Mais sobre For Eachs aninhados

Casos e navegação

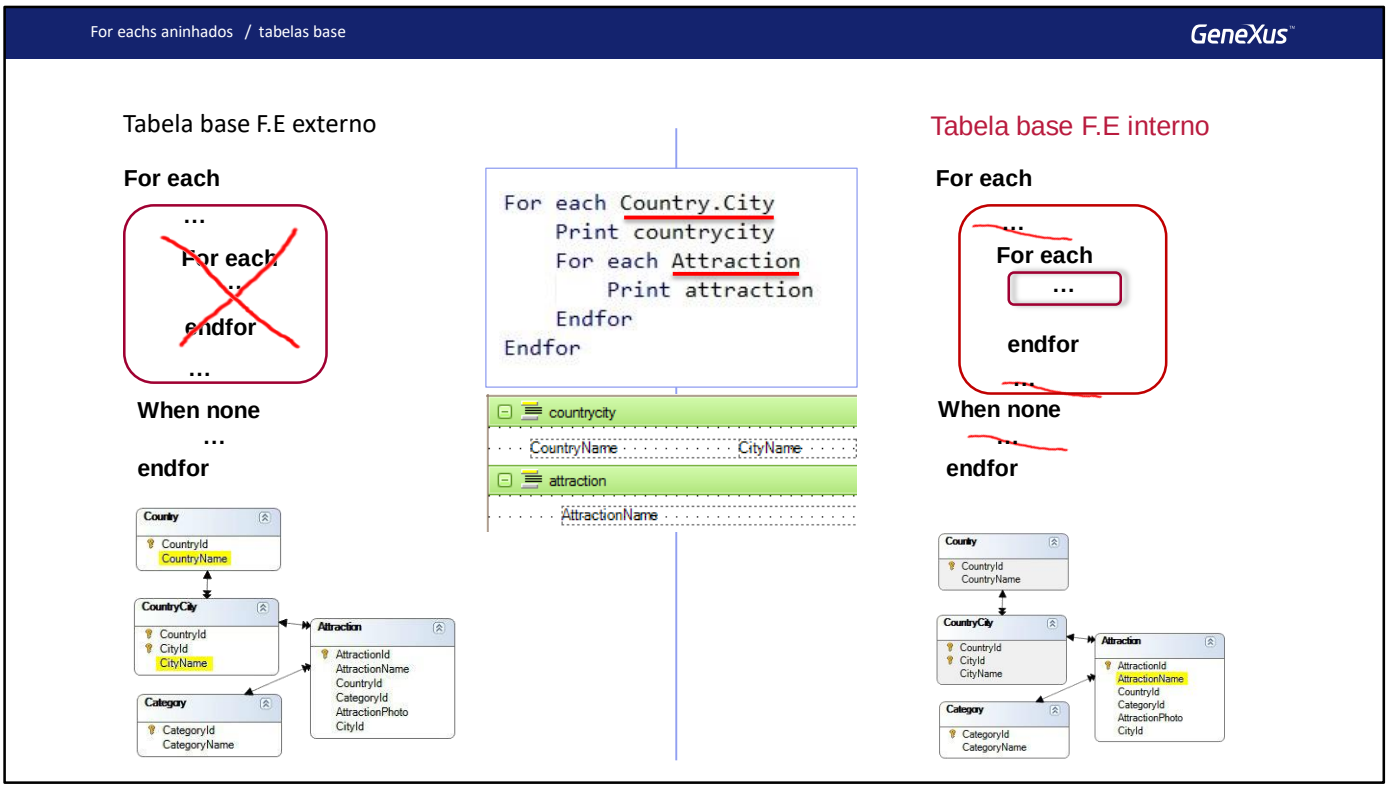
GeneXus 16

Tempo estimado de leitura: 20 min

Tabela base

Tabelas base e Navegação

- ✓ Diante um caso de For eachs aninhados primeiramente o Genexus deve determinar as tabelas base de cada um.
  - Quando o desenvolvedor indica a Transação Base, isto é imediato.
  - Quando não o fizer, GeneXus deve determinar cada tabela base a partir dos atributos presentes em cada for each. Este caso, mais complexo, não será abordado neste curso.
- ✓ Logo, a partir disso, define-se a navegação para resolver a consulta múltipla. Que será uma das três situações:
  - Join
  - Produto Cartesiano
  - Corte de controle



Como dissemos, a primeira coisa que GeneXus faz ao encontrar um par de For Eachs aninhados é determinar a tabela base de cada um, de maneira ordenada, a partir do exterior, começando pelo mais externo. Só então determina a navegação.

Para for each a transação base indicada intervém, e somente os atributos daqueles for each: tanto da ordem, onde, etc., quanto aqueles que estão em seu corpo, exceto aqueles que estão dentro de um outro for each aninhado. Ou seja, removendo o for each aninhado, a tabela base é determinada como no caso de um for each simples. Os atributos da cláusula When none nunca são levados em conta. Todos os atributos devem pertencer à tabela estendida da tabela base encontrada. Os atributos que não estão de acordo com isso não serão "instanciáveis", pois não podem ser alcançados.

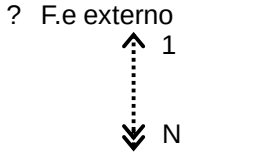
No exemplo acontece nesta ordem :

1) A **tabela base** do for each é determinada. Para isso, considera-se a transação base indicada, isto é, Country.City, e verifica-se que os atributos presentes no printblock (CountryName e CityName) pertencem à sua tabela estendida. Se esse não for o caso, um aviso será produzido na lista de navegação, informando que alguns atributos não podem ser instanciados, uma vez que não podem ser alcançados a partir da tabela estendida do f.ex. Nesse caso, CountryName e CityName pertencem à CountryCity estendida, tabela base para cada.

2) A **tabela base** do for each aninhado é determinada. A transação base de Attraction é considerada e o atributo AttractionName presente no bloco de impressão. Os atributos do for each externo não são levados em conta. Se nenhuma transação base tiver sido gravada, então algo relativo aos atributos do for each seria utilizado para determinar a tabela base interna, mas esse não é o caso. Em seguida, sua tabela base é determinada como se fosse um for each independente. Portanto, sua tabela base será a atração.

Casos e Navegação

Tabelas base ≠



Existe relação implícita que vincule ao For Each externo com um número N de registros do For each aninhado?

Sim → **Join**: são recuperados alguns registros do aninhado: os relacionados. (Caso 1)

Não → **Produto cartesiano**: são recuperados todos os registros do aninhado. (Caso 2)

Tabelas base =

Corte de Controle: Corresponde o caso no qual queremos recuperar informação agrupada. (Caso 3)

A partir da determinação das tabelas básicas, veremos três casos de Para cada ano que vimos antes e que queremos conceituar aqui.

Quando as **tabelas base** são **diferentes**, duas possibilidades se abrem: ou existe um **relacionamento direto ou indireto de 1 a N** entre elas, ou não existe. No primeiro caso, para cada registro do for each principal, o for each aninhado executará suas instruções apenas para os N registros relacionados. Essa operação de filtrar as informações de uma tabela pela de outra, é conhecida como **Join**. No segundo caso, quando não há relacionamento, para cada registro considerado no for each principal, o for each aninhado executará suas instruções para todos os registros na outra tabela, uma vez que não encontrou relação entre eles. A operação é conhecida como **Produto Cartesiano**. (Na próxima página veremos outro caso, mas que responde a uma programação ruim).

Quando as **tabelas base** forem **as mesmas**, se trata de um caso conhecido como Corte de Controle: é quando precisamos agrupar as informações de uma tabela, executar certas instruções que levam em conta as informações comuns do grupo e, em seguida, passar para cada membro do grupo e executar outras instruções, passe para o próximo grupo e repita o processo. Nesse caso, é essencial especificar os atributos que compõem o grupo, usando a cláusula **order**.

- **Tabelas base ≠**

F.e externo
N
1
F.e aninhado

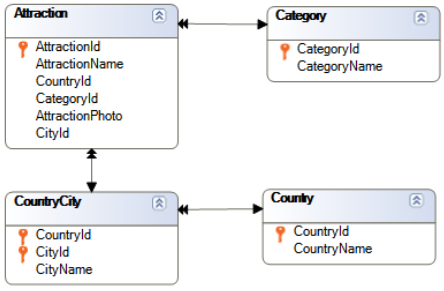
Este é um caso raro, mal programado: quando a tabela base do For each aninhado é parte da tabela estendida del For each externo.

(Caso 4)

```
For each Attraction
    print attraction
    for each Country.City
        print countrycity
    endfor
endfor
```

=

```
For each Attraction
    print attraction
    print countrycity
endfor
```



Este caso surge apenas quando a transação base é utilizada para o for each aninhado Caso contrário, o GeneXus terá que calculá-lo sozinho e, neste caso, escolherá para o for each aninhado a mesma tabela de base que a do pai, em seguida, implementando então um corte de controle. Não veremos isso neste curso.

Note que aqui o segundo for each seria desnecessário, porque para cada atração em que ele está posicionado em um determinado momento no for each externo, há apenas um registro CountryCity relacionado. Seria o mesmo, então, não ter escrito o segundo for each, e enviar diretamente para imprimir o bloco de impressão countrycity, que contém CountryName e CityName, ambos pertencentes à tabela Extended Attraction.

Exemplos de cada caso

Tabelas base ≠

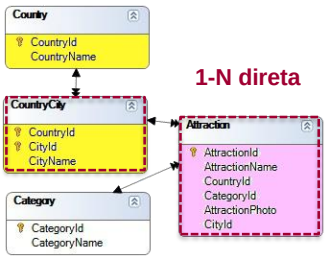
Join

```
For each Country.City
    Print countrycity
Endfor
```

countrycity: CountryName, CityName

1 ↕ N

```
For each Attraction
    Print attraction
Endfor
```

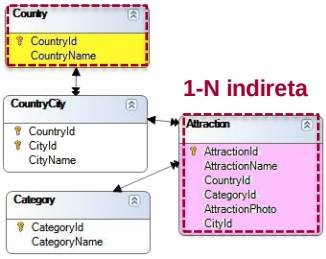



```
for each Country
    print country
Endfor
```

country: CountryName

1 ↕ N

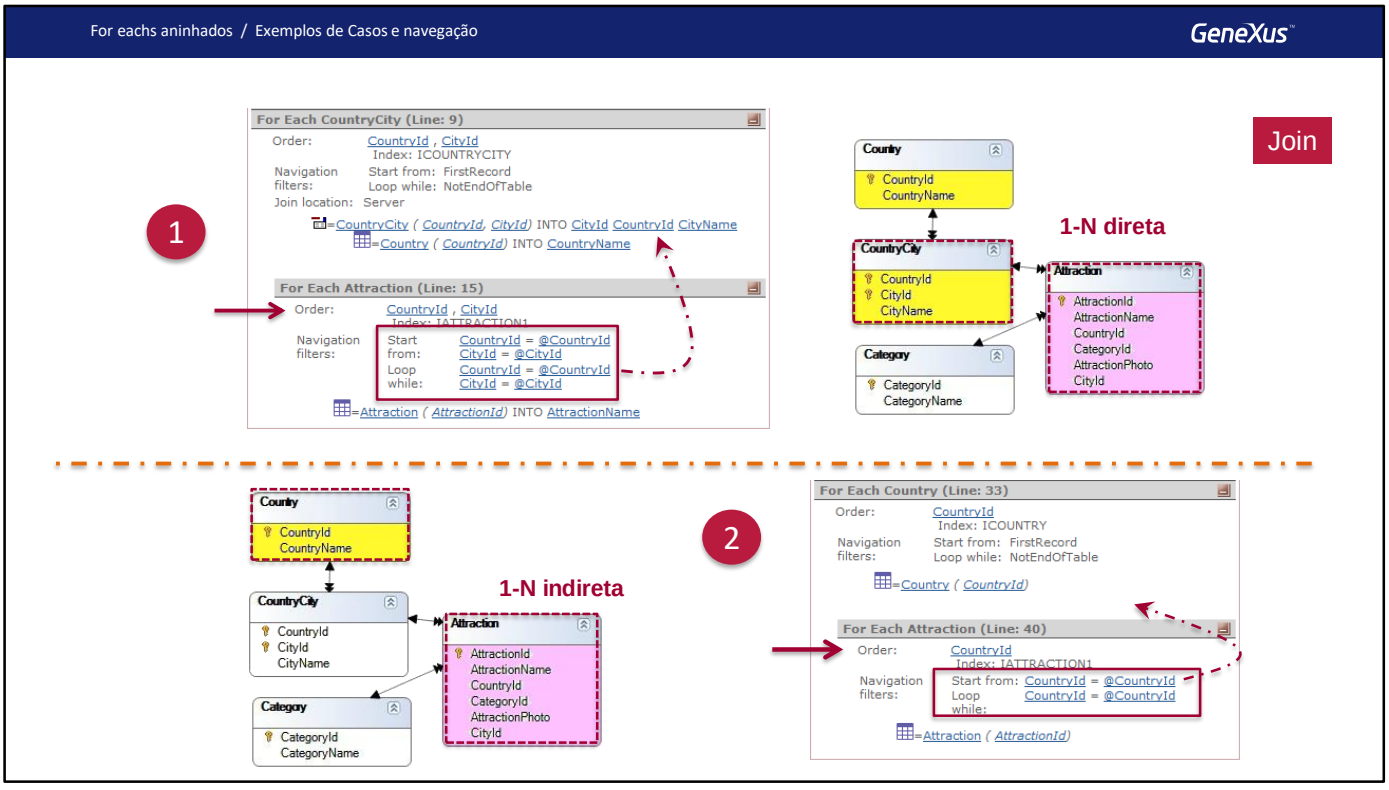
```
For each Attraction
    Print attraction
Endfor
```



Aqui temos dois casos de relação 1 para N.

A primeira é direta. Observemos que as tabelas base do for each externo y aninhado são CountryCity y Attraction, respectivamente, que estão relacionadas por una relação 1 para N.

A segunda é indireta. As tabelas base do for each externo e aninhado são Country e Attraction, que não tem uma relação direta 1 para N, porém são sim indireta, através da tabela CountryCity. Dizendo de outra maneira : observemos que a tabela base do primeiro for each (Country), está incluída na tabela estendida da tabela base do for each aninhado (Attraction).

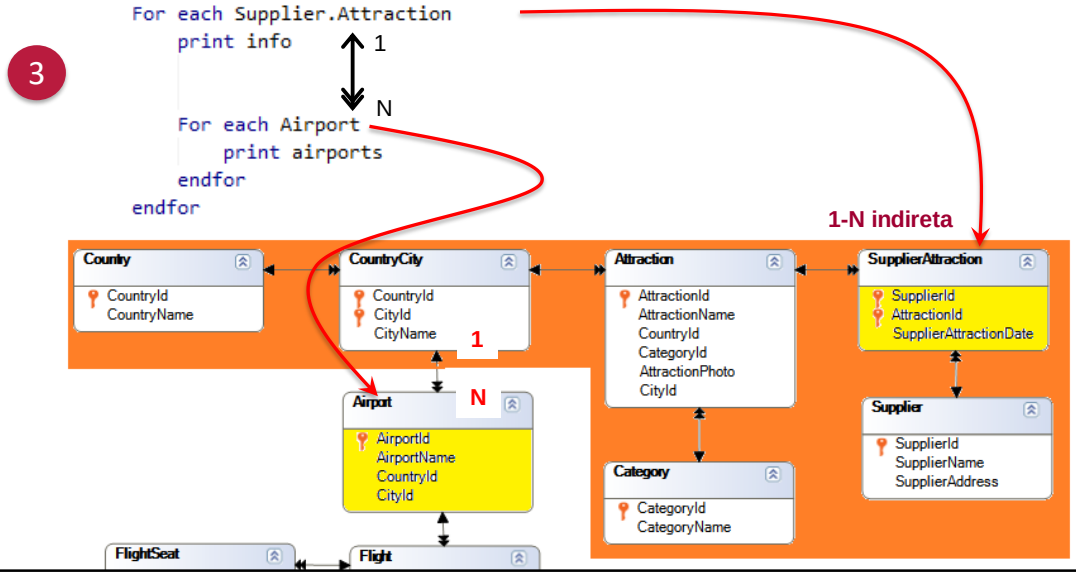


As listagens de navegação indicam claramente o Join: para o for each aninhado, a tabela inteira não é percorrida. Observe que, em ambos os casos, em vez de ordenar pesquisado do for each aninhada pela chave primária de Attraction, que é AttractionId, ela o faz pelo atributo ou conjunto de atributos de relacionamento, para o qual possui um índice criado automaticamente da chave estrangeira. Desta forma, o acesso ao banco de dados estará otimizado.

Portanto, quando o GeneXus determina que realizará um Join, ele tenta otimizar sua navegação.

Tabelas base ≠

Join

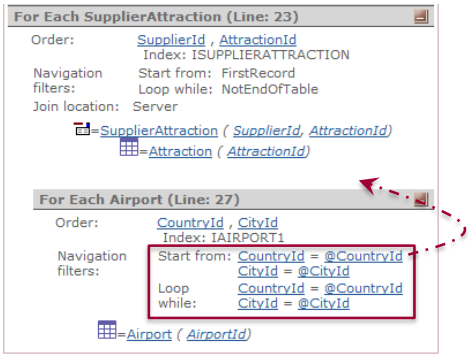


Aqui vemos um terceiro exemplo, em que a tabela base do for each externo é SupplierAttraction e a tabela base do for each aninhado é Airport. Observe que há um relacionamento indireto de 1 para N. Ou seja, para cada registro de SupplierAttraction, haverá apenas uma Attraction da qual obtemos apenas uma da CountryCity, que, por sua vez, está relacionada com N registros do aeroporto (os N aeroportos que estão nesse país / cidade, embora em geral, na realidade, há apenas um. Em nosso modelo, pode existir vários).

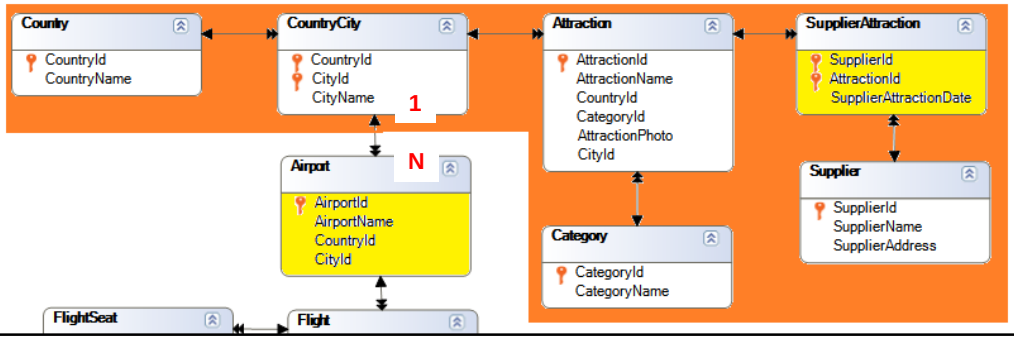
Tabelas base ≠

3

```
For each Supplier.Attraction
    print info
                          1
                          N
    For each Airport
        print airports
    endfor
endfor
```



Join



Ao rodar bem, GeneXus descobre que há atributos em comum entre a tabela estendida do for each principal e a tabela base do aninhamento. Quais? O par {CountryId, CityId}. E para eles ele vai fazer o join.

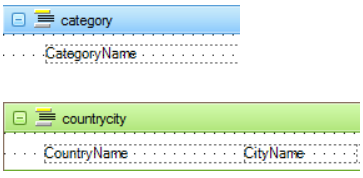
Tabelas base ≠

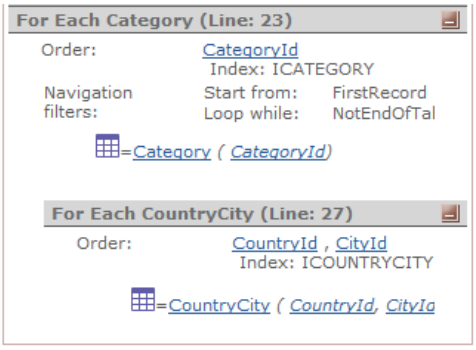
Produto Cartesiano

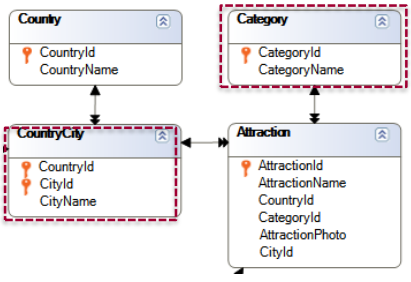
```
For each Category
    Print

    For each Country.City
        Print
    Endfor

Endfor
```







Nesse caso, GeneXus não consegue encontrar um relacionamento 1-N direto ou indireto entre as tabelas e, portanto, não aplica filtros implícitos aos registros do For Each aninhados. Ou seja, faz um produto cartesiano entre as tabelas: para cada registro na tabela base do for each externo (categoria), considera todos os registros na tabela base do aninhamento (CountryCity).

Tabelas base =

Corte de controle

```
For each Attraction order CountryName
    Print
        country
            Country: CountryName

    For each Attraction order CityName
        Print
            city
                City: CityName

        Print
            attractionTitles
                Attraction

        For each Attraction
            Print
                attraction
                    AttractionName
        Endfor

    Endfor

Endfor
```

Country: Brazil
City: Rio de Janeiro
Attraction
The Christ Redeemer
City: Sao Paulo
Attraction
Museum of Football
Country: China
City: Beijing
Attraction
Great Wall
Country: Egypt
City: Cairo
Attraction
Egypt Pyramides

Aqui vemos um caso em que queremos listar cada país, e para ele cada cidade, e para ela cada atração. A restrição: queremos fazê-lo apenas para países e cidades para os quais existem atrações turísticas.

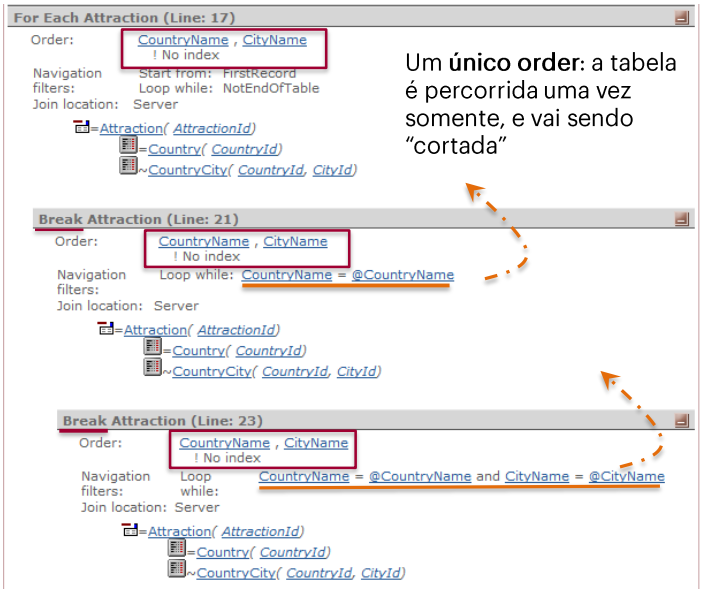
Ou seja, teremos que implementar um **corte de controle duplo**: onde primeiro **agruparemos** por país e, dentro desse grupo, então **agrupamos** por cidade e, dentro dela, **mostrar** os nomes de todas as atrações. Para fazer isso:

Teremos que definir os critérios de agrupamento usando as **cláusulas order**. Lembre-se de que, para um corte de controle, o order tem um peso muito forte: ele não apenas marca o atributo ou atributos para listar as informações, mas também especifica como ele será agrupado.

Poderíamos especificar um order para o for ach mais interno, mas esse order só terá seu uso convencional. Ou seja, que será usado apenas para ordenar.

Tabelas base =

Corte de controle



Nós temos um corte de controle duplo, o que implica três for eachs. No order do primeiro, é estabelecido o grupo mais externo, no segundo, o grupo interno.

Se olharmos para a lista de navegação, vemos a palavra Break para cada for each interno, indicando a mesma tabela base Attraction e, portanto, um corte de controle

Além disso, passará por essa tabela base apenas uma única vez, para o qual é necessáriorordenar a concatenação dos atributos que aparecem os orders do for each. É por isso que você escolhe CountryName, CityName.

Note que no segundo for each ele corta por país, iterando sobre o país em que está posicionado no primeiro for each. E o terceiro for each, corta em cidade, percorrendo a cidade em que está posicionado no segundo for each.

Pense na execução da lista anterior se, em vez de ter ordenado o primeiro for each por CountryName e o segundo por CityName, tivéssemos ordenado pelo par CountryName, CityName. Observe que, nesse caso, a lista de navegação será diferente da que você vê acima, no segundo for each. Loop while será "CountryName = @CountryName e CityName = @CityName".

GeneXus™

The power of doing.

Vídeos	training.genexus.com
Documentação	wiki.genexus.com
Certificações	training.genexus.com/certifications