

Pasemos a conocer los objetos de tipo **Procedimiento** que podemos definir en una base de conocimiento GeneXus



Este tipo de objeto nos permite definir **PROCESOS** para acceder y navegar tablas de la base de datos con distintos objetivos:

 \rightarrow por ejemplo, podríamos necesitar navegar los registros de cierta tabla, que cumplan con determinadas condiciones y actualizar para dichos registros cierto atributo con determinado valor

AttractionId AttractionName AttractionId AttractionName 1 Louvre Museum 2 The Great Wall 3 Eiffel Tower	Gene)			
AttractionId AttractionName 1 Louvre Museum 2 The Great Wall 3 Eiffel Tower				
AttractionId AttractionName AttractionId AttractionName 1 Louvre Museum 2 The Great Wall 3 Eiffel Tower				
AttractionId AttractionName 1 Louvre Museum 2 The Great Wall 3 Eiffel Tower				
AttractionId AttractionName 1 Louvre Museum 2 The Great Wall 3 Eiffel Tower				
Attraction Table 1 Louvre Museum 2 The Great Wall 3 Eiffel Tower	Visites	AttractionName	AttractionId	
2 The Great Wall 3 Eiffel Tower	8245	Louvre Museum	1	traction Table
3 Eiffel Tower	10122	The Great Wall	2	
	11734	Eiffel Tower	3	

ightarrow o navegar cierta tabla, e imprimir todos sus datos en un listado PDF, ordenados por algún criterio

				Genex	۲us
	AttractionId	AttractionName	Countryld	Visites	
	1	Louvre Museum	2	8245	
	2	The Great Wall	3	10122	
	3	Eiffel Tower	2	11734	
		Att	ractions Lis	t	
Attraction Table		Id Name	Country France	Photo	
Attraction Table		Is Name 1 Louvre Museum 2 Great Wall	Country France China	Photo	

 \rightarrow o también definir procesos específicos que necesitemos resolver, que contengan búsquedas, cálculos, actualizaciones a la base de datos y que vayan imprimiendo información.

		GeneXus
Tables	$a^{+2.088}$ +5.000 +1.500 +1.125 +1.062 $a^{2+b^{2}}c^{2}$	



En primer lugar comencemos por definir un procedimiento para listar todas las atracciones turísticas que ofrece la agencia de viajes, ordenadas alfabéticamente.

Así que creamos un nuevo objeto, esta vez de tipo procedure, al que llamamos:

New Object			×
Select a Category:	Select a Type:		
Common Workflow Counternation Web Smart Devices	Color Palette Data Provider Data Selector Procedure Structured Data Type Subtype Group Transaction		
Solve procedural problem	s, process, update and report data.		
Name:	AttractionsList		
Description:	Attractions List]
		Create	Cancel

Una vez creado el objeto vemos que GeneXus nos posiciona en una sección llamada Source:

Start Page × 2 AttractionsList ×	•
Source Layout Rules Conditions Variables	
Subroutines	
1	ਥ
	-
	~
<u>↓</u> <	>

Aquí es donde escribiremos comandos y órdenes que permitan al procedimiento cumplir con el objetivo para el cual lo hemos creado, en nuestro caso imprimir un listado de atracciones turísticas.

Ahora observemos esta otra sección llamada Layout:

Start Page X httractionsList * X	-
Source Layout Rules Conditions Variables	
D · · · · · · · · · · · · · · · · · · ·	1.1.1.1.1.8.1.1
□ 🚍 printBlock1	
· · · · · · · · · · · · · · · · · · ·	
1	
۲	
	Zoom Factor: 100

El layout es el lugar destinado al diseño de la salida, es decir, donde especificaremos cómo deseamos ver nuestros datos.

$$Page\mathbf{3}$$

Se compone de **printblocks**

🐼 Start Page X 🐉 AttractionsList* X
Source Layout Rules Conditions Variables
2 1 1 2 1 3 1 4 1 5 1 6 1 7 1 8
🕒 🚍 printBlock1

y dentro de los printblocks incluiremos lo que deseamos mostrar.

Podemos querer mostrar títulos, líneas, rectángulos, imágenes, así como también valores de atributos o variables. Para ello los arrastraremos dentro del printblock.

🐼 Start Page X 🕃 AttractionsList* X	1	Toolbax	4 ×
Source Layout Rules Conditions Variables	Ξ (Controls	
D · · · · 1 · · · 1 · · · 2 · · · 1 · · · 3 · · · 1 · · · 4 · · · 1 · · · 5 · · · 1 · · · 6 · · · 1 · · · 7 · · · 1 · · · 8 · · ·	abi	Attribute/Variable	l .
🖸 🚍 printBlock1		Image	l.
	-	Line	l.
		Rectangle	l.
	Α	Text Block	l.
Zoom Factor: 100	1		

Observemos que el layout automáticamente contiene un printblock

En este printblock, podemos querer incluir un título, o la fecha del día, y también podremos agregar más printblocks en esta sección, como veremos.

Pensemos ahora cómo queremos que se vea nuestro listado, así podremos proceder a definirlo. Podría ser algo así...

					GeneXus
1000 00		Attraction	is List		
_	ld	Name	Country	Photo	
	1	Louvre Museum	France		
	2	The Great Wall	China	XX	
	3	Eiffel Tower	France	A	

con una imagen un título



columnas

y todas las atracciones turísticas ordenadas alfabéticamente

Notemos que queremos mostrar 2 contenidos fijos, como es el título del listado con la imagen y los títulos de las columnas con la línea debajo

y luego, la información que queremos mostrar de las atracciones

la cual irá cambiando línea a línea, ya que queremos mostrar los datos que tengamos almacenados en nuestra base de datos.

Podríamos definir entonces tres printblocks: uno en el cual incluiríamos el título del listado y la imagen

1000 000	Attract	ions List	
ld	Name	Country	Photo
19.	Louvre Museum	France	
2	The Great Wall	China	XXX
3	Eiffel Tower	France	A

otro para mostrar los títulos de las columnas con la línea debajo

					GeneXus
		Attraction	ns List		
_	ld	Name	Country	Photo	
	1	Louvre Museum	France		
	2	The Great Wall	China		
	3	Eiffel Tower	France	A	
					'

y un tercer printblock

						GeneXus	
	1000 000		Attraction	s List			
		ld	Name	Country	Photo		
		1	Louvre Museum	France			
		2	The Great Wall	China	X		
		3	Eiffel Tower	France	A		
1							

en el cual vamos a desplegar los datos de las atracciones turísticas.

Empecemos entonces a definir esto.

Podemos usar el printblock que se creó cuando creamos el objeto procedimiento para el título y la imagen.

Empecemos por el título. Para esto desde la Toolbox... arrastramos el control Text Block... editamos sus propiedades... y en la de nombre Text escribimos "AtractionsList". Modificamos también su color, su fuente..., y lo ubicamos donde queremos que aparezca en relación a los márgenes.

🐼 Start Page X 🛟 AttractionsList* X		Properties	4 ×
Source Layout * Rules Conditions Variables		🔠 21 🌾 Filte	r 🛛 🗙
ע י י י ו י י י 1 י י י 1 י י י 2 י י י ו י י 3 י י ו י י 4 י י י ו י י 5 י י י ו י י 6 י י י ו י י 7 י י	8	V Label: gxRepo	rtLabel1
□		Name	gxReportLabel1
		Text	Attractions List
Attractions List		х	333
āā		γ	33
		Width	141
		Height	25
gxReportLabel1 { Loc:333.33 Size:141.25 }	Zoom Factor: 100	Fore Color	MidnightBlue
		Back Color	Transparent
Output	4 ×	Borders	None
Show: Build Y Find:	♣ Autoscroll	Border Width	1
Compiling viewattractionsuccess	^	Border Color	Black
Compliing attractionattractionwcsuccess		Alignment	TopLeft
Compiling homesuccess		> Font	Microsoft Sans Serif, 14p
Compiling gx00a0success Compiling gx00b1success		Word Wrap	False

Nombraremos a este printblock con un nombre claro que represente lo que está mostrando. Así que seleccionamos las propiedades del printblock y editamos su propiedad Name, asignándole como nombre "Title".

🐼 Start Page X 🛟 AttractionsList* X	O Properties	. # ×
Source Layout * Rules Conditions Variables	Pilter 2 ↓ 🐒 Filter	\times
<u>y</u>	 Printblock: Title 	
🖸 🗮 Tâle	Name Title	
	Height 100	
Attractions List		
Title { Height: 100 } Zoom Factor: 100		

Page**(**

Insertemos, ahora, la imagen con el avión, a la izquierda. Para ello arrastramos desde la Toolbox el control Image y lo soltamos donde deseamos colocarla. Observemos que se nos abre esta ventana que nos permite seleccionar alguna de las imágenes existentes en la base de conocimiento, o incorporar una nueva, por ejemplo importándola de un archivo:



Este botón nos permite explorar nuestro sistema de archivos y elegir la imagen, que se creará como objeto GeneXus de tipo Image con el mismo nombre del archivo imagen como nombre por defecto . De allí en más podremos utilizar la imagen dentro de nuestra KB libremente.

Select Image	×
Filter (none) ActionCancel ActionDelete	
Act Import Image From File	×
Act: Bai Name plane Con Ge File C:\Images\plane.png Pa	
Pa Pa Pa Pa Pa Pa Pa Pa	OK Cancel
selection file New External Image	OK Cancel



Page

_	-			
ы			3 · ·	
		🗆 🚍 Title		Ĺ
	[
-	ļ			
1	Ļ	Attractions List		
-	Ļ			
-	Ļ			
1	Ļ			
	ŀ			. 🔽
•	ŀ		2	>

Observemos que aquí podemos acceder a todas las imágenes de la KB entre las que se encuentra la de nuestro avión.

Ki Explorer + X Ki Shar Rugo - M Associated in the Images Tool Vindow X -	Properties	7 ×
Open: Name or Pattern Filter 💷 - 🎇 New Image Advanced 🗸	2↓ 🐺 Filter	\times
Category None Decorption Category Category Last Update Category Category None Category Category Category	Image: plane Image: plane Description Qualified Name plane	
> The Design of the Column of		
Show: General V Find:		
> ☐ Generators	Properties T Toolbox 🗔 Form P	Preview

Ahora vamos a crear otro printblock para incluir en él los títulos de las columnas, con una línea debajo. Si presionamos el botón derecho del mouse **sobre determinado printblock** y seleccionamos la opción "Insert Printblock"



se insertará un nuevo printblock debajo de él.



Start Page × 🐉 AttractionsList* ×
Source Layout * Rules Conditions Variables
🖸 🖻 Title
•
Attractions List
🗉 🖻 🚍 printBlock 1
]

El orden de los printblocks en el Layout no es importante, ya que no será necesariamente el orden en el que se imprimirán. **Definimos cuándo imprimir cada printblock** en el código que vamos a escribir en el Source del procedimiento.

🐼 Start Page X 👌 AttractionsList* X
Source Layout * Rules Conditions Variables
ا
🗆 🚍 Title
Attractions List
Teodo and
■
*
1

Veremos esto en breve.

Ahora a este nuevo printblock le pondremos como nombre "ColumnTitles"



... y ahora vamos a insertar en este nuevo printblock un TextBlock por cada texto que queremos mostrar como título de columna.

$$_{Page}9$$

Así que desde la Toolbox, arrastramos el text block

Y en su **propiedad** Text escribimos "Id".



Agregamos otro Textblock:

Y en su propiedad Text agregamos el texto "Name":

🐼 Start Page X 🐉 AttractionsList* X	Properties	4 ×
Source Layout * Rules Conditions Variables	🔡 🖞 🌾 Filte	×
ע י ו י י י 1 י י י 1 י י י 2 י י י ו י י 3 י י י ו י י 4 י י י ו י י 5 י י י ו י י י 6 י י י ו י י י 7 י י ו י י 8 י	V Label: gxReport	rtLabel3
E Title	Name	gxReportLabel3
	Text	Name
Attractions List	x	283
	Y	33
	Width	30
Column Titles	Height	14
	Fore Color	Black
	Back Color	Transparent
	Borders	None
1	Border Width	1
	Border Color	Black
	Alignment	TopLeft
	> Font	Microsoft Sans Serif, 8pt
	Word Wrap	False

y ahora otro Textblock para mostrar el texto "Country". Y por úlitmo un text block para el título "Photo".

Ubicamos los controles en la posición que queremos... Los podemos alienar seleccionándolos todos y luego:





Por último, vamos a insertar una línea debajo de estos títulos de columnas.

Así que volvemos a la ToolBox, y arrastramos un control "Line". Arrastramos desde aquí... dando el largo que deseamos...



Si ahora en el Source escribiéramos las órdenes para que se imprima el primer printblock y a continuación el segundo, estos espacios en blanco de aquí serán respetados:

🐼 Start Page X 🗳 AttractionsList* X
Source Layout * Rules Conditions Variables
• • • • • • • • • • • • • • • • • • •
Attractions List
E Column Titles
·····
id Name: Country: Photo:
r

Vamos a reducir los del segundo printblock:

🐼 Start Page X 🖕 AttractionsList* X	•
Source Layout * Rules Conditions Variables	
1 • • • • • • • • • • • • • • • • • • •	
📃 🖻 Title	
•	
Attractions List	
E Column Titles	
Id Name: Country: Photo:	

Nos está faltando ahora agregar el tercer printblock que habíamos dicho, para mostrar los datos de las atracciones turísticas, así que insertamos un nuevo printblock... y le ponemos como nombre: Attractions.

🖾 Start Page 🗙 🐉 AttractionsList* 🗙	*
Source Layout * Rules Conditions Variables	
D • 1 • • • 1 • • • 2 • • • 1 • • • 3 • • • 1 • • • 4 • • • 1 • • • 5 • • • 1 • • • 6 • • • 1 • • • 7 • • • 1 • • • 8 • •	
📜 Tile	
Attractions List	
-	
E Column Titles	

Como los datos están almacenados en atributos, vamos nuevamente a la Toolbox, seleccionamos un control del tipo "Attribute/Variable"

y lo arrastramos bajo el título "Id"...



En esta ventana que se nos abre elegimos qué variable o atributo queremos mostrar en el control. Veamos que además de &today, en un procedimiento existen estas otras variables del sistema.

Filt	ters				
Patt	em:				
Obie	ect Type:				
, Obje	ect:			-	
Cre	eate New Attribut	e/Variable	Control		
			1-		
_	Name	Module	Туре	Description	
&	Today		Date	Today	
&	Time		Character(8)	Time	
&	Pgmname		Character(128)	Pgmname	
&	Pgmdesc		Character(256)	Pgmdesc	
&	Page		Numeric(6.0)	Page	
&	Line		Numeric(6.0)	Line	
&	Output		Character(3)	Output	
Att.	AirlineDiscountPer		Percentaje	Airline Discount	
At.	Airlineld		ld	Airline Id	
Att.	AirlineName		Name	Airline Name	
Att.	AirportId		ld	Airport Id	
Att.	AirportName		Name	Airport Name	
Att.	AttractionAddress		Address, GeneX	Attraction Address	
A#	AttractionId		ld	Attraction Id	
Att.	AttractionName		Name	Attraction Name	

 $_{\rm Page} 12$

...queremos que muestre el atributo "AttractionId", así que filtramos por "At" y así lo encontramos con facilidad:

'att	em: At			
)bje	ect Type:			*
)bje	ect:			*
Сго	eate New Attribu	te/Variable Co	ntrol	
1	Name	Module	Type	Description
4#	AttractionAddress	modulo	Address, GeneXus	Attraction Address
4 #	AttractionId		ld	Attraction Id
4#	AttractionName		Name	Attraction Name
A#	AttractionPhoto		Image	Attraction Photo

Lo alineamos a la derecha respecto al título "Id"... (El último control seleccionado es el que sirve de referencia: vemos que así aparece resaltado gráficamente



También podemos insertar atributos en un printblock desde la opción Insert / Attribute



—			
TravelAgency GeneXus 15	Trial		
File Edit View Layout Ins	ert Build Kno	owledge Manager	Window Tools Help
🖺 I 🛄 🛄 🔏 🗈 🖻	Object	Ctrl+Shift+J	: Environment 👻 🚽
🛅 KB Explorer	Attribute	Ctrl+Shift+A	AttractionsList X
Open: Name or Pattern	Domain	Ctrl+Shift+D	The second s
V F TravelAgency	Variable	Ctrl+Shift+V	Rules Conditions Variables
V 🛞 Root Module	Function	Ctrl+Shift+F	. 1 1 2 1 3 1 4 1 5 1 6 1 7 1 8
> 🦳 GeneXus	Event	Ctrl+Shift+E	
Airline abl	Attribute/Va	riable	
Airport 🛌	Image		Attractions List
> 🔲 Attraction	line		
AttractionsList			
Category	Rectangle		
> Country	Text Block		Titles
Customer			ld Name County Photo:
🖧 Diagram1			
🐣 Diagram2		= 🖸 🚍 Attract	ions
🖧 Diagram3			
🖁 Diagram4		•	Atra
Flight			
A FlightArrivalAirport	t		

Aquí sólo se ofrecen atributos, no variables. En este cuadro podemos elegir varios atributos a la vez.

GX Start Page X 🔥 Attraction	inal int								
Source Layout * Rules Co	Insert	Attributes				×			
	Fil	ters							
🗆 🚍 Title	Patt	em:	At						
	Ohie	not Turno:	-	/7k					
	Object Type:								
-	Obje	eci.				*			
	Se	lect one or	more	attributes					
E E Column Titles		Name		Module	Туре	Description			
· · · · · · · · · · · · · · · · · · ·	Att.	AttractionAddre	ess		Address, GeneXus	Attraction Address			
	Att:	AttractionId			ld	Attraction Id			
Attractions	Att.	AttractionName	в		Name	Attraction Name			
	Att.	AttractionPhoto)		Image	Attraction Photo			
<									
Attractions { Height:91 }									
Output									
Show: General					0	Cancer			

Así que seleccionamos AttractionName presionamos Control, y AttractionPhoto.

Presionamos OK...

Ubicamos los atributos debajo de los títulos...

Start Page X 🐉 AttractionsList* X
Source Layout * Rules Conditions Variables
y • • • 1 • • • 1 • • • 1 • • • 2 • • • 1 • • • 3 • • • 1 • • • 4 • • • 1 • • • 5 • • • 1 • • • 6 • • • 1 • • • 7 • • • 1 • • • 8 • •
Attractions List
😑 🖂 Solumn Titles
Id Name: Country: Photo:
E C 🚍 Attractions
Attra AttractionName

E insertamos CountryName, de igual forma:

🐼 Start Page 🗙 🎳 Attraction	nd int	• •				
Source Layout * Rules Co	nsert	Attributes				×
× · · · · · · · · · · · · · · · · · · ·	Filt	ters				. 7
🗆 🚍 Title	D-#		,			
	Obic	ent. Co	,			
	Obje	sct type.			-	
1000	0.030				•	
	Se	lect one or mo	ore attributes			
= 🖃 🚍 Column Titles				-	a 1.4	
•	4.11	Name	Module	Туре	Description	
	AU.	Countryld		Namo	Country Id	
🗉 🖃 🚍 Attractions	710.	Country Name		Name	Country Name	
•						
*						
						• • • • • • • • • • • • • • • • • • • •
						>
Attractions { Height:91 }						Zoom Factor: 100
Cutput						₽ X
					OK Cancel	• • • • • •
Show: General						

Alineamos los atributos . Y quitamos los espacios en blanco del printblock.

[D]	7
🖸 🚍 Title	
·	
Attractions List	
1-00 -00-	
· · · · · · · · · · · · · · · · · · ·	
😑 🖃 🗮 Column Titles	
ld: Name: Country: Photo:	
Attra AttractionName CountryName Attrac;	



Vamos a poner en negrita los títulos de las columnas, para que se destaquen. Los seleccionamos a todos a la vez, y en las propiedades vemos cómo podemos aplicar de una vez el cambio al Font.



ĸ	
	🖃 🚍 Title
-	
-	Attractions List
	🖃 🚍 Column Titles
-	Id Name: Country Photo:
1	Attra AttractionName CountryName : Attrac;
-	

No nos dimos cuenta de una cosa: la foto de la atracción no podrá aparecer en ese espacio tan reducido que le dejamos al control atributo AttractionPhoto. Tendremos que agrandarlo un poco.

D	
	🖸 🚍 Title
-	
-	Attractions List
-	
=	😑 🚍 Column Titles
-	id Name Country Photo
Ŀ	
≡	🗆 🚍 Attractions
	AttractionPhoto
1	Attra AttractionName CountryName
-	

Ya tenemos listo el diseño de cómo queremos que se despliegue la información en el listado.

Ahora falta escribir el código necesario para obtener la información apropiada de la base de datos e ir indicando que se impriman los printblocks en el orden que deseamos.

Vamos entonces a la opción Source...

🐼 Start Page 🗙 🐉 AttractionsList* 🗙	*
Source ayout * Rules Conditions Variables	
Subroutines	
1	E
	^
	\sim
	>

Y lo primero que queremos que se imprima, es el título del reporte, así que escribimos "print Title":



Como las instrucciones que escribamos en el Source, se ejecutarán de arriba hacia abajo, **ésta instrucción será la primera que se va a ejecutar**. Estamos indicando con ella que se imprima el contenido del printblock de nombre Title, es decir el título del listado.

El comando Print siempre debe tener a continuación el nombre de un printblock definido en el Layout.



Page_

Lo siguiente que deseamos, es imprimir los títulos de las columnas, así que tenemos que dar la orden de imprimir el printblock "ColumnTitles"...

Así que escribimos "print... ColumnTitles":

🐼 Start Page 🗙 🦆 AttractionsList* 🗙	Ŧ
Source * Layout * Rules Conditions Variables	
Subroutines	
1 print Title	15
2 print ColumnTitles	1

Con estas dos instrucciones hemos indicado que se imprima la parte fija del reporte, es decir, la que no variará de acuerdo a los datos: la que tiene el título del reporte y la imagen del avión y la que tiene los títulos de las columnas.

GeneXus AttractionId AttractionName Countryld CityId Louvre Museum 2 1 1 Attraction The Great Wall 2 3 1 table 3 Eiffel Tower 2 1 CityId City 1 Rio de Janeiro 1 Sao Paulo 1 2 CountryCity table 2 1 Paris 3 1 Beijing 3 2 Shanghai 3 3 Hong Kong 1 Brazil Country 2 France table 3 China

Pero ahora necesitamos imprimir la información de las atracciones, que está almacenada en la base de datos.

Para eso, debemos acceder a la tabla física que tiene almacenada dicha información, o sea a la tabla asociada a la transacción Attraction



Page **1**

El comando que nos permite acceder a una tabla física, es el comando "For Each". La tabla física accedida se denomina tabla base del for each.

							G	ieneXus [°]	
For each	7	AttractionId	Attracti	onName Museum	Countryld 2	Cityld 1			
Attraction	T	2 3	The Gr	eat Wall	3	1		Attraction table	
		Countryld	CityId	Ci	ityName				
		1	1 2	Rio	de Janeiro ao Paulo	Co	untryCi	ity	
		2 3	1 Paris 1 Beijing			tab	table		
		3 3	2 3	S Ho	hanghai ong Kong				
		Countryld	Coι	intyName					
		1 2	Brazil Cou France table			country able			
		3	_	China	_				

Escribimos entonces el comando For Each...

🐼 Start Page 🗙 🎳 AttractionsList* 🗙
Source * Layout * Rules Conditions Variables
Subroutines 🗸
1 print Title 2 print ColumnTitles 3 ⊒ For each

y al lado: Attraction.



¿Por qué escribimos Attraction al lado de For each?

porque es el nombre de la transacción cuya tabla física asociada queremos navegar...

.. y ahora.. dado que queremos imprimir de cada atracción turística, el contenido de los atributos AttractionId, AttractionName, CountryName, y AttractionPhoto escribimos la orden para imprimir el printblock "Attractions" que los contiene. Así que escribimos Print Attractions:

$$_{\rm Page}19$$



y luego cerramos al comando For Each con la instrucción Endfor



¡Y listo!

De esta forma, le hemos indicado a GeneXus que tiene que navegar la tabla física ATTRACTION, correspondiente a la transacción Attraction:



y como dentro del For each, hemos invocado a un printblock que contiene atributos de las tablas ATTRACTION y COUNTRY, aplicando el concepto de tabla extendida, para cada atracción navegada, se accederá a la tabla COUNTRYCITY, y de ésta a COUNTRY, para obtener el nombre del país donde se encuentra dicha atracción:



		GeneXus
For each	AttractionId AttractionName CountryId Country CountryId CountryId	ityld 1 Attraction 1 table
	CountryldCityldCityName11Rio de Janeiro12Sao Paulo21Paris31Beijing32Shanghai33Hong Kong	CountryCity table
	Countryld CountyName 1 Brazil County 2 France table 3 China County	ту

Vamos entonces a ejecutar para ver el resultado.

Pero primero tenemos que definir algunas propiedades necesarias para que se imprima el listado con formato PDF. [para las props usar vista categorizada, no alfabética que apabulla!!!]Vamos a las propiedades del reporte y en la propiedad "Main program" seleccionamos True.

Luego en la propiedad "Call protocol" seleccionamos "HTTP".

• Properties			Ţ	х
🗎 🧎 🌾 Filter				×
Procedure: Attra	ctionsLis	st		^
Name	Attract	ionsList		
Description	Attractio	ons List		
Module/Folder	Root M	odule		
Main program	True			
Call protocol	HTTP			
Execute in new Ll	False			
Qualified Name	Attracti	onsList		
Object Visibility	Public			
> Web informat	ion			
✓ Main object pr	operties			
Application titl				
Application icc				

Y por último tenemos que insertar la regla OutputFile en la sección de las reglas...porque como se ve, un objeto de este tipo también permite definir algunas reglas –aunque menos que en una transacción–, así que seleccionamos Insert/Rule



Y completamos poniendo el nombre del archivo del listado "AttractionsList.PDF" y luego el formato que vamos a utilizar: "PDF".

🐼 Start Page 🗙 🍦 AttractionsList* 🗙
Source * Layout * Rules * Conditions Variables
<pre>1 = Output_file('AttractionsList.pdf', 'pdf'); 2 3</pre>

Salvamos... y ahora ya podemos ejecutarlo.

iy vemos que se crea el listado! ... con el formato que definimos... y aparecen listadas todas las atracciones turísticas que habíamos ingresado, cada una de ellas con el nombre del país al que pertenece y la foto.



Page 4

Video filmado con GeneXustm15

Volvamos a GeneXus

Aquí denominamos a Attraction como: Base transaction

For each Attraction Print Attractions Endfor BASE TRANSACTION

y como dijimos antes, corresponde al nombre de la transacción, cuya tabla física asociada queremos navegar. En nuestro caso, la tabla Attraction, de igual nombre que la transacción.

¿Y si la transacción tuviera más de un nivel? Como Flight por ejemplo... y quisiéramos navegar la tabla física asociada al 2do nivel de la transacción? Es decir: a los asientos del vuelo.

La sintaxis en ese caso sería la siguiente:



O sea: el nombre de la transacción:

For each Flight.Seat

Punto

Y pegado al punto, el nombre del nivel:

$${\rm Page}23$$



Bien. La tabla física que el For each navegará, así como otras decisiones que toma GeneXus, son mostradas en el listado de navegación del procedimiento

Este listado es creado automáticamente cuando se genera el procedimiento para ser ejecutado. En nuestro caso, fue luego del F5.

En él GeneXus nos indica cómo accede a la información de la base de datos. Vamos a observarlo

'attem:	Procedure Attra	actionsList Navigation Repo	rt		_
AttractionsList	Name Description Output Devices Main	* <u>AttractionsList</u> Attractions List File Yes	Environment Spec. Version Form Class Program Name Call Protocol Parameters	Default (C#) fs_0_0-104342 Graphic AttractionsList HTTP	
	Levels				
	For Each Attrac	ction (Line: 10)			
	Order: Navigation filters: Join location: E= <u>Attr</u>	AttractionId Index: IATTRACTION Start from: FirstRecord Loop while: NotEndOfTable Server action (AttractionId) =Country (CountryId)			

Podemos ver que al lado de donde dice "For Each", dice también Attraction:

GX Start Page X 🌓 Attracti	onsList × 🔳 Navigati	ion View ×			•
Pattem: AttractionsList	Procedure Attrac Name Description Output Devices Main	tionsList Navigation Repo <u>}</u> <u>AttractionsList</u> Attractions List File Yes	rt Environment Spec. Version Form Class Program Name Call Protocol Parameters	Default (C#) f 15_0_0-104342 Graphic AttractionsList HTTP	^
	Levels For Each Attract Order: Navigation filters:	ion (Line: 10) AttractionId Index: IATTRACTION Start from: FirstRecord Loop while: NotEndOfTable			
	Join location: S	Server <u>tion (<i>AttractionId</i>) =Country (<i>CountryId</i>)</u>			~

para indicarnos que ésta es la tabla base del For Each.

Recordemos que el For each recorre una tabla física, por lo que el nombre Attraction que aparece en el listado de navegación es el de la tabla física ATTRACTION, no el de la transacción base que hemos escrito en el procedimiento. GeneXus deduce esta tabla porque es la tabla asociada a esa transacción base que indicamos.

$${}_{Page}24$$

🐼 Start Page 🗙 🗳 Attraction	nsList X 📋 Navigation View X	•
Image: X Image	Image: Selist X Image: Selist X Image: Selist X Name Image: Selist X Environment Image: Selist X Name Image: Selist X Environment Image: Selist X Description AttractionsList Spec. Version Image: Selist X Output Devices File Form Class Graphic Main Yes Program Name AttractionsList Base TABLE Program Name AttractionsList Call Protocol HTTP Parameters Hitraction Id Index: Index: IATTRACTION Navigation Start from: FirstRecord filters: Loop while: NotEndOfTable	· ·
😵 0 Errors 🛕 0 Warnings 🔗 1	Success	~

También nos indica que para ordenar el listado de atracciones

Pattern:	Procedure Attrac	tionsList Navigation Repo	rt		_
AttractionsList	Name Description Output Devices Main	AttractionsList Attractions List File Yes	Environment Spec. Version Form Class Program Name Call Protocol Parameters	Default (C#) 15_0_0-104342 Graphic AttractionsList HTTP	
	Levels	ion Line: 10)			
	Order: Navigation filters: Join location: S	AttractionId Index: JATTRACTION Start from: FirstRecord Loop while: NotEndOfTable server ction (<u>AttractionId</u>) = <u>Country (CountryId</u>)			

se utilizó el atributo AttractionId (que es la clave primaria de la tabla Attraction)...

Nos indica también que recorrió todos los registros de la tabla: pues empezó por el primer registro e iteró hasta llegar al fin de la tabla.

Image: X Start Page: X Attraction Pattern: Image: X AttractionsList	IsList X INAvigation View X Procedure AttractionsList Navigation Report Name AttractionsList Environment IDescription Attractions List Spec. Version Information Information Spec. Version Information Informatio	·
	Levels For Each Attraction (Line: 10)	
	Order: AttractionId Index: IATTRACTION Navigation Start from: FirstRecord filters: Loop while: NotEndOfTable Join rocation: Server	
🛛 😧 0 Errors 🛕 0 Warnings 🕑 1	Success	~

es decir que mostró todas las atracciones.... y por último... nos indica que la tabla que **navegó** fue Attraction

Pattem:	Procedure Attra	ctionsList Navigation Repo	rt		e
	Description Output Devices Main	Attractions List Attractions List File	Environment Spec. Version Form Class	Default (C#) 15_0_0-104342 Graphic	^
	Levels	103	Program Name Call Protocol Parameters	AttractionsList HTTP	
	For Each Attrac Order: Navigation filters: Join location:	tion (Line: 10) <u>AttractionId</u> Index: IATTRACTION Start from: FirstRecord Loop while: NotEndOfTable Server			
		action (<u>AttractionId</u>) = <u>Country</u> (<u>CountryId</u>)			~

y que **debió acceder** a Country

Start Page >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	sList X 💽 Naviga Procedure Attra Name Description Output Devices Main	tion View X ctionsList Navigation Repo) AttractionsList Attractions List File Yes	rt Environment Spec. Version Form Class Program Name Call Protocol Parameters	Default (C#) 15_0_0-104342 Graphic AttractionsList HTTP	=	•
	Levels For Each Attrac Order: Navigation filters: Join location: E=Attra	tion (Line: 10) <u>AttractionId</u> Index: IATTRACTION Start from: FirstRecord Loop while: NotEndOfTable Server <u>action (AttractionId</u>) <u>= Country (CountryId</u>)				~

para recuperar información, ya que en nuestro listado mostramos al nombre del país.



Volvamos ahora al Source del procedimiento

Una cosa que nos había quedado pendiente es que las atracciones salgan listadas ordenadas en forma alfabética,

Page 26

o sea por nombre de atracción.

Y esto lo logramos simplemente escribiendo luego del "For Each Attraction", la cláusula "order AttractionName"



Vamos ahora a ejecutar el procedimiento para verlo...

Y vemos que ahora las atracciones turísticas son listadas ordenadas alfabéticamente por su nombre.



Si observamos el listado de navegación, vemos cómo lo indicaba:

Pattem:	Procedure Attrac	ctionsList Navigation Re	port		
AttractionsList	Name Description Output Devices Main	AttractionsList Attractions List File Yes	Environment Spec. Version Form Class Program Name Call Protocol Parameters	Default (C#) 15_0_0-104342 Graphic AttractionsList HTTP	
	Warnings A <u>spc0038</u> There startin Levels For Each Attract	is no index for order <u>Attra</u> ng at line 3.	<u>ctionName;</u> poor perfor	mance may be noticed in g	roup
	Order: Navigation filters: Join location:	AttractionName ! No index Start from: FirstRecord Loop while: NotEndOfTabl Server	e		
	Attractio	<u>ction (<i>AttractionId</i>) INTO g</u> nPhoto <u>AttractionId Attract</u> = <u>Country (<i>CountryId</i>) IN</u>	<u>CountryId</u> <u>AttractionPho</u> ionName TO <u>CountryName</u>	<u>:o.Uri</u>	
					\sim

Page 2

Por ahora no prestemos atención a la advertencia que el listado nos informa.

Ahora vamos a observar algo interesante. Y es que GeneXus nos permite ordenar por el valor de un atributo que no está en la tabla ATTRACTION, sino en su tabla extendida.

Cambiemos el atributo del order, por CountryName:



Este atributo no se encuentra físicamente en la tabla base del For each



pero se encuentra en la tabla extendida de la tabla base navegaday por lo tanto podemos ordenar por él.

Ejecutemos el procedimiento para ver el resultado.

Y vemos que jahora salen listadas las atracciones ordenadas alfabéticamente por el nombre de país!





Así como hemos agregado la **cláusula opcional "order"**, al For each, la sintaxis del For each permite que le agreguemos varias cláusulas y definiciones opcionales más, como veremos.

Por ejemplo, ¿qué pasaría si en la agencia de viajes nos piden que listemos solamente las atracciones turísticas de Francia?

1-00 00	2	Attract	ions List	
-	ld	Name	Country	Photo
	2	The Great Wall	China	XXX
	3	Eiffel Tower	France	
	1	Louvre Museum	France	

Solamente agregaremos al comando For Each, una claúsula llamada **Where**, para que filtre y muestre únicamente los datos que cumplan con la condición deseada.

Así que nos posicionamos en el renglón siguiente al For Each y escribimos Where...CountryId=2, ya que sabemos que el Id de Francia es el 2.



Page.

Hagamos F5.

1000	>	Attracti	ions List	
	ld	Name	Country	Photo
	1	Louvre Museum	France	
	3	Eiffel Tower	France	Å

En lugar de filtrar por el identificador de país, podríamos haber escrito Where CountryName='France'.

1	print Title
2	print ColumnTitles
3	□ For each Attraction order CountryName
4	where CountryName = 'France'
5	print Attractions
6	endfor

Salvamos y ejecutamos otra vez el reporte....

¡Y vemos que el resultado es exactamente el mismo!

T-00 -00		Attractions	s List	
_	ld	Name	Country	Photo
	1	Louvre Museum	France	
	3	Eiffel Tower	France	Å

Si observamos ahora el listado de navegación:

🐼 Start Page 🗙 🦆 Attraction	sList X 🖸 Navigation View X	•
Start Page >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	Stist X Image: State of the state of	•
	Levels Image: CountryName For Each Attraction (Line: 10) Image: CountryName Order: CountryName I No index Image: Navigation Navigation Start from: CountryName = 'France' filters: Loop while: CountryName = 'France' Join location: Server Image: Country (CountryId)	
		~
🛛 🚫 0 Errors 🛕 1 Warnings 📀 0	Success	

Vemos que ya no se recorre toda la tabla Attractions. Como estamos ordenando por CountryName, para quedarnos con los países de nombre 'France' solamente tiene que recorrer un pedacito de la tabla y no toda. Es análogo a buscar en un diccionario la palabra 'France'. No se busca en todo el diccionario. Se accede directamente a la "F".

Finalmente actualizamos los cambios en GeneXus Server.

Repasemos ahora los conceptos aprendidos:

El comando For each se utiliza para recorrer cada registro de una tabla y hacer algo con su información relacionada.

	GeneXus
Conceptualization	
For each <i>TransactionName.LevelName</i> endfor	

Para ello le indicamos el nombre del nivel de la transacción cuya tabla asociada queremos recorrer.

Page 3

	GeneXus
Conceptualization	
For each <u>TransactionName.LevelName</u>	

A esta indicación de nivel le llamamos transacción base del for each:

	GeneXus
Conceptualization	
Base transaction For each TransactionName.LevelName endfor	

Y de ese nivel GeneXus inferirá la tabla a recorrer, a la que llamamos **tabla base** del for each.

	GeneXus
Conceptualization	
Base transaction For each TransactionName.LevelName endfor	Base table



El conjunto de atributos que están entre el For each y el Endfor deben pertenecer a la tabla extendida de esa tabla base a recorrer.

	GeneXus
Conceptualization	
For each TransactionName.LevelName	Base table t e n d e d

Aquí resumimos lo que hemos visto hasta el momento del comando for each:

	GeneXus
Conceptualization	
For each BaseTransaction order Att ₁ , Att ₂ ,, Att _n where condition MainCode endfor	Base table t e n d e d

Mediante la cláusula **Order** es posible indicar el criterio por el cual ordenar la información devuelta por el For each. El orden puede ser de acuerdo a los atributos de la tabla base del For each o de su extendida. Como vemos, se puede ordenar por un solo atributo, o por varios.



		GeneXus
C	Conceptualization	
	For each BaseTransaction order Att ₁ , Att ₂ ,, Att _n where condition MainCode endfor	Base table

< ▷ ∅ ₦ ९ ∞

Por ejemplo, si también mostráramos la categoría de la atracción turística en el listado y quisiéramos ordenarlo por nombre de país y dentro de las atracciones que son del mismo país, por nombre de categoría... escribiríamos ambos atributos en forma ordenada: primero CountryName y luego CategoryName...

Attractions	List
Attractions	List
me Country	
Goundy	Photo Category
e Great Wall China	
el Tower France	Monument
vre Museum France	Museum
H	tel Tower France

Aquí tanto CountryName, como CategoryName no están presentes en la tabla base, Attraction, sino en tablas de la extendida:

 $P_{age}34$



Para filtrar la información devuelta por el For each se utiliza la cláusula **Where**, en la que especificamos la condición que deberán cumplir los registros para ser elegidos:

	GeneXus
Conceptualization	
For each BaseTransaction order Att ₁ , Att ₂ ,, Att _n where <u>condition</u> MainCode endfor	Base table t e n d e d

La condición puede ser compleja, incluyendo varias condiciones unidas por AND u OR, es decir, por ejemplo:

• Condition1 AND Condition2: lo que significa que ambas deben cumplirse a la vez

		Ge	eneXus
Conceptualization			
For each order Att, where co <u>Main</u> endfor	BaseTransaction , Att ₂ ,, Att _n ndition Code condition ₁ and condition ₂	Base table	E x t e n d e d

• Condition1 **OR** Condition2: lo que significa que de cumplirse una de las dos, ya alcanza para que el registro que se está evaluando pase el filtro:

	GeneXus
Conceptualization	
For each BaseTransaction order Att ₁ , Att ₂ ,, Att _n where condition MainCode endfor condition ₁ and condition ₂ condition ₁ or condition ₂	Base table t e n d e d

También podemos colocar varias cláusulas **Where**, lo que es lo mismo que escribir una sola, con las condiciones unidas por **AND**:

 ${}^{\rm Page}36$

	GeneXus
Conceptualization	
For each BaseTransaction order Att ₁ , Att ₂ ,, Att _n where condition ₁ and where condition ₂ and where condition _n and MainCode endfor	Base table t e n d e d

Dentro del comando for each, en su **código principal**, se escriben los comandos que queremos que se ejecuten uno a continuación del otro para realizar paso a paso lo que se necesite con el registro de la tabla base sobre el que se está posicionado en cada momento... y los asociados por tabla extendida:

	GeneXus
Conceptualization	
For each BaseTransaction order Att ₁ , Att ₂ ,, Att _n where condition ₁ where condition ₂ where condition _n <u>MainCode</u> endfor	Base table t e n d e d

Por ejemplo, imprimir un printblock:



				G	eneXus
Conceptualization					
print Title			Attraction	s List	
For each Attraction order CountryName, CategoryName print Attractions endfor	<u> </u>	Name	Country	Photo	Category
Hendow and the second second	2	The Great Wall	China		
Bitra AtractionName CountryName CastgoryName	3	Eiffel Tower	France		Monument
	1	Louvre Museum	France	- ¹ → einii	Museum

Así nos queda, por tanto, la estructura del comando for each hasta donde vimos:

	GeneXus
Conceptualization For each BaseTransaction order Att ₁ , Att ₂ ,, Att _n where condition ₁ where condition ₂ where condition _n MainCode	Base table E x t e n d e d
endfor	

El comando admite más cláusulas y opciones. Alguna será vista en otros videos. Otras se abordarán en otros cursos.

Ge	eneXus	
Videos Documentation Certificactions	training.genexus.com wiki.genexus.com training.genexus.com/certifications	

