

*GeneXus*<sup>™</sup>  
**The power of doing**

Development

*GeneXus*™ 16

Eu sou Federico Salomón, estaremos vendo com Martín, ambos somos parte da equipe técnica de GeneXus. Veremos pontualmente a parte de desenvolvimento da versão 16.

S  
O  
L  
U  
T  
I  
O  
N

Multi-experience & Omnichannel

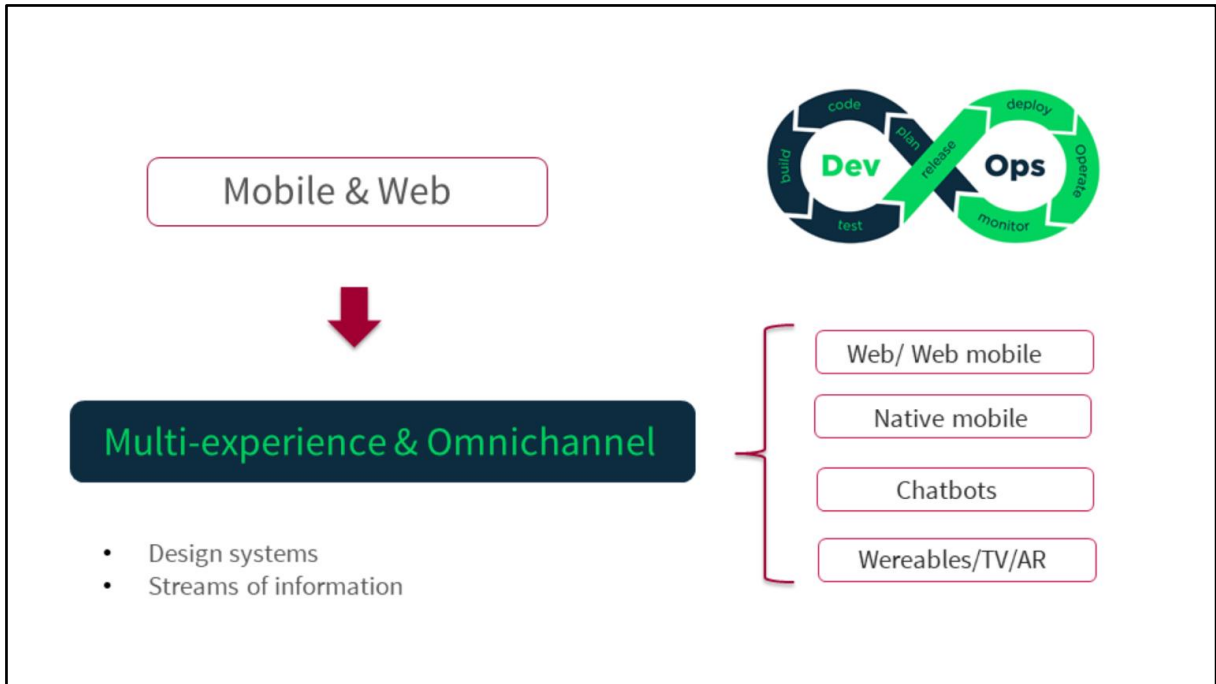
Integration

Security

P  
L  
A  
T  
F  
O  
R  
M

GeneXus IDE / Modeling /  
Language





Começando pelo Design System que é um conceito que vem sendo muito trabalhado durante o GX 28

# Design Systems

Com tracks específicos sobre o conceito de Design System.  
E falamos de Design System como algo muito mais avançado, como uma aposta muito mais alta que apenas o fato de aplicar desenhos às aplicações.  
Design System aumenta muito mais a cerca se se deseja.

Principles

Patterns

Practices

E fala de princípios, de padrões e de práticas com base no desenho, como tem que se comportar a tela, como devem ser as interações de nosso software com os usuários. Fala de uma série de padrões que se deve respeitar.

É um conceito muito mais amplo que só aplicar desenho e que as coisas sejam bonitas.

## Design Systems



Em Design System, o que estamos vendo aí são grandes organizações, grandes empresas que começaram com esta aposta e começaram a implementar seu Design System para ter uma consistência nos produtos que oferecem.

Para que todos os seus produtos tenham a mesma aparência, tenham as mesmas interações com os usuários, independentemente da plataforma que executam.

Vemos grandes empresas, como IBM, AIRBNB, Mozilla, GitHub, General Electric. Várias empresas que começaram com esta ideia de unificar seus desenhos, de unificar as interações de seu produto com os usuários.

Por isso, que dizemos e cremos que Design System é muito importante e que temos que começar a incorporá-lo dentro do conhecimento do GeneXus, de nossa base de conhecimento.

## Integrate the designers (Design Ops)



O Design System fala também de integrar os Designers.

Como Armin falava antes, um conceito que se está falando muito também é o tema das DEVOPS (Development Operations).

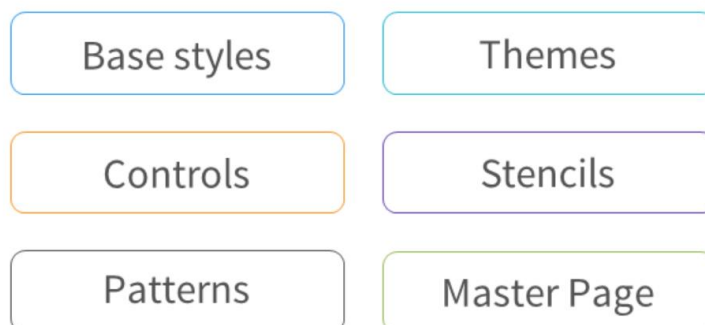
Design Systems fala também de Design OPS.

Integrar os Designers dentro desta cadeia de DEVOPS e que são parte do trabalho e que trabalhem lado a lado com os desenvolvedores para que os resultados sejam realmente os esperados e a que se propõe.

O que vemos abaixo são ferramentas (as principais) que usam os Designers para fazer seu trabalho e para levar adiante toda este esquema de integração.



## Design system



Em Design System o que conseguimos em GeneXus é uma série de estruturas que algumas já são conhecidas como comentou Armin.

Os Temas, os Patterns, as Master Pages, são estruturas já conhecidas pelos usuários GeneXus, pela comunidade GeneXus, que já estamos acostumados a utilizar em nossos projetos, são parte importante deste modelo UI e UX e que devemos ter em nossas aplicações, mas na versão 16 incorporamos novos componentes que fazem parte deste conjunto de estruturas, que necessitam trabalhar juntas para conseguir o resultado.

Falamos de Base Styles, falamos de User Controls e falamos dos Stencils.



Também, outro conceito que está sobre a mesa hoje e se ouve muito são os CSS Frameworks. CSS Frameworks são os que vão nos dar esta Interface, esse desenho bonito que vemos nas aplicações. Se seguir são parte do Design System e são os que definem aquilo que devemos ver nas aplicações, certo?

O Design System está muito relacionado com os User Controls e com os Base Styles.

Eu, particularmente, trabalho muito com Semantic UI, gosto muito, e também vemos aqui que aparece Bootstrap que é muito conhecido por nós também, já que Bootstrap é o motor que nos provê, o comportamento Responsivo das aplicações. O motor do Bootstrap é quem decide como deve ser visualizada a aplicação quando alteramos o tamanho da tela.

Bootstrap já é conhecido para GeneXus, mas temos hoje a possibilidade de incorporar muitos mais e Bootstrap está, de certa forma, desacoplado das soluções.

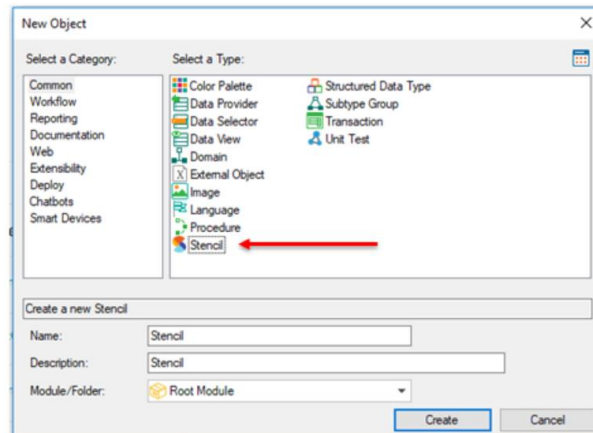
Agora não decide tanto como serão visualizadas as coisas, mas se encarregam de realizar o comportamento Responsivo e vamos poder utilizar a interface, a UI que propõe outros CSS Frameworks. Com isso, quero dizer que agora não vamos ver o desenho que propõe Bootstrap em certos controles que fornecemos, mas vamos poder definir muitos, existem muitos CSS Frameworks.

Agora deixo com Martín que vai falar a vocês sobre Stencils para depois seguir com os outros controles que comentei.



Começo destacando um pouco desse objeto novo em GeneXus 16, que na realidade já está disponível em GeneXus 15 U12, que são os Stencils.

# What is a Stencil?

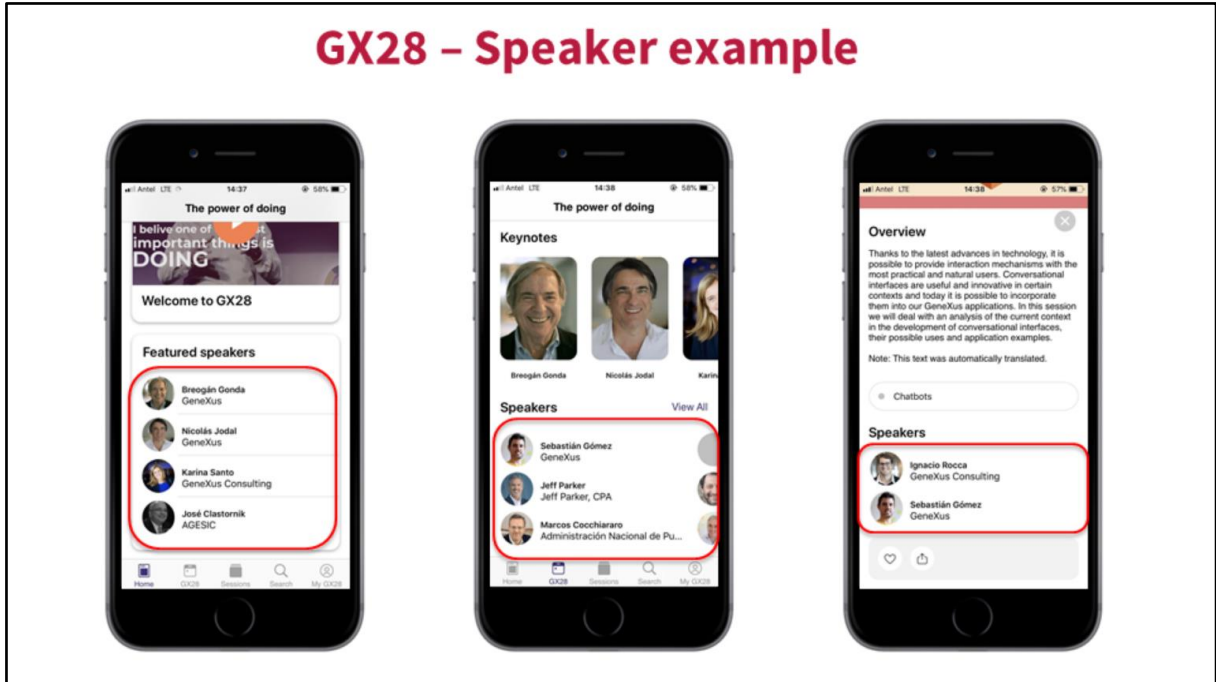


Então, comecemos por “O que é um Stencil ?”

Stencil é, como eu dizia um objeto novo que nos ajuda a criar de forma mais fácil este Design Systems que falava Frederico. São objetos de desenho que nos ajudam a criar a interface de usuário separado do código e favorece muitíssimo tudo o que seja reuso e tudo o que é modificabilidade.

Então como podem ver, é simplesmente Ctrl + New e temos o objeto novo aí, para criá-lo a partir zero.

## GX28 – Speaker example



Em que casos utilizaríamos um Stencil? Pareceu-me que um exemplo bem claro é a aplicação do evento, na qual nós construímos utilizando este novo objeto e como podem ver estão aqui 3 telas.

A primeira do lado esquerdo mostra os palestrantes, os palestrantes favoritos digamos. A segunda mostra uma lista completa de palestrantes e a terceira é o detalhe de uma conferência que mostra quais são os palestrantes responsáveis por esta conferência.

O que tem em comum nestas telas e em algumas outras que não coloquei por falta de espaço? É que nós mantemos uma linguagem visual de como mostramos os palestrantes.

Então, os mostramos com uma foto arredondada, mostramos com o nome do palestrante em negrito, com determinada fonte, com determinada margem que separa o nome da Empresa na qual trabalha.

Certas coisas que mantemos no desenvolvimento da aplicação e que sempre são iguais. Então, como as fizemos até hoje?

A fazemos uma vez em algum dos painéis e depois a copiamos e levamos a outras telas.

Isto traz um problema, que é, o que acontece se amanhã quero mudar a forma na qual são exibidos os palestrantes? Quero que a foto seja quadrada e não redonda.

Então teria que ir painel por painel e modificar esta imagem.

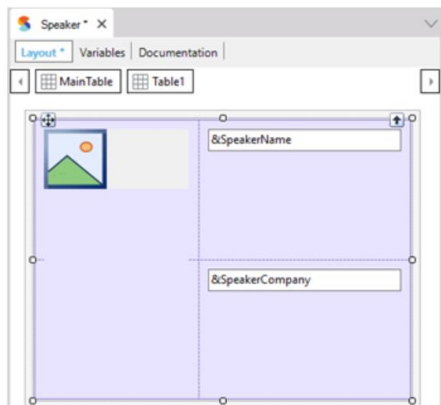
O poder que há nos Stencils é... eu crio um Stencil com este desenho, o aplico em todas as telas que eu queira.

E esta tela que, digamos, como que agregando este Stencil, eu modifico o Stencil e se modificam todas as telas.

Então a modificabilidade é muito beneficiada nestes casos. Porque em outro caso se eu modifico tela por tela, posso esquecer a tela, posso não criar exatamente igual porque pus uma margem um pouco maior, então o Stencil garante que será sempre igual... a linguagem visual vamos manter. Encapsulamos este conhecimento de desenho, digamos, em um Stencil. É muito mais fácil também compartilhá-lo entre diferentes objetos de minha Kb, entre outras Kbs, entre a empresa e com a comunidade, porque como lhes dizia isto está separado do código. Então é muito mais fácil de compartilhar.

## Stencil Object

### Controls + design properties



Sebastián Gómez  
GeneXus

**Pergunta:** Isto vem a ser o equivalente a um Web Component em web?

Sim, a diferença especificamente... uma das grandes diferenças foi comentada por último... está completamente separado do código.

Eu ia comentar, por exemplo, que aqui temos um Stencil Object que é um layout igual a um SD Panel ou Web Panel que estamos acostumados a trabalhar, com a diferença é que é mais fácil. Por quê?... Porque tem somente o que é relacionado ao desenho, este não tem código, apenas propriedades de desenho. Então aqui definimos o layout de como queremos que se visualize um palestrante em execução, e vai ficar desta forma.

Depois como o carregamos, como carregamos a variável, como se carregam os dados, tudo além disso, referente a código fica separado. Então essa é uma das grandes diferenças. E é muito mais fácil, por essa qualidade, muito mais fácil de compartilhar.

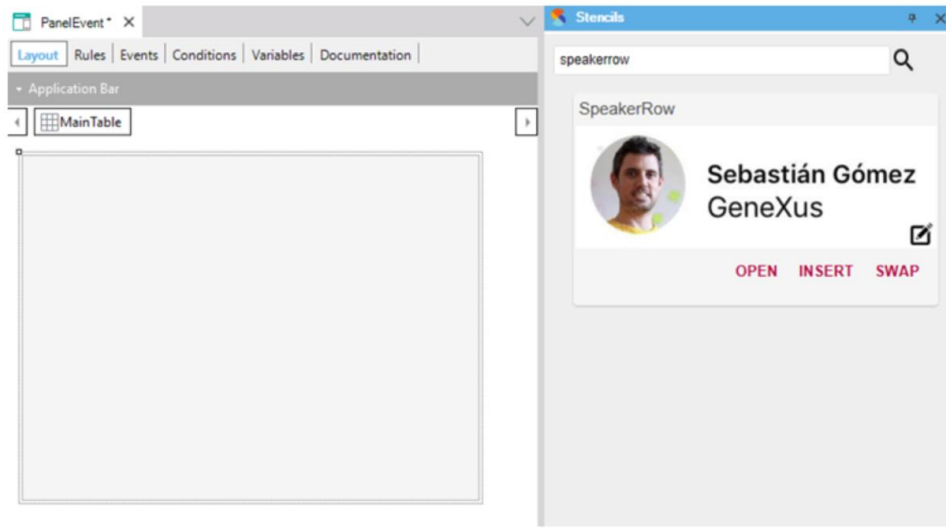
**Pergunta:** É válido tanto para Web quanto para Smart Devices?

É válido tanto para Web quanto para Smart Devices.

Muitas vezes acontece de estar com um colega que diz "poderia me passar este painel que tem em sua aplicação?" e você tem que passar um xpz com código, com tudo, então isso facilita muito mais o tema de compartilhar.

Então, aqui temos o objeto Stencil, nós criamos o desenho como queremos que se visualize. Será visto desta maneira...

## Inserting an Stencil on my layout



E em seguida para inseri-lo no meu layout, o que temos que fazer é ir para meu SD Panel ou Web Panel, o que seja, e buscar na toolbox de Stencils, buscar o Stencil que eu quero inserir e simplesmente com o botão de Insert o agrego ao layout e aí sim, posso começar a codificar como lido com esse comportamento.

**Pergunta:** Este Stencil pode ser criado a partir de um desenho Sketch?

Sim, mais adiante vamos falar.

Então esta seria a toolbox para inserir Stencils. Como podem ver contamos com a tela para que seja mais fácil identificar o que estamos agregando ao nosso layout. Com o botão Insert, inserimos.

Inserimos neste painel como inserimos em "N" painéis que queremos usar e, como lhes dizia, depois modifico simplesmente o Stencil e esta mudança impacta em todos.

## Stencil versions

V.1



Sebastián Gómez  
GeneXus

V.2

Sebastián Gómez  
GeneXus



V.3



Sebastián Gómez  
GeneXus

V.n...

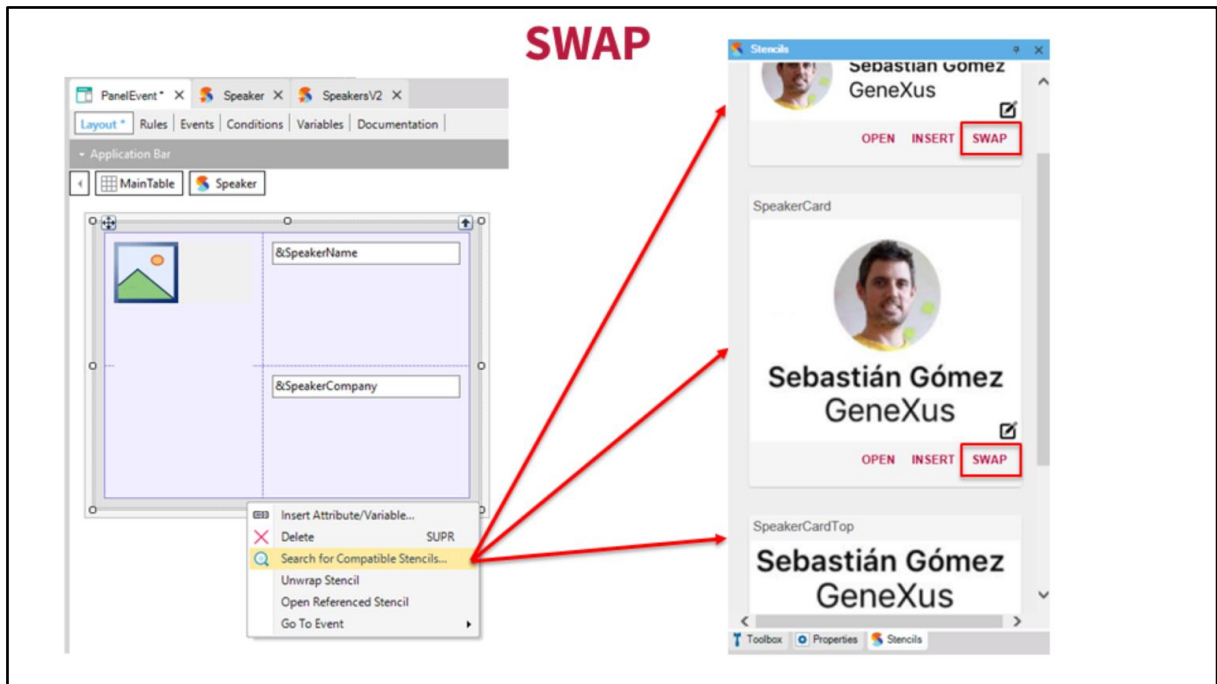


Lorem Ipsum

Os Stencils também nos ajudam no tema de versionamento. O que acontece se estamos trabalhando neste desenho e nos dá novas ideias, ou o cliente talvez diz, “eu gostaria mais que os palestrantes sejam desta outra forma”. Então podemos criar diferentes versões de como mostrar esta linguagem visual que eu lhes dizia, de como mostrar um palestrante. Aí temos o da direita, acima, abaixo e “N” maneiras de como podemos mostrar um palestrante.

Então com os Stencils podemos criar diferentes Stencils com cada uma destas representações e para fazer a troca em nosso painel é bem fácil...





Simplesmente temos que voltar ao painel que estávamos trabalhando antes. Já temos o Stencil inserido, clicamos com o botão direito sobre ele... Search por Stencils compatíveis... e irá mostrar novamente a toolbox com os Stencils compatíveis com este esquema... e aqui vemos todos os que mostramos anteriormente, todos estes aqui...

Então a única coisa que faço é selecionar o que quero e pressionar sobre o botão SWAP. Isto é o que faz a troca do Stencil.

**Pergunta:** Quando faço uma atualização tenho que ir painel por painel para atualizá-lo? Como?

**Pergunta:** Aí mostra três versões diferentes para o mesmo Stencil... a pergunta é, se vou querer que aplique... tem para todos aplicado sobre os painéis o Stencil 1, quando quero passar todos para a versão 3, tendo que ir painel por painel para atualizá-lo?

Sim, terá que fazer o SWAP porque, digamos, este...

**Pergunta:** Não há uma forma que se atualizem todos?

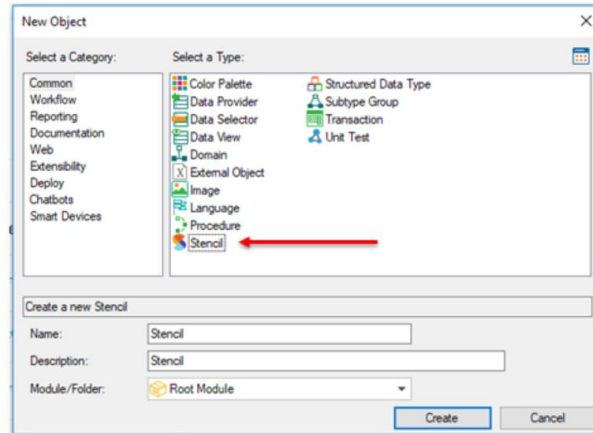
Se estamos usando o 1... estamos utilizando o 1 em todos, se queremos trocar pelo 3 temos que ir um por um, mas a feature de trocar todos da versão 1 pela 3, eu adoraria... não se ainda quando vamos poder fazer.

## How do I create a new Stencil?

1. Create new stencil object
2. Wrap as stencil an existing design
3. Import from Sketch 

Então seguimos... a forma de criar Stencils. Agora com o exemplo do evento. Estamos vendo o ponto 1. Se criar um Stencil a partir do zero.

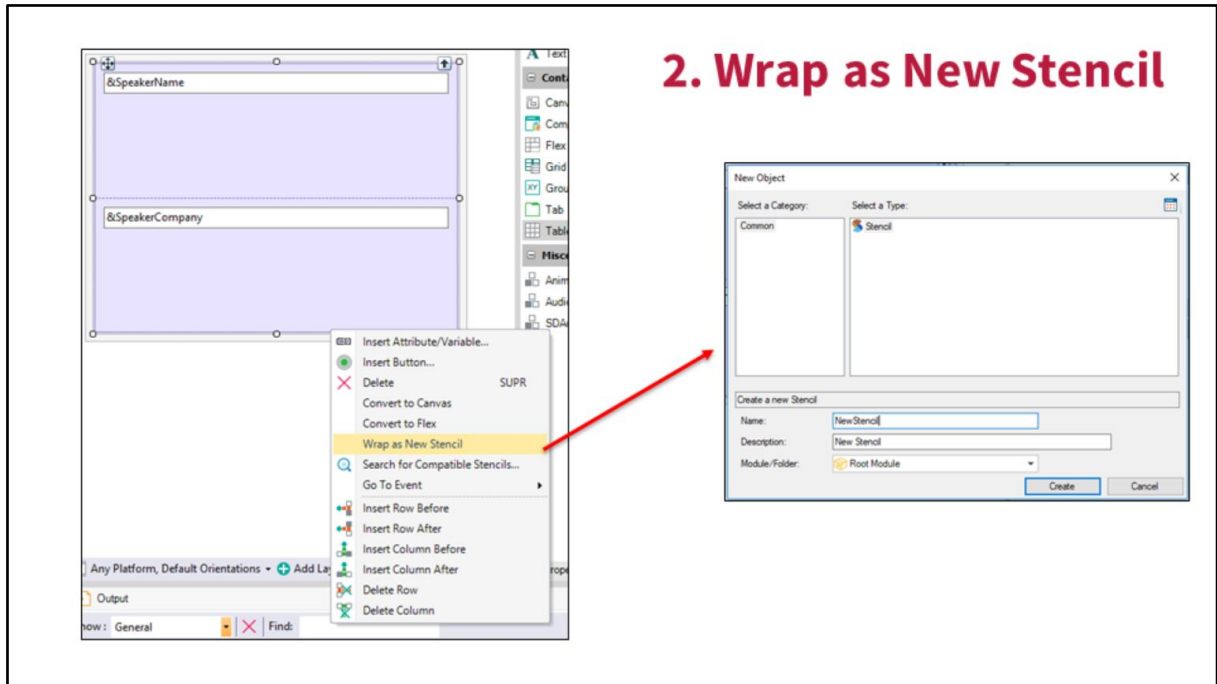
# 1. Create new Stencil



Ctrl + New e cria um Stencil vazio. E a partir daí começo a criar um desenho com um layout vazio.

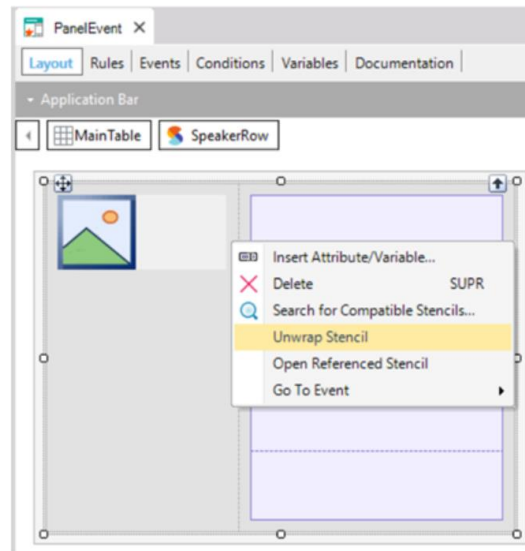
Há uma outra forma de criar um Stencil, por exemplo, com uma base de conhecimento que já temos, que já temos um desenho aplicado que queremos encapsular este desenho e poder reutilizá-lo.

## 2. Wrap as New Stencil



Então, o que teríamos que fazer é ir à seção do layout que queremos encapsular, com o botão direito clicamos em “Empacotar como Stencil” e automaticamente nos abre esta caixa de diálogo para personalizar o nome e salvá-lo. Isto o que faz é desacoplar esse desenho do código e se salvá-lo como Stencil para que possamos reutilizá-lo em qualquer tela como eu comentava antes.

# Unwrap Stencil



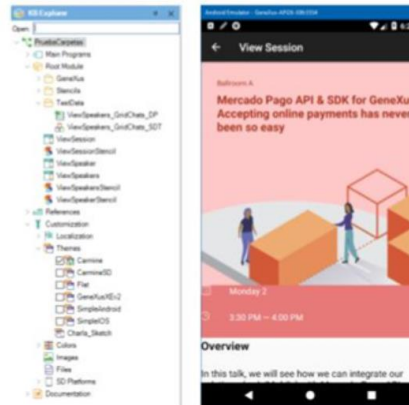
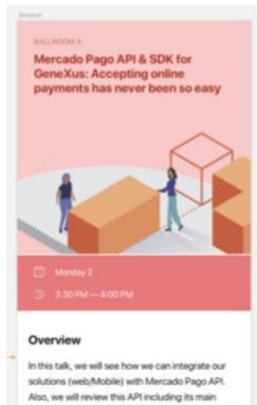
O mesmo para o outro lado. O que acontece se estava trabalhando com um Stencil e quero cortar o vínculo do Stencil com meu layout...

O que temos que fazer é simplesmente ir ao Stencil aplicado e fazer um “Desempacotar Stencil”. O que acontece aí automaticamente é... já perde essa a conexão com o Stencil, digamos, se modifica o Stencil já não vai modificar essa tela, mas vai manter as propriedades de desenho que já tenho aplicadas. Agora, vou trabalhar livremente, totalmente desacoplado do Stencil.

### 3. Import from Sketch (Coming soon)



GeneXus<sup>®</sup> 16  
The power of doing

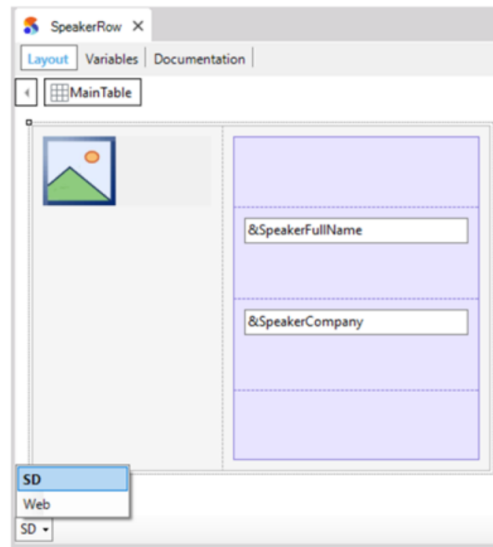


E um pouco do que você perguntou. A terceira forma que em breve haverá para criar Stencils é a partir das extensões que nos passam os designers, por exemplo, Sketch.

Aqui à esquerda podem ver o arquivo Sketch com suas telas e desenho, também com suas navegações.

E o que faz GeneXus é um Import desse arquivo. Então o que GeneXus fará é... vai criar os Web Panels, vai criar os Stencils e vai criar as navegações entre estes Web Panels. Então, você já pode executar, digamos, com o desenho incorporado. O que restaria é codificar o comportamento, mas o desenho estaria trazendo e criando os Stencils.

## SD & WEB



Também se perguntou se isto é para SD e para Web.

Os Stencils têm layouts. Porque geralmente se vemos a apresentação de uma página Web e uma Aplicação SD não é a mesma. O conceito por trás do que se quer mostrar, sim, é o mesmo. Os palestrantes eu iria querer mostrar de uma forma Web e outra forma em SD, mas o conceito é o mesmo, então temos um layout para manter esta hierarquia dos Stencils... temos um layout SD e um layout Web que podemos modificar separadamente.



**+ Modifiability**  
**+ Reuse**  
**+ Flexibility**

*“Stencils help to define standards and consistency,  
simplifying the implementation of Design Systems.”*

Então em resumo, os Stencils definem Standards e nos dão consistência, nos ajudam muito a criar Design Systems. Aumenta muito a modificabilidade, por isso que falamos e reuso sem dúvidas, facilita muito compartilhar desenho e nos dão uma flexibilidade enorme, ou seja, os convido a testá-lo. E agora se falará um pouco sobre User Controls.





Bem sobre User Control Object, para o que tenho experiência desenvolvendo User Controls já é um conceito bastante conhecido e já o trazemos em várias versões anteriores.

O User Control Object se queremos uma nova forma de modelar seus controles que não suplanta a forma anterior, mas o User Control como conhecemos vai continuar existindo, mas o User Control Object apresenta uma série de vantagens que facilita essa integração com código nativo, com a modelagem de UI e UX. Não sei se tiveram a oportunidade de ir na apresentação sobre o tema, os convido a vê-la que tem muitos detalhes sobre o objeto, mas basicamente o que permite isto é facilitar a forma de modelar o User Control. Tem uma série de vantagens que, por exemplo, será um objeto da KB GeneXus (portanto é Built-in) e resolve muito o jeito da modelagem de controles, sua manutenção, temos para os User Controls anteriores, temos o User Control Editor como parte da suíte GeneXus, que permitia modelar o controle mas era uma ferramenta satélite, o controle não era parte desse conhecimento Genexus. Tinha que modelá-lo fora, incorporá-lo na KB depois instalando-o em nossa instância do GeneXus. Isso era um trabalho bastante tedioso, tinha que ser muito detalhista porque podia falhar.

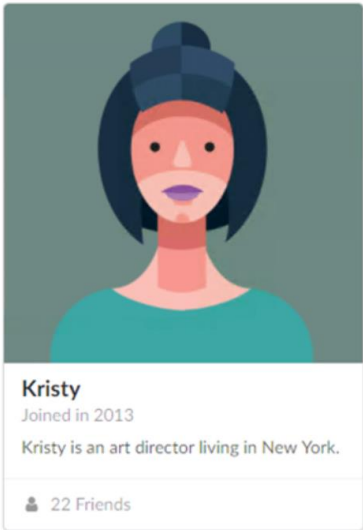
Então, o fato de ser Built-in já facilita muito as coisas. Em segundo lugar a portabilidade aumenta muitíssimo, é um objeto a mais da KB e portanto se estamos trabalhando com um esquema SSM com GeneXus Server vai ser muito mais fácil também.

No caso anterior de User Controls se estamos trabalhando com um User Control na KB, na instância local do GeneXus e fazemos um Commit ao Server e este Server não tinha determinado controle nos daria uma mensagem do tipo "não conheço esta estrutura, por favor revisar o que ocorre", instalá-lo nesse caso e depois podemos continuar trabalhando. Vamos esquecer este problema, já que é um xpz a mais o esquema, digamos, de consolidar em Genexus Server, funciona da mesma maneira que com qualquer outro objeto, portanto a portabilidade é muito melhor e a distribuição também, é apenas um xpz, fazemos um export e podemos distribuí-lo. Não temos que zipar, não temos que manter versão por fora, é realmente muito mais fácil, é muito mais potente, temos histórico, temos controle de versão, temos Cross Reference para saber quem faz uso do controle e a quem referencia esse controle. Portanto, é muito mais simples de usar.

Por outro lado também pode utilizar o Base Style, o Base Style que comentamos antes e que vai dar essa apresentação que o User Control precisa. Baseando-nos em CSS Frameworks que vamos incorporar à KB.

Por outro lado este User Control apresenta a particularidade que gera Server Side, o Html que

nos servirá o controle será gerado pelo Server, portanto, delegamos a responsabilidade ao servidor de apresentar esse controle de forma otimizada e rápida. É muito mais fácil escalar o servidor que escalar o browser do cliente, portanto lhe delegamos a responsabilidade de usar o Html e esse uso se traduz em load performance, ou seja, será muito mais rápido, muito mais otimizado e isto para um objeto que tem uma interface muito rica, que fazem muito uso de controles realmente faz a diferença.



Controls

Base styles

```

<div class="ui card">
  <div class="image">
    
  <div class="content">
    <a class="header">Kristy</a>
    <div class="meta"> <span class="date">Joined in 2013</span> </div>
    <div class="description"> Kristy is an art director living in New York. </div>
  </div>
  <div class="extra content">
    <a <i class="user icon"></i> 22 Friends </a>
  </div>
</div>

```

Aqui o que vemos é um controle Card do Semantic UI, que faz basicamente mostrar informações de usuários, de clientes ou o que seja, de uma forma muito mais bonita do que estamos acostumados a ver.

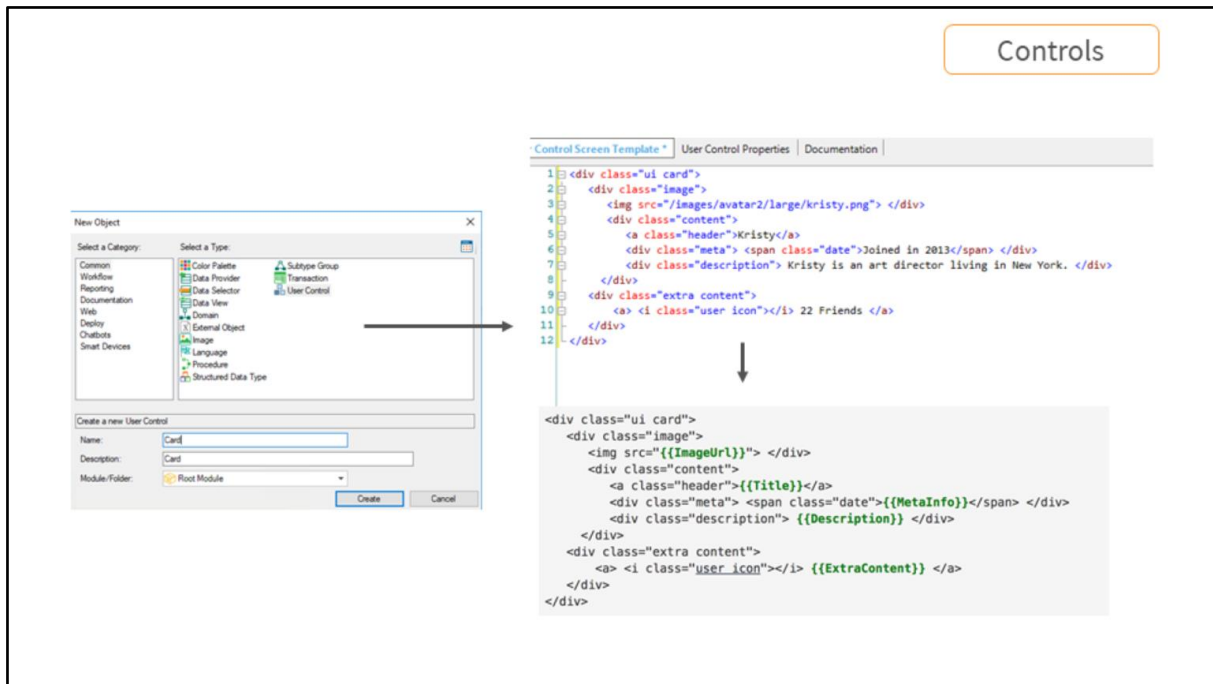
Mostra a foto da pessoa, seu nome, e uma data, neste caso de quando entrou na rede social, por exemplo e uma breve view da pessoa. Isto que vemos aqui se modela com estas 10 linhas de código + ou - Html que nos fornece este CSS Framework de Semantic UI.

Então vamos à página de Semantic UI, vamos pesquisar o Control Card e o que vamos fazer é simplesmente copiar este código Html, vamos colar dentro da estrutura de nosso User Control, agora irá constar como a estrutura e automaticamente ao utilizar este User Control num Web Panel ou em um objeto que tenha interface gráfica... o veremos desta mesma forma e o que vemos aqui é que este controle tem classes definidas que são próprias do Semantic UI e estas classes estarão disponíveis dentro da Kb GeneXus e quando utilizamos os Base Styles, como vemos aqui em cima o Base Style está diretamente relacionado com o User Control.

O Base Style vamos utilizar tanto em User Controls como no objeto Tema. Adicionamos uma nova propriedade que permite referência a esse CSS Framework dentro da Kb.

A forma que temos de referenciá-los é utilizando o file da KB que com certeza já o conhece também e o que vamos fazer é um zip desse set de arquivos que nos fornece esse CSS Framework ou nossos designers, porque o resultado final que nos gera nossos designers é basicamente o mesmo que tem no CSS Framework. É uma série de CSSs, te alguns Assets e tem outras estruturas estáticas que são as que vamos utilizar. Então o que fazemos aqui é... lhe colocamos uma extensão que é gxllibrary a extensão.

Vejo uns rostos preocupados. Veremos na prática depois para que tenham mais detalhes de como funciona, é verão que realmente é muito fácil. Mas, vamos colocar uma extensão que é .gxllibrary e é neste momento que ao utilizá-lo como file na KB, GeneXus se dará conta que tem que fornecer estes estilos para poder utilizar mediante uma propriedade em nosso controles.



O que vemos aqui é a criação do objeto User Control.

**Pergunta:** Olha, minha consulta é... nós temos referências a classes, estão em uma library, estará empacotado na nuvem, não sei. O tema é... aumenta demais de tamanho a KB...

Porque imagine, incorpora os controles. Vou rodar de forma local, desconectada. Essas classes são incorporadas à KB?

Correto. Estão e... GX16 trabalha por referência de nomes e quando você define este HTML e a classe que está sendo utilizada aí, a tenha definida dentro do CSS Framework que esteja utilizando, nesse caso SemanticUI, automaticamente já vai tomar esta classe.

**Pergunta:** Vai tomar esta classe?

Vai tomar esta classe. Mas, você está importando todos este set de classes do SemanticUI, está importando o CSS completo. Não deveria ser muito pesado este arquivo estático, mas sim, pode aumentar alguns Kbs... de kilobytes o tamanho.

**Pergunta:** Isso é só para Web?

Sim

**Pergunta:** Poderá mais adiante ser para SD?

O que acontece é que temos outras estruturas que podem modelar uma série de coisas em SD, digamos, e na realidade a necessidade inicial foi em Web. Não apenas pela modelagem de UI, mas também porque tem muito poder o script nisto, como havia comentado, e permite muita integração com muitas APIs REST, digamos, que em Web são muito utilizadas.

Por isso a primeira intenção é desenvolver Web e, bem, depois veremos. Não imagino agora um caso muito específico em SD do User Control... podemos avaliar em primeira instância para Web para dar poder à modelagem, potência ao Script para integração de APIs. Vamos ver como amadurecem e as necessidades que surjam.

O que eu comentava aqui é como se faz a criação do User Control. É um novo objeto da KB, vamos fazer Ctrl + N. O diálogo de novo objeto, definiremos um nome e o que vemos aqui à direita é a estrutura que eu comentava, a anatomia deste novo objeto, é basicamente dividida em duas partes, o Screen Template e as Properties.

O Screen Template é o que irá permitir a modelagem nativa. Nós dizemos modelagem nativa de UI e UX em GeneXus. E esta seção do User Control é justamente a que nos permite isto. Se fixarem aqui, este código HTML é exatamente o mesmo que estávamos utilizando no Card de SemanticUI dentro do GeneXus. Já está incorporado na base de conhecimento. E aqui o que vemos é que tem dados que são estáticos. Vemos um caminho de uma imagem, um nome em hardcode na tag "A", vemos os dados que comentei antes.

E o que vemos abaixo é como o User Control permite o dinamismo de seus dados, porque o que queremos é justamente usar dados dos atributos de nossas transações, de variáveis que tenhamos definidas na KB, etc .

Esta sintaxe aqui se chama mustache pela chave que é similar a um bigode, por isso se chama mustache, é a que nos permitirá esse dinamismo nos dados do controle. Ao definirmos essa sintaxe, automaticamente estamos dizendo ao GeneXus que pode ser um dado dinâmico que inicializaremos depois num evento GeneXus qualquer, como por exemplo o load de um Web Panel eu vou dizer Control Card... ponto... propriedade imageUrl... a URL da imagem.

**SemanticUI**

- BSBreadcrumb
- BSPanel
- ButtonLike
- Card



```
Event Start
  card1.Title = "Facundo"
  card1.MetaInfo = "Nacio en 2011"
  card1.Description = "Minecraft fan"
  card1.ExtraContent = "No tiene Facebook"
  card1.ImageUrl = "https://genexususa.com/wp-content/uploads/2017/08/Gaston-Milano-GeneXus-CT0-cropped.jpg"
EndEvent
```

Aqui o que vemos são uma série de controles dentro da Toolbox. Ao criar o User Control automaticamente o GeneXus a coloca como parte da Toolbox de Controles Built-in. E aqui temos uma série de controles .

Aqui temos a inicialização que comentei antes. No evento Start inicializa... o título que queremos que se exiba para a pessoa, os dados que vimos nos exemplo anterior. E isto é como é gerado o resultado em execução.



**+ Portability**  
**+ Load Performance**  
**+ Easy to extend & share**

O que podemos concluir deste objeto User Control?

Em primeiro lugar a portabilidade que oferece pelo que comentei antes, pelo tema da facilidade de trabalhar com GeneXus Server, porque é Built-in, porque é muito fácil de distribuir e de manter, porque é uma estrutura a mais do GeneXus.

Uma melhor performance na carga porque as aplicações não têm apenas que parecer boa, mas têm que responder rapidamente, porque o usuário a cada segundo está decidindo se quer continuar navegando ou não.

E, em terceiro lugar, muito fácil de estender e compartilhar, como o objeto Stencil.

**Pergunta:** E poderia ter nossos controles dentro de um Grid, por exemplo?

Poderia

**Pergunta:** Poderíamos ter vários controles?

Sim. E terá... dentro de um objeto GeneXus e dentro de um mesmo User Control usando um SDT. As propriedades que comentei dos User Control são as que estamos acostumados em GeneXus. Pode definir uma propriedade como SDT, pode definir a propriedade como numérico, como String. Pode incluir o controle dentro de um Grid ou pode ter dentro do mesmo controle usando um SDT. Consegue-se ter um ciclo, digamos, aí, interativo.

**Pergunta:** Uma pergunta nossa... Pode-se anexar arquivos Javascript a estes controles?

Sim, pode. Isso é um overview básico digamos do que é o objeto. Mas sim, tem muito poder de scripting. Dentro da seção Properties que comentei o que GeneXus faz é... inicializar a propriedade que você define dentro do template, coloca a entrada, digamos, no arquivo... define o tipo e por sua vez, dentro dessa seção você pode definir scripts.

Utiliza a tag script como está acostumado e aí é onde você define, digamos seu Javascript para poder integrar com algumas APIs.

**Pergunta:** Será possível utilizar os User Controls anteriores, que eram os arquivos compartilhados? Para JQuery?

Sim, poderia fazer uma referência a arquivos JQuery ou o que seja que esteja utilizando. Sim.

**Pergunta:** A minha pergunta é a mesma, se poderemos migrar os controles que hoje já tenho e uso do Marketplace?

Sim, poderá migrar... a ideia é começar em algum momento a converter certos User Control para este novo esquema justamente para facilitar e ter o poder que o objeto oferece.

**Pergunta:** Tenho uma consulta sobre Sketch, se estará disponível também para Web a importação do Sketch?

A importação de Sketch para Web?...

A princípio temos para SD... não há nada que nos impeça de fazê-lo para Web, mas para hoje temos justamente os Users Controls que estamos priorizando por agora. Queremos... mas antes prioridade para SD, pois os Designers podem fazer HTML, Javascript e criar como User Controls.

**Pergunta:** Eu queria fazer uma pergunta sobre Deploy do User Control... se temos referência a uma Library. E quando fazemos o Deploy dessa aplicação se vamos levar essa Library? Se há algum build?

Sim. Deveria levar, com relação às referências há várias opções, digamos, pode ter referências a coisas que tem dentro de seu modelo, digamos, arquivos estáticos, por exemplo JQuery ou pode usar CDN também, pode usar referência à URL desses arquivos que tem em algum lugar e você pode referenciar.

Nesse caso da referência ao CDN não deveria adicionar ao Deploy e no caso de que utilize alguns arquivos estáticos em seu modelo... sim.

**Pergunta:** Como interage GeneXus?

Bom o motor de Deploy, agora estamos trabalhando justamente para que calcule as referências e possa levar a estas dependências.

Como no caso de Deploy para TomCat que fazemos quando prototipamos local, a ideia e também que leve, digamos, essa library e as dependências que tenha o controle.



**Streams of information  
and more about grids**

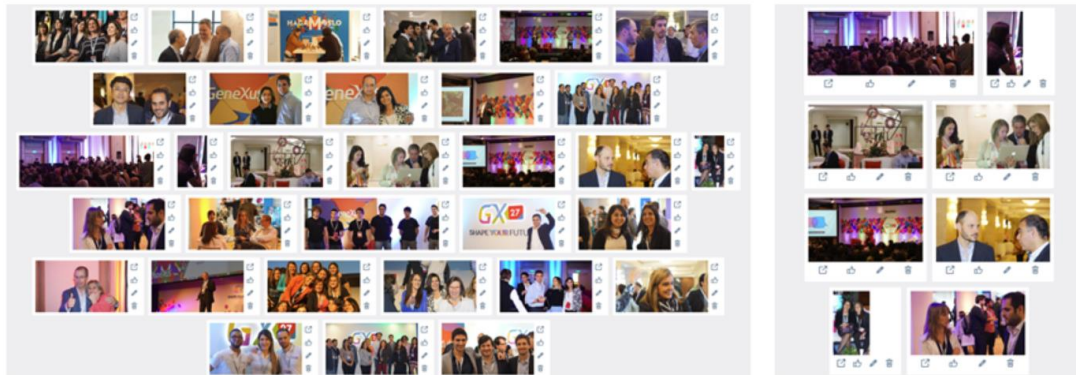
## Flex layouts



Vamos falar um pouco sobre Flex layouts que estamos trabalhando também. E que Flex layouts, basicamente o que permite é maior flexibilidade como lhes comentou Armin no Overview. A ideia não é apenas trabalhar com o poder que nos oferece Bootstrap, no conceito da divisão da tela em doze colunas e como os dados devem ser reorganizados quando eu jogo com os tamanhos da tela. Isto é complementar e, digamos, dá uma flexibilidade maior já que nos permite, como nos disse Armin antes, este tipo de interface.

Web/SD

GeneXus



Flex layout container, Flex Grid

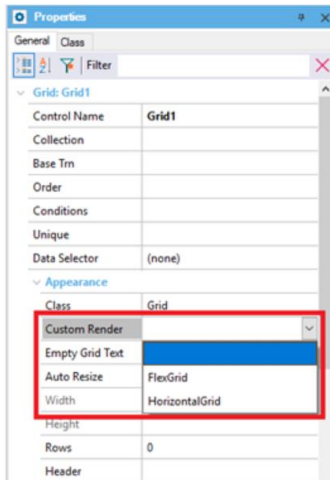
Onde não sabemos exatamente qual é a largura que tem cada um dos elementos, pode ter uma largura variável e também a possibilidade de calcular e de mover estes elementos com base no tamanho da tela. Também, não só mover o elemento, todo o elemento em si, mas também poder ajustar particularmente as tabelas e conteúdo de cada elemento com base no tamanho.

Neste caso, em execução Web vemos que temos cada uma das fotos com suas ações à direita e nesta tela menor, as ações abaixo, se colocam abaixo. Com isso, falamos de Flex Layout, com certas funcionalidade que tem CSS no caso da Web, o que nos permite modelar este tipo de interfaces de uma forma muito mais potente que o Bootstrap inicialmente, com a divisão de doze colunas.

Web

## Flex layout container, Flex Grid

GeneXus



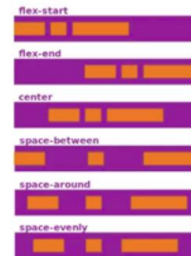
### Flex Direction



### Flex Wrap



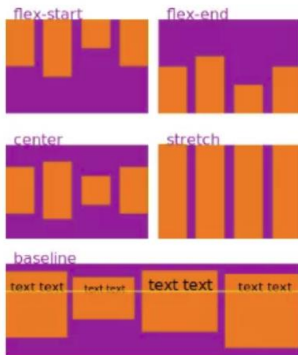
### Justify Content



Para o caso de Web, o que temos é uma nova forma de visualizar o grid. Um Custom Render do Grid, que aqui permite definir com Flex View e, no momento de habilitar esta propriedade o que o IDE faz é disponibilizar estas outras propriedades, que dizem justamente como deve ser comportar este Flex layout. A direção que terá, se é por linha ou por coluna, se é linha invertida ou coluna invertida, aqui são essas quatro. Como se faz o Wrap dos elementos. O que acontece, digamos, quando eu tenho uma certa linha com uma quantidade de elementos e o tamanho não é suficiente para mostrar todos. Por exemplo, aqui o que está fazendo é jogá-los para a próxima. Como irá justificar o conteúdo, se começa da esquerda para a direita, da direita para a esquerda, se estará centralizado, se deixaremos espaço entre os elementos. Aqui há muitas opções que nos permitem, justamente, esta flexibilidade.

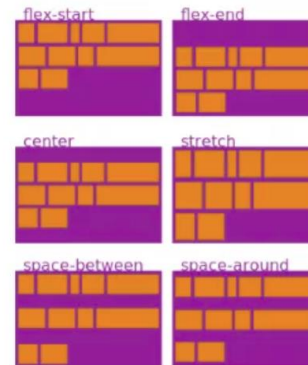
## Align Items

It defines the alignment of the items in the rows.

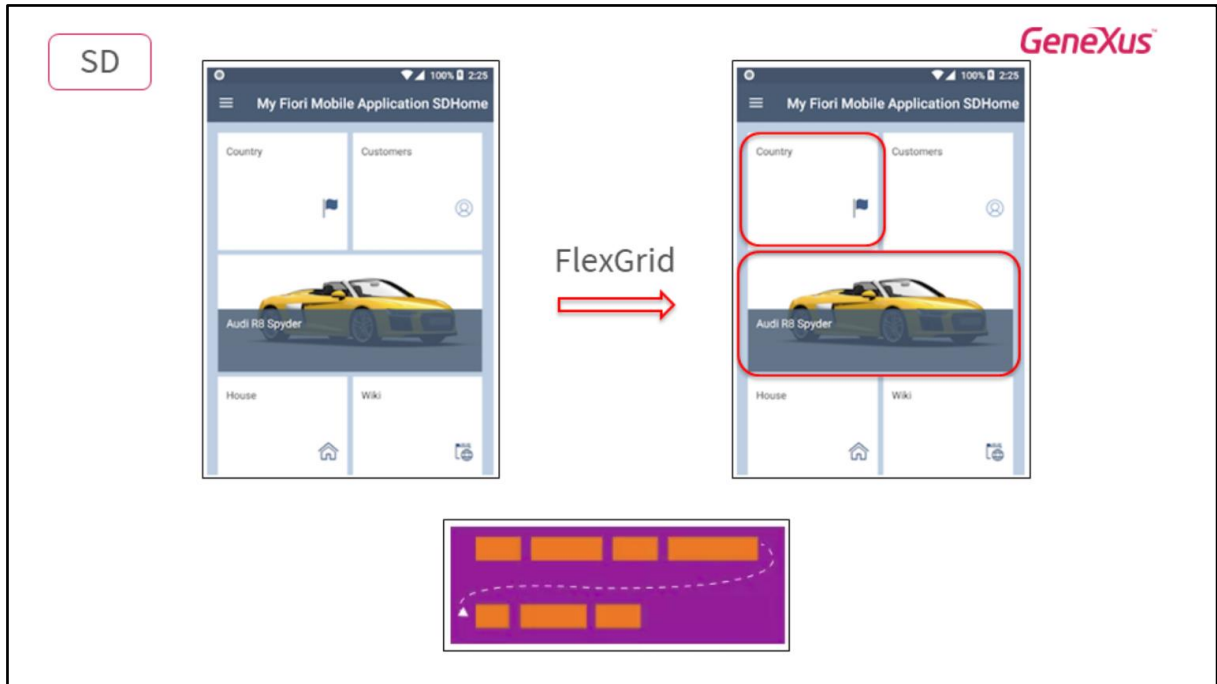


## Align Content

Aligns the row when there is extra space in the container.



Mais opções dentro de Flex layout, como queremos o alinhamento dos itens, isto é basicamente como se alinham itens nas linhas.  
Aqui separou um pouco o texto mas, é como alinharemos a linha quando há espaço extra nos containers. Se nos sobra espaço podemos visualizar este conteúdo de diferentes maneiras.



Bem, e com relação a SD, digamos, aplica-se o mesmo conceito que estava sendo dito para Web, referente a UI. Neste caso em SD é quando queremos mostrar componentes com largura variável. Temos aí como exemplo este tipo de componente que são de uma determinada largura e depois este outro que ocupa toda a largura da tela então, vão se acomodando com Flex Grid. Vão se acomodando conforme o espaço do container se este o permite. Se este é, por exemplo, um tablet, se acomodariam em sua linha, por exemplo. E quando se tem menos espaço, iria para baixo. Por exemplo, com a propriedade que comentava, o Flex Wrap que mostra no desenho, se vê o comportamento de como vai carregando os itens.

SD

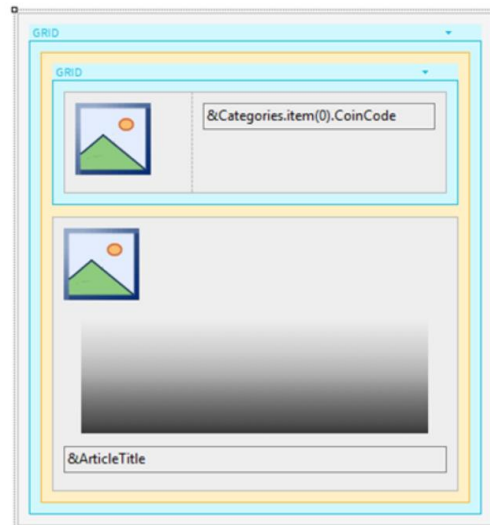
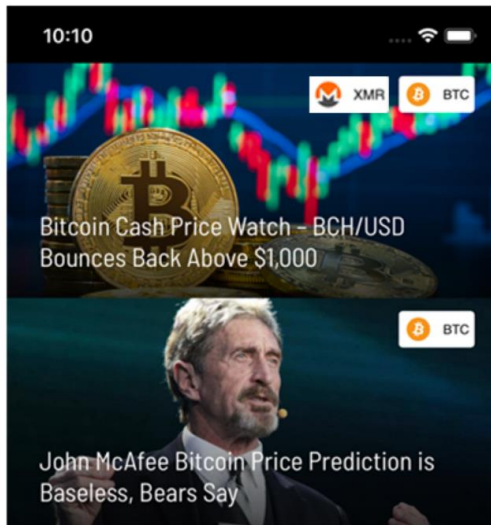
## BitBitNews

Flex + Nested Grids

GeneXus

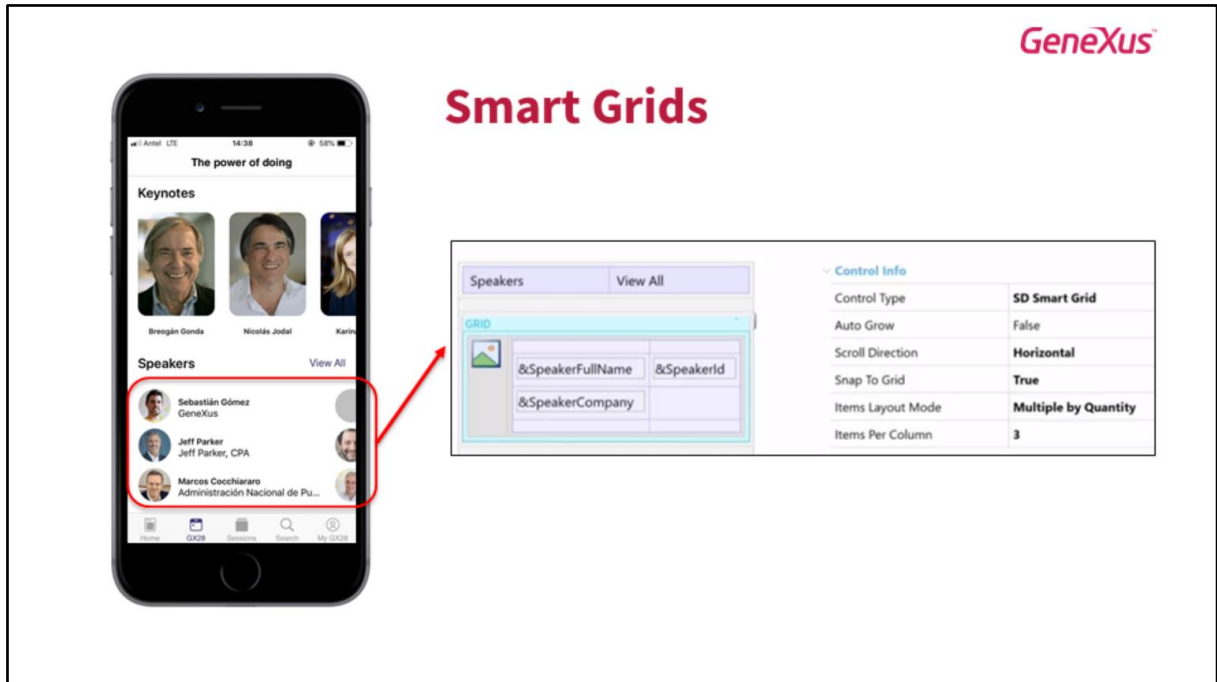
Justify Content

flex-end



Então, para agregar algo mais, no wiki podemos ver o exemplo desta aplicação BitBitNews, que é uma aplicação de notícias sobre criptomoedas, então, temos um grid com as notícias de criptomoedas e aninhado temos outro grid que mostra a criptomoeda associada a esta notícia. Então, como não sabemos esta quantidade, nem tamanho, dizemos com Justify Content que irão se agrupando estas criptomoedas da direita para a esquerda na parte superior da notícia.

## Smart Grids



Então, um pouco mais sobre Grids, isto também na versão 16. Temos Smart Grids o que traz muito mais poder aos grids que vínhamos trabalhando. Este é um exemplo da KB do evento, este com os palestrantes, o usamos para conseguir o comportamento deste grid de palestrantes que é paginado horizontal à direita, no qual pode ser visto como um Hint, uma dica de que a navegação será à direita porque, se vê um pedacinho do item que está oculto. Se conta como podem ver com três itens, três palestrantes por coluna e esta paginação. Antes, se quiséssemos fazer desta forma, poderíamos fazer mas, tínhamos que codificar muito mais. Agora, é simplesmente como na imagem que veem à direita, aplicando SD Smart Grid com Scroll Direction Horizontal e, com itens por coluna, três... só com esta programação já temos o resultado que vemos aqui na tela.



**More about...**



Então, algumas coisas mais:

## More about Grids

SD

- Unique property for grids (Web & SD)
- Inverse Loading property
- Grid Refresh method is available on SDPanels and Work With for Smart Devices
- Improvements to Grids with Multiple Selection for Smart Devices
- ShowPageController property in runtime

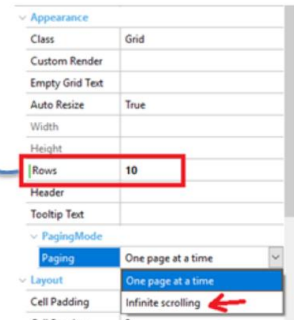
GeneXus

- A Unique Property para grids, tanto em Web, como Smart Devices, se adicionou
- Inverse Loading nos grids, é a carga inversa, este uso poderia ser em um aplicação de chat...
- O Refresh Method para grids em SDPanels e Work With for Smart Devices
- Arranjos e melhorias no que é Multiple Selection for Smart Devices
- E a mudança em runtime da propriedade ShowPageController em grids

## More about Grids

Web

- Unique property for grids
- Inverse Loading property (when **Rows  $\neq$  0**) → use case: chat
- It's possible to set up a Fixed Grid header with vertical scroll
- Header property and Header Text Class property (for grids and tables)
- Select method and Deselect method



GeneXus

Alguma coisa mais sobre grids em web...

Bem, sobre os grids em Web também, melhorias que tivemos nesta última versão.

- Também a propriedade Unique
- O Inverse Loading também. Tem esta consideração quando as Rows são diferentes de zero, também no caso do chat
- É possível setar um header fixo no grid. O tema é a edição dos grids e a modelagem dos estilos, será bem mais simples agora
- Header Property e Header Text que são duas novas propriedades e Select e Deselect, que também fizemos muitas melhorias.

**Pergunta:** Com relação ao tema de CSS Frameworks, estava falando que se subia como arquivo com as referências das classes disto. Haverá alguns incluídos em GeneXus, que já tenha dentro de suas classes, suas referências?

A princípio não. Porque, não vemos necessidade de já fornecer essas coisas que sobrecarreguem a KB, digamos, partindo do zero e isto deixamos para o usuário decidir.

**Pergunta:** Mas, por exemplo, o Bootstrap que já está incluído e empacotado dentro do GeneXus...

Sim, mas, o Bootstrap tem a particularidade que o utilizamos justamente para resolver o tema de colunas responsivas. Por isso, usamos inicialmente o Bootstrap com todo o set de coisas, não somente os scripts que resolvem, digamos, como devem comportar-se... a aplicação com base no tamanho mas, também alguns estilos que já estavam aplicados a controles que nós distribuíamos. Agora, o que fizemos foi separar isto. Deixar o poder do Bootstrap para calcular o tamanho do referido controle na interface e permitir integrar outros CSS Frameworks que possam coexistir com esta lógica mas, não com o estilo Bootstrap.

**Pergunta:** Então, nós seremos responsáveis por manter a versão do Framework que queremos?

Correto, dependendo de que controle queira utilizar e de qual seja seu provedor, digamos, neste caso SemanticUI, você vai importar este CSS Framework e vai utilizá-lo ao modo controles.

**Pergunta:** Dizia que quando passa a importar um objeto pode fazer uma referência a um CSS Framework ou inclusive a um CSS feito por designer (sobre este CSS Framework). O que acontece se o designer usa outras classes que não são classes do CSS Framework, são classes definidas por ele, no momento da importação? Isto substitui as classes que já tenho no tema

da KB GeneXus? Ou como funciona neste caso? Em nosso caso em particular, trabalhamos com designers, que montam um CSS e depois a consolidação é um processo manual que temos que fazer.

Claro, bem, aí depende de muitas coisas, digamos, do CSS mesmo... se utilizar os mesmos nomes de classes... se seu designer utiliza os mesmos nomes de classes que você já tenha definido, aí provavelmente vai haver um conflito e você terá que definir prioridades na propriedade da classe para que uma ganhe da outra mas, isto, digamos, as classes podem complementar, podem utilizar as classes de seu designer ou do CSS Framework e adicionar classes próprias suas ao tema, com a concatenação de classe que permite o CSS, um espaço no nome da outra classe. Creio que já está muito bem resolvido já que com este poder você terá uma modelagem bem mais simples do que é agora.

**Pergunta:** Não há um processo de importação de um arquivo que estava numa consolidação? Não. O CSS que gera seu tema é independente do que vai utilizar, digamos, o CSS Framework. O CSS Framework ao ser inserido em GeneXus, definirá este arquivo estático que se chamará, por exemplo, SemanticUI.css, que será independente do que gera seu objeto tema.

Mobile & Web



Multi-experience & Omnichannel

- Design systems
- Streams of information
- **Conversational UI: Chats/ Chatbots**



Web/ Web mobile

Native mobile

Chatbots

Wearables/TV/AR

*GeneXus*<sup>™</sup>  
**The power of doing**