

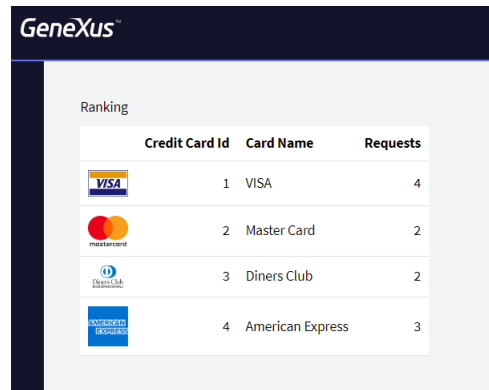
## Carga de un SDT colección en el grid de un Web Panel

GeneXus™





Muchas veces necesitamos mostrar el contenido de una colección de datos, y un web panel es una opción.

Vamos a ver un ejemplo de carga de un SDT colección en el grid de un Web panel, para luego poder realizar alguna acción sobre sus líneas.

Example: Credit cards Ranking

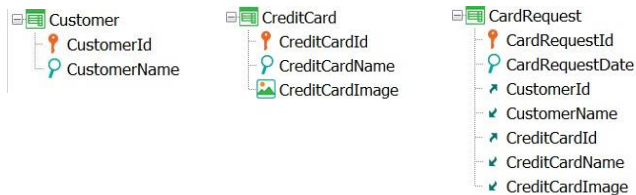


The screenshot shows a web application interface with a dark blue header containing the GeneXus logo. Below the header, the word "Ranking" is displayed. A table with three columns: "Credit Card Id", "Card Name", and "Requests" is shown. The table contains four rows of data, each with a small logo icon to the left of the "Card Name" column.

Credit Card Id	Card Name	Requests
	1 VISA	4
	2 Master Card	2
	3 Diners Club	2
	4 American Express	3

Supongamos entonces que la Agencia de viajes ofrece la posibilidad de registrar las solicitudes de los clientes para la emisión de distintas tarjetas de crédito con las cuales trabaja la agencia. Es así que, queremos implementar un ranking de tarjetas de crédito, según la cantidad de solicitudes registradas en el año 2020. Al tratarse de un ranking, la información deberá presentarse ordenada de mayor a menor por dicha cantidad.

## Example: Credit cards Ranking



Name	Type	Description	Is Collection
SDTCardRequests		SDTCard Requests	<input checked="" type="checkbox"/>
SDTCardRequestsItem			
• CreditCardId	Attribute:CreditCardId	Credit Card Id	<input type="checkbox"/>
• CreditCardImage	Attribute:CreditCardImage		<input type="checkbox"/>
• CreditCardName	Attribute:CreditCardName	Card Name	<input type="checkbox"/>
• CreditCardRequests	Numeric(4,0)	Requests	<input type="checkbox"/>

Consideremos entonces el diseño de transacciones que estamos viendo y el tipo de dato estructurado que permitirá cargar la colección de todas las tarjetas, cada una con su correspondiente cantidad de solicitudes registradas en el año 2020.

## Example: Credit cards Ranking

```
SDTCardRequests from CreditCard
{
  SDTCardRequestsItem
  {
    CreditCardId
    CreditCardImage
    CreditCardName
    CreditCardRequests = count(CardRequestDate,CardRequestDate.Year())=2020)
  }
}
```


Name	Type	Description	Is Collection
SDTCardRequests		SDTCard Requests	<input checked="" type="checkbox"/>
SDTCardRequestsItem			
• CreditCardId	Attribute:CreditCardId	Credit Card Id	<input type="checkbox"/>
• CreditCardImage	Attribute:CreditCardImage		<input type="checkbox"/>
• CreditCardName	Attribute:CreditCardName	Card Name	<input type="checkbox"/>
• CreditCardRequests	Numeric(4,0)	Requests	<input type="checkbox"/>

Vayamos a GeneXus:

¿cómo realizamos la carga del SDT? Con un Data provider

Este Data provider cargará cada tarjeta de crédito con su correspondiente cantidad de solicitudes registradas en el año 2020, de ahí la condición de cálculo declarada en la fórmula.

## Example: Credit cards Ranking

Ranking			
GRID			
	Credit Card Id	Card Name	Requests
	&CardRequests.item(0).CreditCardId	&CardRequests.item(0).CreditCardName	&CardRequests.item(0).CreditCardRequests

```

Event Start
  &CardRequests = DPRequests()
  &CardRequests.Sort("[CreditCardRequests]")
Endevent

```

Name	Type	Is Collection
Variables		
Standard Variables		
CardRequests	SDTCardRequests	<input type="checkbox"/>

Luego, desde el evento Start de un web panel recibimos la salida de este Data provider, y lo hacemos en la variable &CardRequests que ha sido definida basada en el tipo de dato SDTCardRequests.

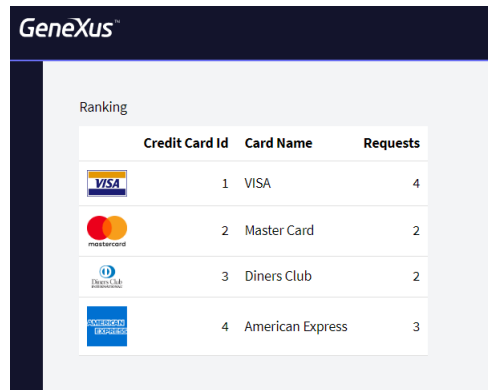
Bien, pero para que sea realmente un ranking, debemos ordenarla de mayor a menor por la cantidad de solicitudes registradas, y para eso utilizamos el método Sort.

Nos está faltando mostrar efectivamente este contenido en un grid.





Para esto, desde el form del web panel, vamos a insertar la variable &CardRequests. Como se trata de una variable estructurada y colección, automáticamente GeneXus le asociará un grid. Pero es una variable, y ya sabemos que las variables en un web panel son datos de entrada. Por lo tanto, para que no se pueda modificar su contenido, vamos a indicar que es Read only.

Observemos que también podemos elegir si queremos que su contenido se muestre en un Free style grid. En caso de no marcar esta opción, se mostrará en un grid común.

En ejecución:



The screenshot shows a web application interface with a dark blue header containing the GeneXus logo. Below the header, the word "Ranking" is displayed. A table with three columns: "Credit Card Id", "Card Name", and "Requests" is shown. The table contains four rows of data, each with a credit card logo in the first column, a rank and card name in the second, and a number of requests in the third.

Credit Card Id	Card Name	Requests
	1 VISA	4
	2 Master Card	2
	3 Diners Club	2
	4 American Express	3

Pensemos ahora lo siguiente:

¿Hay alguna diferencia entre este grid cargado a partir de una variable estructurada colección y otro grid cargado con variables de la manera habitual? No, en cualquier caso se trata de un grid sin tabla base.

Vamos a verlo en ejecución. Presionemos F5:

Bien, ya tenemos implementado nuestro ranking. Ahora queremos resolver lo siguiente:

Necesitamos poder seleccionar una de estas tarjetas de crédito, y que se despliegue un listado pdf con las solicitudes registradas para poder obtenerla.

Volvamos a GeneXus.

Ya hemos definido el listado CardRequestList

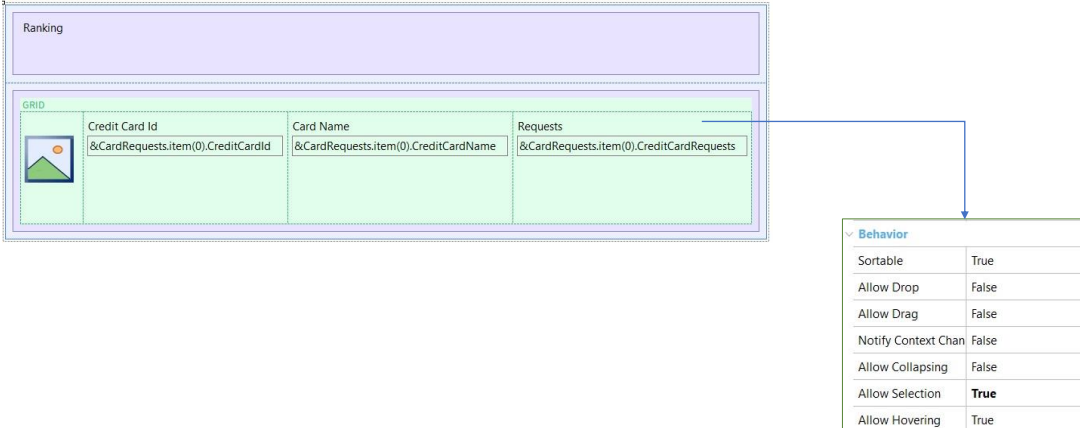
```
Parm(in:CreditCardId, in:&CreditCardName);  
output_file("List", "pdf");
```

```
printblock1  
.....  
.....&CreditCardName.....  
.....Card requests.....  
.....  
printblock2  
.....CardRe CustomerName.....  
.....
```

```
1 print printblock1  
2  
3 For each CardRequest  
4   print printblock2  
5 endfor
```

Para facilitar, ya hemos definido el procedimiento CardRequestList que recibe en su regla Parm el identificador y el nombre de la tarjeta de crédito, y muestra las solicitudes registradas.

## Propiedades Allow Selection y Allow Hovering



The screenshot shows a web panel with a 'Ranking' header and a 'GRID' below it. The grid has three columns: 'Credit Card Id', 'Card Name', and 'Requests'. The data rows are: '&CardRequests.item(0).CreditCardId', '&CardRequests.item(0).CreditCardName', and '&CardRequests.item(0).CreditCardRequests'. A blue arrow points from the grid to a 'Behavior' table.

Behavior	
Sortable	True
Allow Drop	False
Allow Drag	False
Notify Context Chan	False
Allow Collapsing	False
Allow Selection	<b>True</b>
Allow Hovering	True

Bien, volvamos entonces a nuestro web panel.

Necesitamos poder seleccionar una línea del grid. Así que vamos a ver las propiedades del grid, y vamos a habilitar la propiedad Allow Selection con el valor True.

De esta forma se habilita también la propiedad Allow Hovering que permite ir resaltando con color a medida que pasamos por las líneas. Vale mencionar que los colores por defecto pueden modificarse a través del Theme aplicado, editando las clases asignadas a las propiedades Selected Row Class y Hover Row Class a nivel de la clase del Grid;

Bien. Pero también necesitamos un evento que capture la línea seleccionada y permita realizar una acción. Ese evento es el evento OnlineActivate asociado al grid.

Vamos entonces al sector de los eventos, y escribimos Event, y el nombre del grid, que en este caso es Grid1, punto, y seleccionamos el evento onlineActivate

Lo que queremos hacer es llamar al listado que hemos visto anteriormente, y pasarle por parámetro el Id y el Nombre de la tarjeta



seleccionada en la línea.

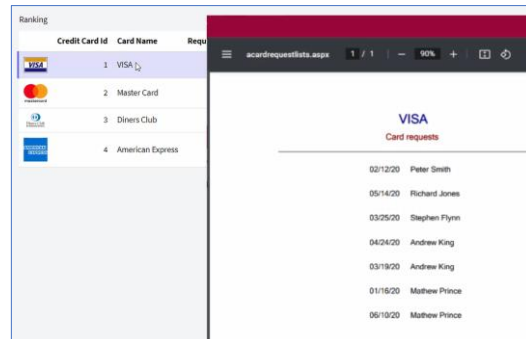
## Propiedad CurrentItem

```

Event Start
  &CardRequests = DPRequests()
  &CardRequests.Sort("[CreditCardRequests]")
EndEvent

Event Grid1.OnLineActivate
  &CreditCardId = &CardRequests.CurrentItem.CreditCardId
  &CreditCardName = &CardRequests.CurrentItem.CreditCardName
  CardRequestLists.Popup(&CreditCardId, &CreditCardName)
EndEvent

```



¿Cómo lo hacemos?

Como el grid muestra la carga de una colección, debemos posicionarnos en el ítem seleccionado de la colección para poder obtener entonces el id y el nombre. Para eso, vamos a utilizar la propiedad CurrentItem.

Así que vamos a definir la variable &CreditCardId... **vemos que quedó** basada automáticamente en el atributo de igual nombre, y le vamos a asignar a la variable colección &CardRequests punto, CurrentItem, punto, CreditCardId, para obtener de esta forma el Id de la tarjeta seleccionada en la línea.

Vamos a hacer lo mismo para obtener el Nombre.

Definimos la variable &CreditCardName, verificamos que quedó basada en el atributo de igual nombre, y definimos la asignación correspondiente.

Finalmente llamaremos al listado definido para que se abra como una ventana popup, y le pasamos los parámetros necesarios: la variable &CreditCardId, y la variable &CreditCardName.

Vamos a verlo en ejecución. Presionemos F5:

## Collection SDT in the grid of a Panel object

- The collection is received in the Refresh event.
- The variable is inserted in the form and GeneXus will automatically create the grid with the selected fields.
- To perform any action on a selected line, we must define an event and use the CurrentItem property
- This event should be defined as the default action of the grid (Default Action property).

Para finalizar, vale mencionar que si se estuviera trabajando con un objeto de tipo Panel, el proceso es similar:

En el evento Refresh se recibe la colección en una variable.

Se inserta esta variable en el form y G creará automáticamente el grid asociado con los campos seleccionados.

En caso de querer también realizar alguna acción sobre una línea seleccionada, se deberá definir un evento y utilizar la propiedad CurrentItem

Finalmente se deberá definir dicho evento como la acción por defecto del grid, utilizando para esto la propiedad Default Action.

*GeneXus*<sup>™</sup>

[training.genexus.com](http://training.genexus.com)  
[wiki.genexus.com](http://wiki.genexus.com)