



DESENVOLVIMENTO DE SISTEMAS COM C#

Cleverson Lopes Ledur

Revisão técnica:

Jeferson Faleiro Leon

*Desenvolvimento de Sistemas
Especialista Formação Pedagógica de Professores
Professor do curso Técnico em informática*



L475d Ledur, Cleverson Lopes.

Desenvolvimento de sistemas com #C [recurso eletrônico] / Cleverson Lopes Ledur; [revisão técnica: Jeferson Faleiro Leon]. – Porto Alegre : SAGAH, 2018.

ISBN 978-85-9502-314-7

1. Ciência da computação. 2. Linguagens de programação de computador. I. Título.

CDU 004.43

Criar relatórios e utilizá-los em uma Aplicação Windows Forms – II

Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Reconhecer o componente DataSet.
- Criar um DataTable com TableAdapter.
- Gerar relatórios com relacionamento entre tabelas utilizando DataSet e TableAdapter.

Introdução

Criar um relatório com apenas uma tabela é uma tarefa bastante simples no Visual Studio. No entanto, quando você precisa inserir mais de uma tabela e há relacionamentos entre elas, essa tarefa fica um pouco mais complexa. Para isso, você pode utilizar dois componentes chamados de DataSet e TableAdapter. Basicamente, o TableAdapter permite que você crie tabelas (consultas) em que reúne diversas tabelas utilizando as relações entre elas. Se deseja dados de duas tabelas A e B, você pode criar uma tabela C com um TableAdapter que faz um Join nessas duas tabelas. Tudo isso fica armazenado dentro de um DataSet. Então, você utiliza esse DataSet para pegar os dados para o relatório.

Neste capítulo, você vai aprender mais sobre o DataSet e o TableAdapter. Além disso, vai ver como utilizá-los para criar relatórios com relacionamentos.

O componente DataSet

O componente DataSet é um conjunto de objetos que armazenam dados de um banco de dados em memória. Ele também acompanha o rastreamento de alterações para ativar operações de criação, leitura, atualização e exclusão (CRUD) desses dados sem a necessidade de estar sempre conectado ao banco de dados. Os componentes DataSet foram projetados para formulários simples em aplicativos de negócios de dados. Para novas aplicações, considere usar Entity Framework para armazenar e modelar dados na memória.



Fique atento

É importante que, para trabalhar com conjuntos de dados, você já tenha algum conhecimento básico de conceitos de banco de dados.

A seguir, você vai conhecer um pouco mais sobre as propriedades e métodos desse componente, já que, em alguns casos, sua utilização é extremamente necessária (MICROSOFT, c2018a).

Propriedades do componente

O componente DataSet possui uma grande variedade de propriedades que permitem maiores controle e caracterização dos dados. Veja, no Quadro 1, as principais propriedades desse componente.

Quadro 1. Propriedades do DataSet.

Nome	Descrição
CaseSensitive	Obtém ou define um valor que indica se comparações de cadeia de caracteres dentro de objetos DataTable diferenciam maiúsculas de minúsculas.
Container	Obtém o contêiner para o componente (herdado de MarshalByValueComponent).
DataSetName	Obtém ou define o nome do DataSet atual.

(Continua)

(Continuação)

Quadro 1. Propriedades do DataSet.

Nome	Descrição
DefaultViewManager	Obtém uma exibição personalizada dos dados contidos no DataSet para permitir a filtragem, a pesquisa e a navegação usando um DataViewManager personalizado.
DesignMode	Obtém um valor que indica se o componente está atualmente no modo de design (herdado de MarshalByValueComponent).
EnforceConstraints	Obtém ou define um valor que indica se as regras de restrição são seguidas ao tentar qualquer operação de atualização.
Events	Obtém a lista de manipuladores de eventos que estão anexados ao componente (herdado de MarshalByValueComponent).
ExtendedProperties	Obtém a coleção de informações de usuário personalizadas associadas a um DataSet.
HasErrors	Obtém um valor que indica se há erros em qualquer um dos objetos DataTable no DataSet.
IsInitialized	Obtém um valor que indica se o DataSet é inicializado.
Locale	Obtém ou define as informações de localidade usadas para comparar cadeias de caracteres na tabela.
Namespace	Obtém ou define o namespace do DataSet.
Prefix	Obtém ou define um XML prefixo que nomeia o namespace do DataSet.
Relations	Obtém a coleção de relações que vinculam tabelas e permitem a navegação de tabelas pais para tabelas filhas.
RemotingFormat	Obtém ou define um SerializationFormat para o DataSet usado durante a comunicação remota.
SchemaSerializationMode	Obtém ou define um SchemaSerializationMode para um DataSet.
Site	Obtém ou define um System.ComponentModel.ISite para o DataSet (substitui o MarshalByValueComponent.Site).
Tables	Obtém a coleção de tabelas contidas no DataSet.

Métodos do componente

O componente DataSet possui um conjunto de métodos que permitem a manipulação dos dados presentes nele. Além da manipulação, também há métodos de leitura de dados.

Veja, no Quadro 2, os principais métodos desse componente.

Quadro 2. Métodos do DataSet.

Nome	Descrição
AcceptChanges()	Confirma todas as alterações feitas nesse DataSet desde que foi carregado ou desde a última vez que AcceptChanges foi chamado.
BeginInit()	Começa a inicialização de um DataSet usado em um formulário ou por outro componente. A inicialização ocorre no tempo de execução.
Clear()	Limpa o DataSet de todos os dados, removendo todas as linhas em todas as tabelas.
Clone()	Copia a estrutura do DataSet, incluindo todos os esquemas, relações e restrições DataTable. Não copia todos os dados.
Copy()	Copia a estrutura e os dados para o DataSet.
CreateDataReader()	Retorna um DataTableReader com um resultado definido por DataTable, na mesma sequência em que as tabelas são exibidas na coleção Tables.
CreateDataReader (DataTable[])	Retorna um DataTableReader com um conjunto de resultados por DataTable.
DetermineSchema-SerializationMode(Serializat ionInfo, StreamingContext)	Determina o SchemaSerializationMode para um DataSet.
DetermineSchemaSeriali zationMode(XmlReader)	Determina o SchemaSerializationMode para um DataSet.
Dispose()	Libera todos os recursos usados pelo MarshalByValueComponent (herdado de MarshalByValueComponent).

(Continua)

(Continuação)

Quadro 2. Métodos do DataSet.

Nome	Descrição
Dispose(Boolean)	Libera os recursos não gerenciados usados pelo MarshalByValueComponent e opcionalmente libera os recursos gerenciados (herdado de MarshalByValueComponent).
EndInit()	Termina a inicialização de um DataSet que é usado em um formulário ou usado por outro componente. A inicialização ocorre em tempo de execução.
Equals(Object)	Verifica se o objeto especificado é igual ao objeto atual (herdado de Object).
Finalize()	Permite que um objeto tente liberar recursos e executar outras operações de limpeza antes que ele seja recuperado pela coleta de lixo (herdado de MarshalByValueComponent).
GetChanges()	Obtém uma cópia do DataSet que contém todas as alterações feitas nele desde que ele foi carregado ou desde que o AcceptChanges foi chamado pela última vez.
GetChanges(DataRowState)	Obtém uma cópia do DataSet que contém todas as alterações feitas desde o último carregamento ou desde que AcceptChanges foi chamado, filtrado por DataRowState.
GetDataSetSchema(XmlSchemaSet)	Essa API dá suporte à infraestrutura do produto e não se destina a ser usada diretamente no seu código. Obtém uma cópia de XmlSchemaSet para DataSet.
GetHashCode()	Serve como a função de hash padrão (herdado de Object).
GetObjectData(SerializationInfo, StreamingContext)	Povoa um objeto de informações de serialização com os dados necessários para serializar o DataSet.
GetSchemaSerializable()	Essa API dá suporte à infraestrutura do produto e não se destina a ser usada diretamente no seu código. Retorna uma instância T:System.Xml.Schema.XMLSchema serializável.

(Continua)

(Continuação)

Quadro 2. Métodos do DataSet.	
Nome	Descrição
GetSerializationData(SerializationInfo, StreamingContext)	Essa API dá suporte à infraestrutura do produto e não se destina a ser usada diretamente no seu código. Desserializa os dados da tabela do binário ou fluxo XML.
GetService(Type)	Obtém o implementador do IServiceProvider (herdado de MarshalByValueComponent).
GetType()	Obtém o Type da instância atual (herdado de Object).
GetXml()	Retorna a representação XML dos dados armazenados no DataSet.
GetXmlSchema()	Retorna o esquema XML para a representação XML dos dados armazenados no DataSet.
HasChanges()	Obtém um valor que indica se o DataSet tem alterações, incluindo linhas novas, excluídas ou modificadas.
HasChanges(DataRowState)	Obtém um valor que indica se o DataSet tem alterações, incluindo linhas novas, excluídas ou modificadas, filtradas por DataRowState.
InferXmlSchema(Stream, String[])	Aplica o esquema XML do Stream especificado ao DataSet.
InferXmlSchema(String, String[])	Aplica o esquema XML do arquivo especificado para o DataSet.
InferXmlSchema(TextReader, String[])	Aplica o esquema XML especificado TextReader para o DataSet.
InferXmlSchema(XmlReader, String[])	Aplica o esquema XML especificado XmlReader para o DataSet.
InitializeDerivedDataSet()	Essa API dá suporte à infraestrutura do produto e não se destina a ser usada diretamente no seu código. Desserializa todos os dados de tabelas do conjunto de dados do fluxo XML ou binário.
IsBinarySerialized(SerializationInfo, StreamingContext)	Verifica o formato da representação serializada do DataSet.

(Continua)

(Continuação)

Quadro 2. Métodos do DataSet.

Nome	Descrição
Load(IDataReader, LoadOption, DataTable[])	Preenche um DataSet com valores de uma fonte de dados usando o IDataReader fornecido, usando uma matriz de instâncias DataTable para fornecer o esquema e as informações de namespace.
Load(IDataReader, LoadOption, FillErrorHandler, DataTable[])	Preenche um DataSet com valores de uma fonte de dados usando o IDataReader fornecido, usando uma matriz de instâncias DataTable para fornecer o esquema e as informações de namespace.
Load(IDataReader, LoadOption, String[])	Preenche um DataSet com valores de uma fonte de dados usando o IDataReader, usando uma matriz de cadeias de caracteres para fornecer os nomes das tabelas dentro de DataSet.
MemberwiseClone()	Cria uma cópia superficial do Object atual (herdado de Object).
Merge(DataRow[])	Mescla uma matriz de objetos DataRow no DataSet atual.
Merge(DataRow[], Boolean, MissingSchemaAction)	Mescla uma matriz de objetos DataRow no atual DataSet, preservando ou descartando as alterações no DataSet e tratando um esquema incompatível de acordo com os argumentos determinados.
Merge(DataSet)	Mescla um DataSet especificado e seu esquema no DataSet atual.
Merge(DataSet, Boolean)	Mescla um DataSet especificado e seu esquema no DataSet atual, preservando ou descartando as alterações no DataSet de acordo com o argumento determinado.
Merge(DataSet, Boolean, MissingSchemaAction)	Mescla um DataSet especificado e seu esquema com o atual DataSet, preservando ou descartando alterações no DataSet atual e manipulando um esquema incompatível de acordo com os argumentos fornecidos.
Merge(DataTable)	Mescla um DataTable especificado e seu esquema no DataSet atual.

(Continua)

(Continuação)

Quadro 2. Métodos do DataSet.

Nome	Descrição
Merge(DataTable, Boolean, MissingSchemaAction)	Mescla um DataTable especificado e seu esquema com o atual DataSet, preservando ou descartando alterações no DataSet e manipulando um esquema incompatível de acordo com os argumentos fornecidos.
OnPropertyChanging(PropertyChangingEventArgs)	Gera o OnPropertyChanging evento.
OnRemoveRelation(DataRelation)	Ocorre quando um objeto DataRelation é removido de um DataTable.
OnRemoveTable(DataTable)	Ocorre quando um DataTable é removido de um DataSet.
RaisePropertyChanging(String)	Envia uma notificação de que a propriedade DataSet especificada está prestes a se alterar.
ReadXml(Stream)	Lê o esquema XML e os dados para o DataSet usando o System.IO.Stream especificado.
ReadXml(Stream, XmlReadMode)	Lê o esquema XML e os dados no DataSet usando o System.IO.Stream e o XmlReadMode especificados.
ReadXml(String)	Lê o esquema XML e os dados no DataSet usando o arquivo especificado.
ReadXml(String, XmlReadMode)	Lê os dados e o esquema XML no DataSet usando o arquivo especificado e o XmlReadMode.
ReadXml(TextReader)	Lê o esquema XML e os dados para o DataSet usando o System.IO.TextReader especificado.
ReadXml(TextReader, XmlReadMode)	Lê o esquema XML e os dados para o DataSet usando o System.IO.TextReader e o XmlReadMode especificados.
ReadXml(XmlReader)	Lê o esquema XML e os dados para o DataSet usando o System.Xml.XmlReader especificado.
ReadXml(XmlReader, XmlReadMode)	Lê o esquema XML e os dados no DataSet usando o System.Xml.XmlReader e o XmlReadMode especificados.

(Continua)

(Continuação)

Quadro 2. Métodos do DataSet.

Nome	Descrição
ReadXmlSchema(Stream)	Lê o esquema XML do Stream especificado no DataSet.
ReadXmlSchema(String)	Lê o esquema XML do arquivo especificado no DataSet.
ReadXmlSchema(TextReader)	Lê o esquema XML para o DataSet usando o TextReader especificado.
ReadXmlSchema(XmlReader)	Lê o esquema XML do XmlReader especificado no DataSet.
ReadXmlSerializable(XmlReader)	Essa API dá suporte à infraestrutura do produto e não se destina a ser usada diretamente no seu código. Ignora os atributos e retorna um conjunto de dados vazio.
RejectChanges()	Reverte todas as alterações feitas ao DataSet desde que foi criado ou desde a última vez em que o DataSet.AcceptChanges foi chamado.
Reset()	Limpa todas as tabelas e remove todas as relações, restrições externas e tabelas de DataSet. As subclasses devem substituir Reset para restaurar um DataSet ao seu estado original.
Reset()	Limpa todas as tabelas e remove todas as relações, restrições externas e tabelas de DataSet. As subclasses devem substituir Reset para restaurar um DataSet ao seu estado original.
ShouldSerializeRelations()	Obtém um valor indicando se a propriedade Relations deve ser persistida.
ShouldSerializeTables()	Obtém um valor indicando se a propriedade Tables deve ser persistida.
ToString()	Retorna um String que contém o nome do componente, se houver. Esse método não deve ser substituído (herdado de MarshalByValueComponent).
WriteXml(Stream)	Grava os dados atuais para o DataSet usando o System.IO.Stream especificado.

(Continua)

(Continuação)

Quadro 2. Métodos do DataSet.

Nome	Descrição
WriteXml(Stream, XmlWriteMode)	Grava os dados atuais e, opcionalmente, o esquema para o DataSet usando o System.IO.Stream e o XmlWriteMode especificados. Para gravar o esquema, defina o valor do parâmetro mode como WriteSchema.
WriteXml(String)	Grava os dados atuais do DataSet no arquivo especificado.
WriteXml(String, XmlWriteMode)	Grava os dados atuais e, opcionalmente, o esquema do DataSet no arquivo especificado usando o XmlWriteMode especificado. Para gravar o esquema, defina o valor do parâmetro mode como WriteSchema.
WriteXml(TextWriter)	Grava os dados atuais para o DataSet usando o TextWriter especificado.
WriteXml(TextWriter, XmlWriteMode)	Grava os dados atuais e, opcionalmente, o esquema do DataSet usando o TextWriter e o XmlWriteMode especificados. Para gravar o esquema, defina o valor do parâmetro mode como WriteSchema.
WriteXml(XmlWriter)	Grava os dados atuais do DataSet no XmlWriter especificado.
WriteXml(XmlWriter, XmlWriteMode)	Grava os dados atuais e, opcionalmente, o esquema para o DataSet usando o XmlWriter e o XmlWriteMode especificados. Para gravar o esquema, defina o valor do parâmetro mode como WriteSchema.
WriteXmlSchema(Stream)	Grava a estrutura DataSet como um esquema XML para o objeto System.IO.Stream especificado.
WriteXmlSchema(Stream, Converter<Type, String>)	Grava a estrutura DataSet como um esquema XML para o objeto System.IO.Stream especificado.
WriteXmlSchema(String)	Grava a estrutura DataSet como um esquema XML para um arquivo.

(Continua)

(Continuação)

Quadro 2. Métodos do DataSet.

Nome	Descrição
WriteXmlSchema(String, Converter<Type, String>)	Grava a estrutura DataSet como um esquema XML para um arquivo.
WriteXmlSchema(TextWriter)	Grava a estrutura DataSet como um esquema XML para o objeto TextWriter especificado.
WriteXmlSchema(TextWriter, Converter<Type, String>)	Grava a estrutura DataSet como um esquema XML para o TextWriter especificado.
WriteXmlSchema(XmlWriter)	Grava a estrutura DataSet como um esquema XML em um objeto XmlWriter.
WriteXmlSchema(XmlWriter, Converter<Type, String>)	Grava a estrutura DataSet como um esquema XML para o XmlWriter especificado.

TableAdapter

TableAdapters são componentes gerados por designer que se conectam a um banco de dados, executam consultas ou procedimentos armazenados e preenchem seus DataTables com os dados retornados. TableAdapters também enviam dados atualizados do seu aplicativo de volta para o banco de dados.



Fique atento

Você pode executar quantas consultas quiser em um TableAdapter, desde que elas retornem dados que estejam de acordo com o esquema da tabela com a qual o TableAdapter está associado.

Enquanto os TableAdapters são projetados com o Dataset Designer de conjunto de dados, as classes TableAdapter não são geradas como classes aninhadas do DataSet. Elas estão localizadas em namespaces separados que são específicos para cada conjunto de dados (MICROSOFT, c2018b).

Propriedades do componente

Quadro 3. Propriedades dos TableAdapters.

Nome	Descrição
CaseSensitive	Indica se a comparação de cadeias de caracteres dentro da tabela diferencia maiúsculas de minúsculas.
ChildRelations	Obtém a coleção das relações filhas do DataTable.
Columns	Obtém a coleção de colunas que pertencem à tabela.
Constraints	Obtém a coleção de restrições mantidas pela tabela.
Container	Obtém o contêiner para o componente (herdado de MarshalByValueComponent).
DataSet	Obtém o DataSet ao qual a tabela pertence.
DefaultView	Obtém uma exibição personalizada da tabela que pode incluir uma exibição filtrada ou uma posição do cursor.
DesignMode	Obtém um valor que indica se o componente está atualmente no modo de design (herdado de MarshalByValueComponent).
DisplayExpression	Obtém ou define a expressão que retorna um valor usado para representar essa tabela na interface do usuário. A propriedade DisplayExpression permite exibir o nome da tabela em uma interface do usuário.
Events	Obtém a lista de manipuladores de eventos que estão anexados ao componente (herdado de MarshalByValueComponent).
ExtendedProperties	Obtém a coleção de informações de usuário personalizadas.
HasErrors	Obtém um valor que indica se há erros em alguma das linhas em uma das tabelas de DataSet a que a tabela pertence.

(Continua)

*(Continuação)***Quadro 3.** Propriedades dos TableAdapters.

Nome	Descrição
IsInitialized	Obtém um valor que indica se o DataTable é inicializado.
Locale	Obtém ou define as informações de localidade usadas para comparar cadeias de caracteres na tabela.
MinimumCapacity	Obtém ou define o tamanho inicial para a tabela.
Namespace	Obtém ou define o namespace da representação XML dos dados armazenados no DataTable.
ParentRelations	Obtém a coleção de relações de pai da DataTable.
Prefix	Obtém ou define o namespace da representação XML dos dados armazenados no DataTable.
PrimaryKey	Obtém ou define uma matriz de colunas que funcionam como chaves primárias da tabela de dados.
RemotingFormat	Obtém ou define o formato de serialização.
Rows	Obtém a coleção de linhas que pertencem a essa tabela.
Site	Obtém ou define um System.ComponentModel.ISite para o DataTable (substitui o MarshalByValueComponent.Site).
TableName	Obtém ou define o nome do DataTable.

Métodos do componente

Quadro 4. Métodos dos TableAdapters.

Nome	Descrição
AcceptChanges()	Confirma todas as alterações feitas na tabela desde a última vez em que AcceptChanges foi chamado.
BeginInit()	Começa a inicialização de um DataTable usado em um formulário ou por outro componente. A inicialização ocorre no tempo de execução.
BeginLoadData()	Desativa as notificações, a manutenção de índice e as restrições durante o carregamento de dados.
Clear()	Limpa o DataTable de todos os dados.
Clone()	Clona a estrutura do DataTable, incluindo todos os esquemas e restrições de DataTable.
Compute(String, String)	Calcula a expressão especificada nas linhas atuais que passam os critérios de filtro.
Copy()	Copia a estrutura e os dados para o DataTable.
CreateDataReader()	Retorna um DataTableReader correspondente aos dados nesse DataTable.
CreateInstance()	Essa API dá suporte à infraestrutura do produto e não se destina a ser usada diretamente no seu código. Cria uma nova instância de DataTable.
Dispose()	Libera todos os recursos usados pelo MarshalByValueComponent (herdado de MarshalByValueComponent).
Dispose(Boolean)	Libera os recursos não gerenciados usados pelo MarshalByValueComponent e opcionalmente libera os recursos gerenciados (herdado de MarshalByValueComponent).
EndInit()	Encerra a inicialização de um DataTable que é usado em um formulário ou por outro componente. A inicialização ocorre no tempo de execução.

(Continua)

(Continuação)

Quadro 4. Métodos dos TableAdapters.

Nome	Descrição
EndLoadData()	Desativa as notificações, a manutenção de índice e as restrições após o carregamento de dados.
Equals(Object)	Verifica se o objeto especificado é igual ao objeto atual (herdado de Object).
Finalize()	Permite que um objeto tente liberar recursos e executar outras operações de limpeza antes que ele seja recuperado pela coleta de lixo (herdado de MarshalByValueComponent).
GetChanges()	Obtém uma cópia do DataTable que contém todas as alterações feitas nele desde que ele foi carregado ou que o AcceptChanges foi chamado pela última vez.
GetChanges(DataRowState)	Obtém uma cópia do DataTable que contém todas as alterações feitas desde o último carregamento ou desde que AcceptChanges foi chamado, filtrado por DataRowState.
GetDataTableSchema(XmlSchemaSet)	Retorna uma instância XmlSchemaSet que contém a linguagem WSDL que descreve o DataTable para os Serviços Web.
GetErrors()	Obtém uma matriz de objetos DataRow que contém erros.
GetHashCode()	Serve como a função de hash padrão (herdado de Object).
GetObjectData(SerializationInfo, StreamingContext)	Povoa um objeto de informações de serialização com os dados necessários para serializar o DataTable.
GetRowType()	Essa API dá suporte à infraestrutura produto e não se destina a ser usada diretamente no seu código. Obtém o tipo de linha.
GetSchema()	Essa API dá suporte à infraestrutura do produto e não se destina a ser usada diretamente no seu código. Para obter uma descrição desse membro, consulte <code>IXmlSerializable.GetSchema</code> .

(Continua)

(Continuação)

Quadro 4. Métodos dos TableAdapters.

Nome	Descrição
GetService(Type)	Obtém o implementador do IServiceProvider (herdado de MarshalByValueComponent).
GetType()	Obtém o Type da instância atual (herdado de Object).
ImportRow(DataRow)	Copia um DataRow em um DataTable, preservando quaisquer configurações de propriedade, bem como os valores originais e atuais.
Load(IDataReader)	Preenche um DataTable com valores de uma fonte de dados usando o IDataReader fornecido. Se o DataTable já contiver linhas, os dados de entrada da fonte de dados serão mesclados com as linhas existentes.
Load(IDataReader, LoadOption)	Preenche um DataTable com valores de uma fonte de dados usando o IDataReader fornecido. Se o DataTable já contiver linhas, os dados de entrada da fonte de dados serão mesclados com as linhas existentes de acordo com o valor do parâmetro loadOption.
Load(IDataReader, LoadOption, FillErrorEventHandler)	Preenche um DataTable com valores de uma fonte de dados usando o IDataReader fornecido como um delegado de tratamento de erro.
LoadDataRow(Object[], Boolean)	Localiza e atualiza uma linha específica. Se nenhuma linha correspondente for encontrada, uma nova linha será criada usando os valores fornecidos.
LoadDataRow(Object[], LoadOption)	Localiza e atualiza uma linha específica. Se nenhuma linha correspondente for encontrada, uma nova linha será criada usando os valores fornecidos.
MemberwiseClone()	Cria uma cópia superficial do Object atual (herdado de Object).
Merge(DataTable)	Mescla o DataTable especificado ao DataTable atual.

(Continua)

(Continuação)

Quadro 4. Métodos dos TableAdapters.

Nome	Descrição
Merge(DataTable, Boolean)	Mescla o DataTable especificado com o DataTable atual, indicando se as alterações no DataTable atual devem ser preservadas.
Merge(DataTable, Boolean, MissingSchemaAction)	Mescla o DataTable especificado com o DataTable atual, indicando se as alterações e a maneira de manipular o esquema ausente no DataTable atual devem ser preservadas.
NewRow()	Cria um novo DataRow com o mesmo esquema que a tabela.
NewRowArray(Int32)	Essa API dá suporte à infraestrutura do produto e não se destina a ser usada diretamente no seu código. Retorna uma matriz de DataRow.
NewRowFromBuilder(DataRowBuilder)	Cria uma nova linha de uma linha existente.
OnColumnChanged(DataColumnChangeEventArgs)	Gera o evento ColumnChanged.
OnColumnChanging(DataColumnChangeEventArgs)	Gera o evento ColumnChanging.
OnPropertyChanging(PropertyChangedEventArgs)	Gera o evento PropertyChanged.
OnRemoveColumn(DataColumn)	Notifica o DataTable que um DataColumn está sendo removido.
OnRowChanged(DataRowChangeEventArgs)	Gera o evento RowChanged.
OnRowChanging(DataRowChangeEventArgs)	Gera o evento RowChanging.
OnRowDeleted(DataRowChangeEventArgs)	Gera o evento RowDeleted.
OnRowDeleting(DataRowChangeEventArgs)	Gera o evento RowDeleting.
OnTableCleared(DataTableClearEventArgs)	Gera o evento TableCleared.

(Continua)

(Continuação)

Quadro 4. Métodos dos TableAdapters.

Nome	Descrição
OnTableClearing(DataTableClearEventArgs)	Gera o evento TableClearing.
OnTableNewRow(DataTableNewRowEventArgs)	Gera o evento TableNewRow.
ReadXml(Stream)	Lê o esquema XML e os dados para o DataTable usando o Stream especificado.
ReadXml(String)	Lê o esquema XML e os dados para o DataTable do arquivo especificado.
ReadXml(TextReader)	Lê o esquema XML e os dados para o DataTable usando o TextReader especificado.
ReadXml(XmlReader)	Lê o esquema XML e os dados no DataTable usando o XmlReader especificado.
ReadXmlSchema(Stream)	Lê um esquema XML para o DataTable usando o fluxo especificado.
ReadXmlSchema(String)	Lê um esquema XML para o DataTable do arquivo especificado.
ReadXmlSchema(TextReader)	Lê um esquema XML para o DataTable usando o TextReader especificado.
ReadXmlSchema(XmlReader)	Lê um esquema XML para o DataTable usando o XmlReader especificado.
ReadXmlSerializable(XmlReader)	Essa API dá suporte à infraestrutura do produto e não se destina a ser usada diretamente no seu código. Lê de um fluxo XML.
RejectChanges()	Reverte todas as alterações feitas na tabela desde que foi carregado ou desde a última vez em que AcceptChanges foi chamado.
Reset()	Redefine o DataTable com o estado original. A redefinição remove todos os dados, índices, relações e colunas da tabela. Se um DataSet incluir um DataTable, a tabela ainda fará parte do DataSet depois que for redefinida.
Select()	Obtém uma matriz de todos os objetos DataRow.

(Continua)

(Continuação)

Quadro 4. Métodos dos TableAdapters.

Nome	Descrição
Select(String)	Obtém uma matriz de todos os objetos DataRow que correspondem aos critérios de filtro.
Select(String, String)	Obtém uma matriz de todos os objetos DataRow que correspondem aos critérios de filtro, na ordem de classificação especificada.
Select(String, String, DataRowViewRowState)	Obtém uma matriz de todos os objetos DataRow que correspondem ao filtro, na ordem de classificação correspondente ao estado especificado.
ToString()	Obtém o TableName e a DisplayExpression, se houver algum como uma cadeia de caracteres concatenada (substitui o MarshalByValueComponent.ToString()).

Relatório com relacionamento entre tabelas

Criar um relatório quando se têm informações em tabelas distintas é uma tarefa que necessita de algumas ações adicionais se comparada com a tarefa de criar um relatório simples de apenas uma tabela. Você precisa, por meio de um TableAdapter, criar uma nova tabela no banco de dados que irá exibir as informações do relacionamento entre as diferentes tabelas. A vantagem é que não precisa realizar consultas SQL no banco com diversas cláusulas JOIN, o que aumenta sua produtividade e otimiza seu tempo (MICROSOFT, c2018c).

Veja a seguir os procedimentos para criar um relatório com o apoio de um TableAdapter e do componente DataSet.

Primeiramente, você deve clicar duas vezes sobre o DataSet em que irá trabalhar. Assim, irá abrir o editor de modelo, como mostra a Figura 1. Nele, além de criar a nova tabela, você pode também modificar as já existentes.

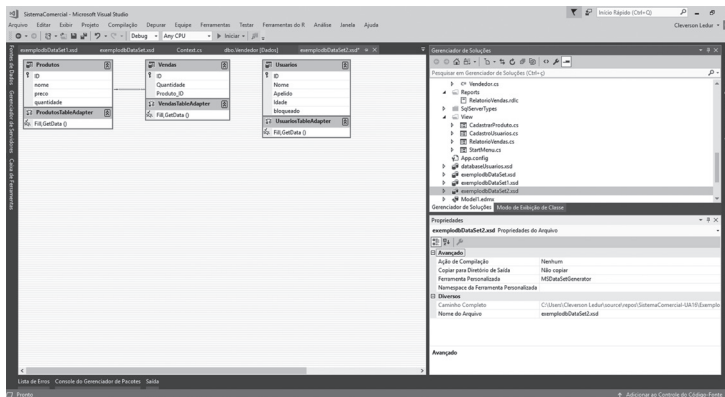


Figura 1. Editor de modelo.

Então, na **Caixa de Ferramentas**, você pode arrastar o componente **TableAdapter** para o seu **DataSet**, como na Figura 2.

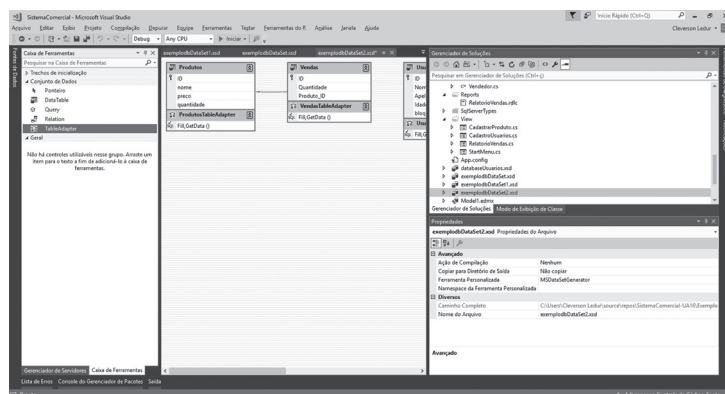


Figura 2. Caixa de Ferramentas.

Assim que você adiciona o componente no **DataSet**, a janela do **Assistente de Configuração** do **TableAdapter** é aberta. Você pode, então, configurar a conexão com o banco de dados. Se você está utilizando uma aplicação com o **Entity Framework** configurado e sendo utilizado, então a conexão já aparecerá. Basta clicar em **Avançar** (Figura 3).

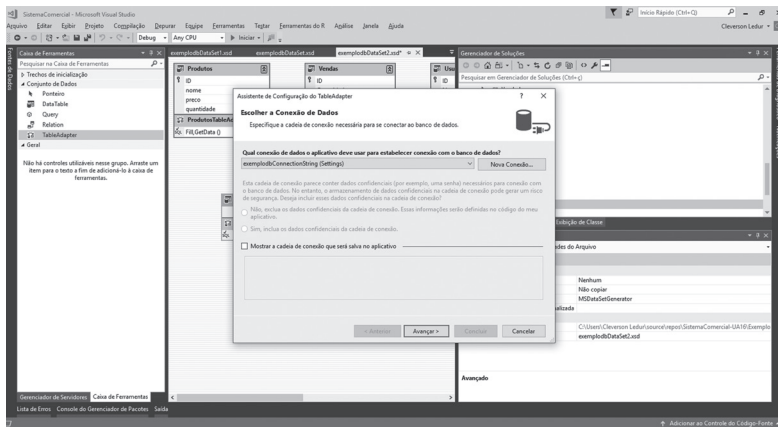


Figura 3. Você deve configurar a conexão com o banco de dados e clicar em **Avançar**.

Observe a Figura 4, a seguir. Você deve escolher a primeira opção apresentada, ou seja, **Usar instruções SQL**. Dessa forma, vai realizar uma operação do tipo Join entre as tabelas do seu banco.

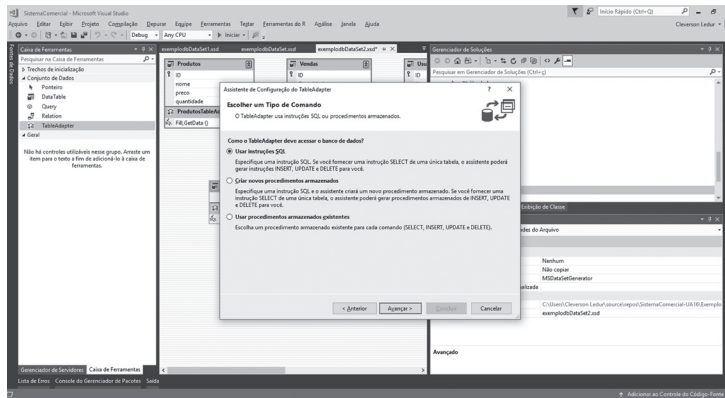


Figura 4. Selecione a primeira opção apresentada.

Clique em **Avançar** e então será aberta uma janela de inserção da instrução SQL (Figura 5). Nesse momento, você deve clicar em **Construtor de Consultas** para utilizar a ferramenta do Visual Studio.

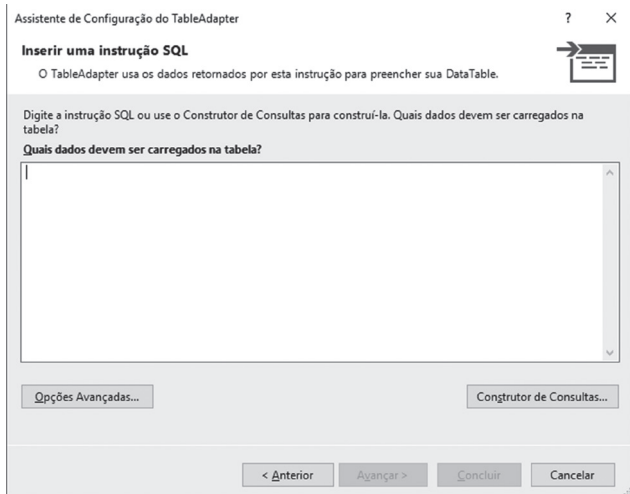


Figura 5. Clique em **Construtor de Consultas**.

Então, selecione as tabelas que deseja utilizar nessa consulta (Figura 6). Você pode selecionar mais de uma tabela segurando a tecla **Ctrl** e clicando nos nomes das tabelas.

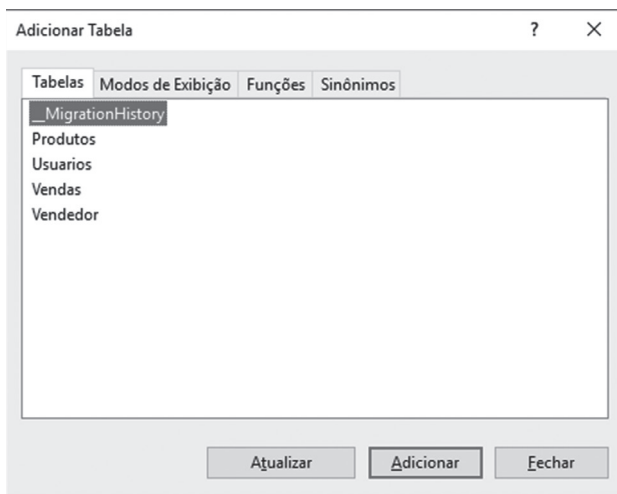


Figura 6. Selecione as tabelas.

Logo que você clicar em **Adicionar**, a janela **Construtor de Consultas** exibirá os dados de cada tabela. Nesse momento, você deve selecionar todos os dados que deseja utilizar no relatório, como mostra a Figura 7. Ao final, clique em **OK**.

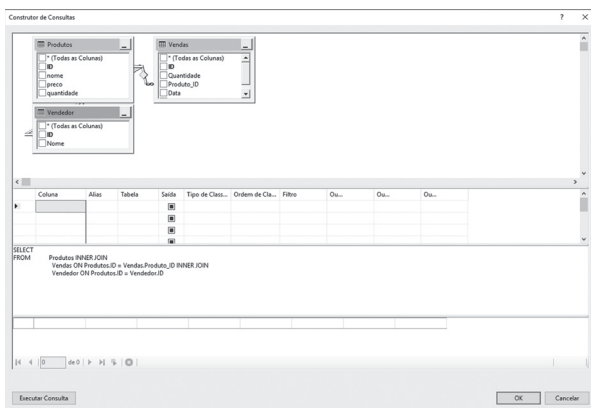


Figura 7. Selecione todos os dados que deseja utilizar.

Então, você deve clicar em **Concluir** na janela inicial do assistente (Figura 8).

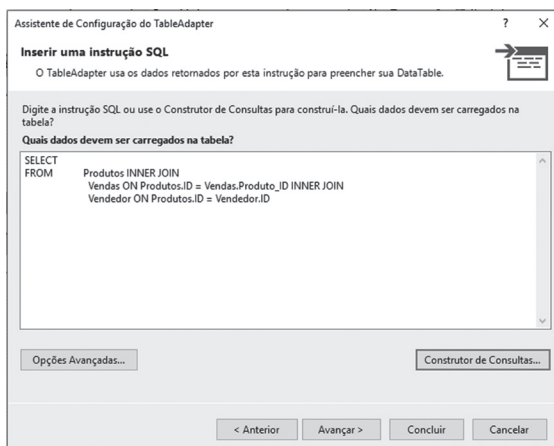


Figura 8. Clique em **Concluir**.

Logo após você clicar em **Concluir**, poderá notar a presença de um novo **DataTable** (tabela) no **DataSet** (Figura 9). Você pode alterar o nome desse **DataTable** clicando duas vezes sobre ele.

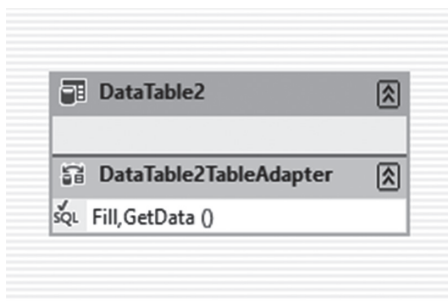


Figura 9. Você pode notar a presença de um novo **DataTable** no **DataSet**.

A partir desse momento, você pode proceder com as mesmas atividades necessárias para criar um relatório simples de apenas uma tabela. A única diferença é que você irá utilizar a tabela que criou com o **TableAdapter**.



Referências

MICROSOFT. *DataSet Class*. Redmond, c2018a. Disponível em: <[https://msdn.microsoft.com/en-us/library/system.data.dataset\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.data.dataset(v=vs.110).aspx)>. Acesso em: 18 jan. 2018.

MICROSOFT. *DataTable Class*. Redmond, c2018b. Disponível em: <[https://msdn.microsoft.com/en-us/library/system.data.datatable\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.data.datatable(v=vs.110).aspx)>. Acesso em: 18 jan. 2018.

MICROSOFT. *Fill datasets by using TableAdapters*. Redmond, c2018c. Disponível em: <<https://msdn.microsoft.com/en-us/library/ms171919.aspx>>. Acesso em: 18 jan. 2018.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.

Conteúdo:



SOLUÇÕES
EDUCACIONAIS
INTEGRADAS