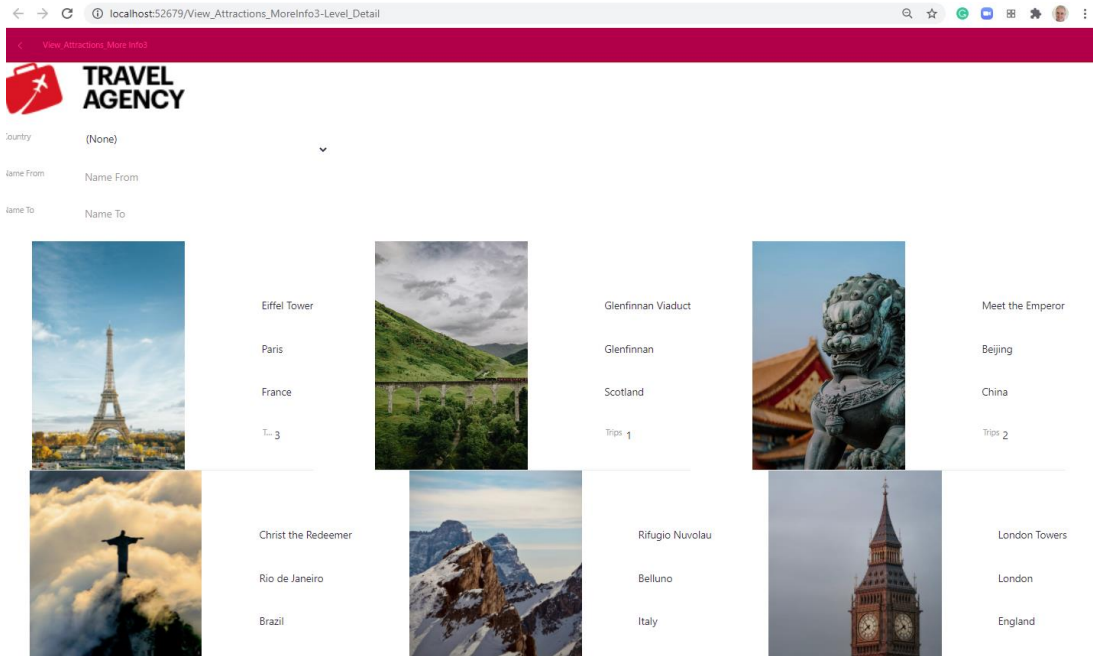
El Flex Control (también llamado Flex Table) es un contenedor de controles, como las Tablas o las Tablas Responsivas. Su objetivo es ofrecer nuevas formas de organizar los controles en la interfaz de usuario, incluso si se desconocen los tamaños de los controles contenidos.

Para seguir con el ejemplo que nos propusimos, salvamos el panel View_Attractions_MoreInfo2 con el nombre View_Attractions_MoreInfo3. Y arrastramos un Flex Control dentro del grid, a la derecha de los otros controles.

Vemos que queda con el nombre Table1 y si vamos a sus propiedades, reconocemos las mismas propiedades que teníamos para el Flex Grid.

Ponemos la propiedad Flex Direction en Column y movemos todos los datos de la atracción menos la foto, para dentro de la Flex Table. El Flex Grid todavía tiene las mismas propiedades que en el ejemplo anterior, y a la tabla del grid, cambiamos el valor de la fila de pd a 400 dips. Y ejecutamos.

Ejemplo de uso del Flex Control



Ahora vemos a la información de la atracción a la derecha. Luego si queremos después podemos ajustar los valores de las propiedades del Flex Control para mejorar la forma de ver los datos.

En este video vimos cómo podemos utilizar distintos tipos de grids y configurarlas para ver la información de variadas maneras.

En siguientes videos seguiremos mejorando la experiencia de usuario de nuestra aplicación.

GeneXus™

training.genexus.com
wiki.genexus.com
training.genexus.com/certifications