

- [Canais](#)
- [Fóruns](#)
- [Multimídia](#)
- [Blogs](#)
- [Shop](#)
- [Eventos](#)
- [Pro](#)
- [Banco de Currículos](#)



## Por uma internet mais criativa e dinâmica

[Faça Login](#) ou [Cadastre-se](#)

E-mail:  Senha:   ☐ Manter conectado

[Esqueci a senha](#)

- [Arquitetura e Design](#)
  - [3ds max](#)
  - [Acessibilidade](#)
  - [Arquitetura da Informação](#)
  - [Fireworks](#)
  - [Flash](#)
  - [Lightwave 3D](#)
  - [Photoshop](#)
  - [Suite Corel](#)
  - [Teoria/Design](#)
  - [Usabilidade](#)
- [Banco de Dados](#)
  - [Access](#)
  - [DB2](#)
  - [Interbase](#)
  - [MySQL](#)
  - [Oracle](#)
  - [PostgreSQL](#)
  - [SQL Server](#)
- [Carreira](#)
  - [Certificações](#)
  - [Mercado](#)
  - [Tendências](#)
- [CMS e Framework](#)
  - [CakePHP](#)
  - [Django](#)
  - [Drupal](#)
  - [Joomla](#)
  - [Ruby on Rails](#)
  - [Sistemas de E-commerce](#)
  - [Wordpress](#)
  - [Zend Framework](#)
- [Desenvolvimento](#)
  - [Agile](#)
  - [Ajax](#)
  - [Aplicativos móveis](#)
  - [Dreamweaver](#)
  - [Flash](#)

- [Flex](#)
  - [Gerência de Projetos](#)
  - [SEO](#)
  - [Software Livre](#)
  - [Visual FoxPro](#)
  - [Visual Studio](#)
  - [Web Standards](#)
  - [WebServices](#)
- [Gerência de TI](#)
  - [B. Intelligence](#)
  - [Computação Forense](#)
  - [Direito e Web](#)
  - [E-Gov](#)
  - [Governança de TI](#)
  - [Mercado](#)
- [Linguagens](#)
  - [.NET](#)
    - [ADO.NET](#)
    - [ASP.NET](#)
    - [C#.NET](#)
    - [VB.NET](#)
  - [ActionScript](#)
  - [ASP](#)
  - [ColdFusion](#)
  - [CSS](#)
  - [Delphi](#)
  - [Java](#)
    - [Java para Mobile](#)
    - [Java para web](#)
  - [Javascript](#)
    - [Ajax](#)
    - [jQuery](#)
    - [Prototype](#)
  - [Perl](#)
  - [PHP](#)
  - [Ruby](#)
  - [UML](#)
  - [Visual Basic](#)
  - [XHTML](#)
  - [XML](#)
- [Mídia e Marketing Digital](#)
  - [E-commerce](#)
  - [E-Learning](#)
  - [E-mail Marketing](#)
  - [Mídia Social](#)
  - [Mobile Marketing](#)
  - [Publicidade Online](#)
  - [SEO](#)
  - [Web Analytics](#)
  - [Web Marketing](#)
  - [Web Writing](#)
- [Redes e Servidores](#)
  - [Apache](#)
  - [Cisco](#)
  - [Linux](#)
  - [Segurança](#)
  - [Windows Server](#)
- [Tecnologia](#)
  - [Gadgets](#)
  - [TV Digital](#)
  - [VoIP](#)

## **Desenvolvimento**

 [Feeds](#)  [Newsletter](#)

Quinta-feira, 26 de novembro de 2009 às 11h00

# NHibernate - usando o ActiveRecord - Parte 01



Conheça o curso ao vivo:  
**HTML5 na Prática**, com Paulino Michelazzo.

Apenas como introdução, vamos lembrar que o NHibernate é a versão para a plataforma .NET do framework de persistência Hibernate desenvolvido para a plataforma Java. Ele é um projeto open source e pode ser usado para gerar o mapeamento objeto relacional para diversos bancos de dados. Uma das suas principais vantagens é não ter que escrever as instruções SQL, pois todos os comandos para persistência são gerados em tempo de execução.

Atualmente a comunidade tem discutido muito sobre a utilização de ferramentas OR/M, ou seja, ferramentas que realizam o mapeamento objeto relacional. Existem dezenas de ferramentas que se propõem a realizar este serviço e, para a comunidade .NET, atualmente o **NHibernate** é uma das ferramentas mais usadas e mais maduras do mercado.

A Microsoft lançou recentemente o **Entity Framework** que tem o mesmo propósito, mas a ferramenta está evoluindo e ainda não atingiu o nível de maturidade que os desenvolvedores estão esperando. (A próxima versão da plataforma .NET trará uma versão mais madura da ferramenta, vamos aguardar...)

Embora o NHibernate seja uma ferramenta consagrada e muito usada, ela tem um problema: dá um trabalho danado gerar os arquivos de mapeamento XML na mão.

É aqui que entra o ActiveRecord, um projeto Castle que é uma implementação do padrão de projeto ActiveRecord, que consiste em instanciar propriedades que representam um registro de um banco de dados. Com o ActiveRecord não há necessidade de configurar arquivos xml para criar os mapeamentos do Nhibernate.

O padrão active record (em inglês: active record pattern) é um padrão de projeto encontrado frequentemente em um software que guarda dados em banco de dados relacionais. Foi nomeado por Martin Fowler no seu livro Patterns of Enterprise Application Architecture.

Active Record é uma abordagem para acessar dados em um banco de dados. Uma tabela do banco de dados ou view é envolta em uma classe, desta maneira, uma instância é vinculada a uma única linha (registro) na tabela. Após a criação de um objeto, um novo registro é adicionado na tabela após salvar. Qualquer objeto carregado obtém informações do seu banco de dados; quando um objeto é atualizado, o registro correspondente na tabela também é atualizado. A classe wrapper implementa os métodos de acesso ou propriedades para cada coluna na tabela ou view.

Este padrão é comumente usado por ferramentas de persistência de objetos, e em mapeamento objeto-relacional. Normalmente relacionamentos de chave estrangeira serão expostos como um objeto de tipo apropriado por meio de uma propriedade.

O Castle ActiveRecord é uma implementação deste padrão e, neste artigo, eu vou mostrar como usar o ActiveRecord da Castle para gerar os arquivos de mapeamento XML necessários em uma aplicação Windows Forms que utiliza o NHibernate como ferramenta OR/M.

## Requisitos necessários

No exemplo prático que irei mostrar neste artigo vamos precisar das seguintes ferramentas:

### ActiveRecord

- Você deve baixar a última versão disponível do ActiveRecord neste link: <http://www.castleproject.org/castle/download.html>
- A documentação pode ser baixada aqui: <http://www.castleproject.org/activerecord/documentation/index.html>

### NHibernate

- Baixe a última versão do NHibernate aqui: <http://sourceforge.net/projects/nhibernate/>

Após fazer o download você deve descompactar os arquivos em uma pasta própria.

### SQL Server 2005

- Baixe o SQL Server 2005 Express Edition aqui: [Microsoft SQL Server 2005 Express Edition](#)

### Management Studio Express Edition

- Pegue o Management Studio aqui: [Microsoft SQL Server Management Studio Express](#)

Para poder usar os recursos do ActiveRecord e do NHibernate você terá que incluir as seguintes referências no projeto.

Para fazer isso clique sobre o nome do projeto e selecione Add Reference;

A seguir selecione Browse e localize as seguintes DLLs referente ao ActiveRecord selecionando-as:

**Castle.ActiveRecord.dll**  
**Castle.Core.dll**  
**Castle.Components.Validator.dll**  
**Castle.DynamicProxy.dll**

Repita o processo e agora selecione as DLLs referente ao NHibernate:

**NHibernate.dll**  
**Iesi.Collections.dll**  
**log4net.dll**

Após concluída esta etapa você deverá declarar o seguinte namespace na classe do seu projeto:

<b>C#</b>	<b>VB .NET</b>
<b>using Castle.ActiveRecord</b>	<b>Imports Castle.ActiveRecord</b>

O objetivo principal do artigo é mostrar como usar o ActiveRecord para gerar os arquivos de mapeamento XML, mas não vou detalhar este conceito aqui.

Com o ActiveRecord você não vai precisar aprender sobre o esquema de mapeamento do NHibernate para poder gerar os seus arquivos XML, nem vai ter que tratar com ISession e ISessionFactory, pois toda esta complexidade é tratada pelo ActiveRecord.

O ActiveRecord oferece um subconjunto das funcionalidades de mapeamento do NHibernate, sendo que o mapeamento é feito usando classes, campos e atributos a nível de propriedade (*O ActiveRecord esta apto a inferir os nomes da tabela e das colunas mesmo que forem omitidas*).

*Obs: O Ruby On Rails manuseia dados usando o **Active Record**, que faz o trabalho de ORM.*

Abaixo temos um exemplo de uma classe criada para gerar o mapeamento usando o ActiveRecord através de atributos:

```
using Castle.ActiveRecord;

[ActiveRecord]
public class Category : ActiveRecordBase
{
    private int id;
    private string name;
    private Category parent;
    private IList<Category> subcategories = new List<Category>();

    [PrimaryKey]
    public int Id
    {
        get { return id; }
        set { id = value; }
    }

    [Property]
    public string Name
    {
        get { return name; }
        set { name = value; }
    }

    [BelongsTo("parent_id")]
    public Category Parent
    {
        get { return parent; }
        set { parent = value; }
    }

    [HasMany]
    public IList<Category> SubCategories
    {
        get { return subcategories; }
        set { subcategories = value; }
    }
}
```

Neste exemplo, os atributos HasMany e BelongsTo podem ser usados em uma classe pai e filho, respectivamente, para indicar

uma relação um-para-muitos.

O ActiveRecord atua sobre o que é chamado de tipos ActiveRecord que são classes, que usam o atributo **ActiveRecordAttribute** e herdam a partir de uma das classes base ActiveRecord. Atualmente a classe base é a classe **ActiveRecordBase**, que é uma classe genérica.

O atributo ActiveRecord é usado para definir uma classe como um tipo ActiveRecord e para associar a informação do mapeamento. No exemplo acima vemos isso no trecho de código:

```
using Castle.ActiveRecord;

[ActiveRecord]
public class Category : ActiveRecordBase
{
    .....
```

Uma outra forma seria informar o nome da tabela:

```
using Castle.ActiveRecord;

[ActiveRecord("Category")]
public class Category : ActiveRecordBase
{
    .....
```

No exemplo acima não está informado de forma explícita o nome da tabela, nem o esquema do banco de dados. Neste caso o ActiveRecord, então, assume a classe Category como a classe que está sendo mapeada para a tabela do banco de dados com o mesmo nome. Neste caso o esquema do banco de dados será nulo.

Para informar o nome da tabela e do esquema do banco de dados, usamos as propriedades Table e Schema conforme abaixo:

```
using Castle.ActiveRecord;

[ActiveRecord(Table="Category", Schema="dbo")]
public class Category : ActiveRecordBase
{
    .....
```

Antes de utilizar o ActiveRecord em tempo de execução, você precisa inicializar de forma apropriada o Framework uma única vez para o tempo de vida da aplicação.

Afim de inicializar o framework, você precisa fornecer as seguintes informações:

- Qual o tipo de banco de dados você está usando;
- Como efetuar a conexão com este banco de dados;

Opcionalmente você também pode ativar o debug, o caching e efetuar outras configurações.

Antes de usar o ActiveRecord sua aplicação precisa invocar o método Initialize da classe ActiveRecordStarter para realizar a inicialização do framework.

O método Initialize precisa da implementação da interface IConfigurationSource e da definição dos tipos de ActiveRecord (que são as classes do domínio) a serem examinadas, dessa forma o NHibernate é configurado para examinar os tipos para erros de sintaxe e para construir o mapeamento para todos os tipos definidos.

A seguir, temos um exemplo de como efetuar tal tarefa para duas classes usando a linguagem C#:

```
using Castle.ActiveRecord;

IConfigurationSource config = ActiveRecordSectionHandler.Instance ;
ActiveRecordStarter.Initialize(config, typeof(Class1), typeof(Class2));
...
```

A classe ActiveRecordSectionHandler faz a leitura da configuração a partir de uma entrada para um 'activerecord' no arquivo xml associado com a AppDomain;

Outra forma de realizar a mesma tarefa é obter a informação da configuração de um arquivo AppConfig.xml;

```
imports Castle.ActiveRecord;

Dim source As XmlConfigurationSource =
    New XmlConfigurationSource("AppConfig.xml")
```

```
ActiveRecordStarter.Initialize(source,
GetType(Category))
```

```
.....
```

A classe que deve ser criada para realizar o mapeamento deve herdar da classe abstrata ActiveRecordBase que expõe membros públicos e protegidos que provavelmente você vai usar na sua classe ActiveRecord. Abaixo temos um exemplo de utilização da classe para gerar o mapeamento:

## VB .NET

```
Imports Castle.ActiveRecord

<ActiveRecord("Contatos")> _
Public Class Contato
    Inherits ActiveRecord(Of Contato)

    Private m_id As Integer

    <PrimaryKey("ID")> _
    Public Property Id() As Integer
    Get
        Return (m_id)
    End Get
    Set(ByVal value As Integer)
        m_id = value
    End Set
End Property
.....
```

A classe ActiveRecord também expõe instâncias de membros para Save, Create, Update e Delete sendo que a operação Save está apta a distinguir se a classe precisa ser criada ou atualizada.

Você pode consultar mais detalhes na documentação do ActiveRecord: <http://www.castleproject.org/activerecord/documentation/v1rc1/index.html>

## Usando o ActiveRecord e o NHibernate em uma aplicação Windows Forms

Para que você possa entender como implementar a utilização dos recursos do ActiveRecord em sua aplicação C# ou VB .NET, vou criar um exemplo usando a linguagem VB .NET no qual iremos realizar as seguintes tarefas:

- Criação do banco de dados e da tabela;
- Criação das classe para realizar o mapeamento;
- Definição do arquivo de configuração;
- Inicializando o Framework;
- Usando as classes na aplicação Windows Forms;

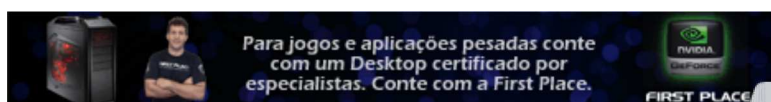
Aguarde na continuação deste artigo a criação do projeto completo usando NHibernate e ActiveRecord.

## Interação

- 1

Curtir

- [Relatar Link Quebrado](#) [Imprimir](#)



## 1 comentário



**Medieval**

25/07/2010 04h08

Aí vai também uma dica:

Use Mygeneration (pesquisa aí no google). Com ele você não precisa criar nem as classes (entities) e nem os mapeamentos xml. Você simplesmente manda ele gerar tudo isso lendo o banco.

É o que eu faço aqui, nem me lembro mais como monto esses mapeamentos no nhibernate..ehhehehehe.

[Responder](#)

[Cancelar resposta](#)

### Qual a sua opinião?

Se você já possui conta iMasters, o login será feito abaixo.

Nome:

E-mail:

Comentário:

Atenção: comentários considerados spams e/ou ofensivos serão moderados.

Patrocínio: 

**Sobre o Autor**



**José Carlos Macoratti** é referência em Visual Basic no Brasil e autor dos livros "Aprenda Rápido: ASP" e "ASP, ADO e Banco de Dados na Internet". Mantenedor do site [macoratti.net](http://macoratti.net).

[macoratti@yahoo.com](mailto:macoratti@yahoo.com)

**Outros artigos do mesmo autor:**

- [ASP .NET - Gerenciamento de serviços usando o...](#)
- [VB .NET - Criando gráficos no VB 2010 Express...](#)
- [VB .NET - Criando gráficos no VB 2010 Express...](#)
- [Crystal Reports para o Visual Studio 2010...](#)

[Ver mais artigos de José Carlos Macoratti](#)

[Indique para um amigo](#)



#### **Microsoft Visual Studio LightSwitch**

Desenvolva suas aplicações de uma maneira muito mais ágil e simples.



#### **RowFeeder**

Monitore e analise diversas palavras-chaves no Twitter e Facebook.



#### **Office 2010**

Versão de avaliação do pacote de escritório da Microsoft.



#### **VirtualBox**

Emule um sistema operacioanl dentro de outro.



#### **Google App Inventor**

Programa de criação de aplicativos para Android aberto para todos.



## Parceiros



- [\*\*iMasters\*\*](#)

- [Sobre o iMasters](#)
- [Política de privacidade](#)
- [Anuncie](#)
- [Feeds iMasters](#)
- [Fóruns iMasters](#)
- [Fale conosco](#)

2001 - iMasters FFPA Informática Ltda - Todos os direitos reservados.