

Licenciatura em Engenharia Informática e de Computadores

Trabalho Prático (Parte 2)

Projeto
de
Sistemas de Informação II
2019 / 2020 inverno

Pedro Tereso 44015
Tiago Pereira 43592
Wilson Costa 43593

Docente:
Eng.º Afonso Remédios

Turma: LI51N

INTRODUÇÃO	3
EF vs. ADO.NET	4
FACILIDADE DE PROGRAMAÇÃO E DESEMPENHO	4
GARANTIA DA CONSISTÊNCIA DE DADOS	6
CONCLUSÃO	7
ANEXOS	8

Introdução

O presente trabalho é sobre as diversas metodologias de acesso a dados residentes em bases de dados externas à máquina do utilizador.

Usando as tecnologias de sistemas de informação e de programação orientada a objectos e tomando como base o modelo desenvolvido na fase 1, criou-se duas aplicações distintas, ambas baseadas em C#, para a realização dos vários exercícios propostos.

As duas aplicações são baseadas em .NET mas usam implementações de acesso aos dados diferentes, uma usa ADO.NET e a outra Entity Framework.

O último ponto do trabalho prende-se com a comparação e análise de *performance* dos dois sistemas.

No final deverá ser claro as vantagens e desvantagens dos dois sistemas.

EF vs. ADO.NET

A. Facilidade de programação e desempenho

Segundo a nossa pesquisa, não é claro que uma abordagem seja melhor ou pior que a outra, cada caso é um caso e deverá ser tido em conta, o nível de abstração que o desenvolvedor quer ter bem como o nível de controlo que necessita para uma escrita correcta da linguagem usada.

EF, ou ADO.NET *Entity Framework*, resulta de uma adição de ferramentas ao ADO.NET o que teoricamente nos leva a acreditar que o EF é naturalmente mais lento que o ADO.NET mas será assim tão simples?

O EF, no seu trabalho de segundo-plano, encapsula o ADO.NET mas com o auxílio de LINQ e os outras ferramentas, auxilia o programador na manutenção do código e na facilidade de escrita de *querys* que de outra forma, tinham que ser escritas com um maior nível de detalhe aumentando assim o seu tamanho.

Já o ADO.NET deposita a responsabilidade de um escrita eficiente de código do lado do programa. O nível de abstração é bastante menor o que possibilita o acesso a propriedades mais nucleares.

ADO.NET	EF
<ul style="list-style-type: none"> • Fornece acesso consistente a fontes de dados como SQL Server e XML e a fontes de dados expostas por meio do OLE DB e ODBC. • As aplicações podem usar o ADO.NET para conectar-se a essas fontes de dados e inserir, apagar e atualizar os dados que eles contêm. 	<ul style="list-style-type: none"> • É um mapeador objecto-relacional (O/RM) para .NET com diversos anos de desenvolvimento e estabilização. • Permite aos desenvolvedores focarem-se na lógica de negócio da aplicação. • Elimina a necessidade de escrita de <i>querys</i> SQL repetidas

B. Garantia da consistência de dados

	ADO.NET	Entity Framework	LINQ
What is it?	.NET libraries for connecting to data sources and manipulating data.	An Object-Relational Mapper (ORM) which is built on ADO.NET classes.	.NET libraries for querying data across different data sources, also built on ADO.NET classes.
When is it released?	It is released with the .NET Framework 1.0 in 2002 .	Initially released with the .NET 3.5 SP 1 in 2008 .	Released as a major part of .NET 3.5 in 2007 .
Performance	Excels at the performance.	Earlier versions of Entity Framework is much slower compared to ADO.NET. However, the performance is always improved along with the new releases.	In most cases, it is slower compared to ADO.NET.
Development time	Developers need to write the data access layer code manually so that the development time takes more time compared to if we use the Entity Framework.	Has great tooling support to maintain data access in a much shorter time. It saves development time.	LINQ saves development time as it provides a consistent way for querying data, which is its main benefit.

Conclusão

Com o desenvolvimento deste trabalho prático conseguimos materializar a matéria lecionada durante as aulas teóricas.

Vistas, funções, procedimentos guardados e ainda gatilhos são os novos termos no nosso dicionário de capacidades em SQL-SERVER.

Também conseguimos exercitar as nossas capacidades de programação em C# com vista a um correto acesso aos dados

Servimo-nos também das ferramentas que adquirimos durante a disciplina introdutória (SI I).

O nível de isolamento usado foi *Read Committed*.

Anexos

