

02

Preparando o ambiente

Iniciando o projeto

Nosso objetivo nesse curso será implementar um projeto de controle financeiro integrando que terá uma interface web em que o usuário da aplicação poderá criar sua conta de usuário, cadastrar movimentações e montar relatórios utilizando as informações do banco de dados.

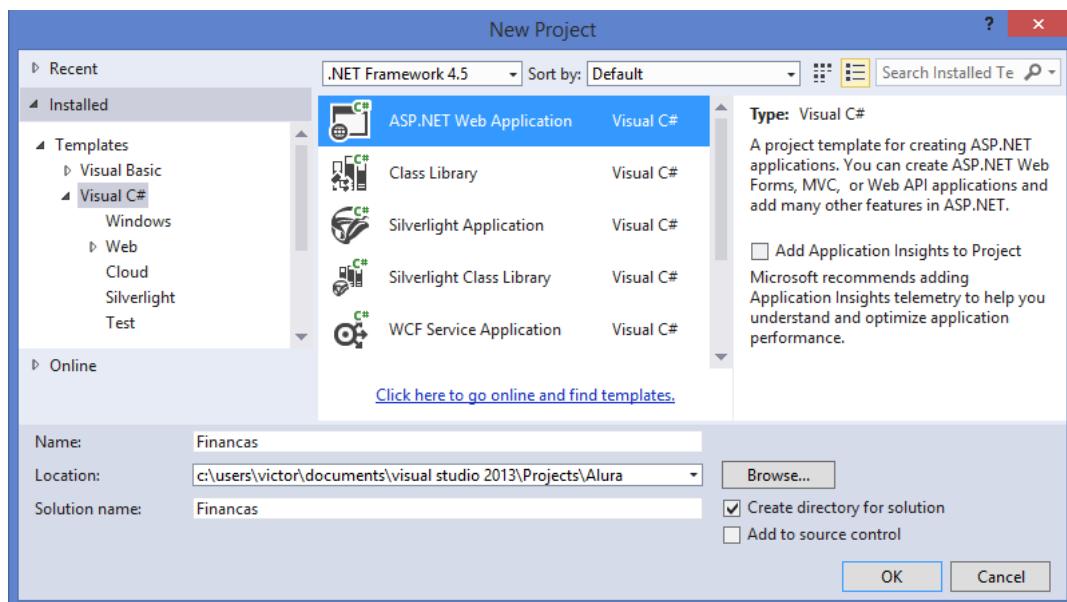
No controle financeiro, utilizaremos o Asp.Net MVC 5 com o Twitter bootstrap para criar a interface gráfica da aplicação. O acesso ao banco de dados utilizará o Entity Framework 6 e a integração da camada de acesso ao banco de dados com o controller do Asp.Net MVC será feita através da biblioteca de injeção de dependências Ninject.MVC3. Para a camada de segurança da aplicação, utilizaremos uma biblioteca da Microsoft chamada Simple Membership, que cuidará da autenticação e autorização de usuários.

O objetivo desse curso é mostrar como as tecnologias que foram estudadas na formação se integram dentro de uma aplicação mais complexa, além de mostrar como resolver alguns problemas comuns que são encontrados em projetos MVC e, por isso, esse é um curso para alunos que já fizeram os cursos anteriores da trilha C#.

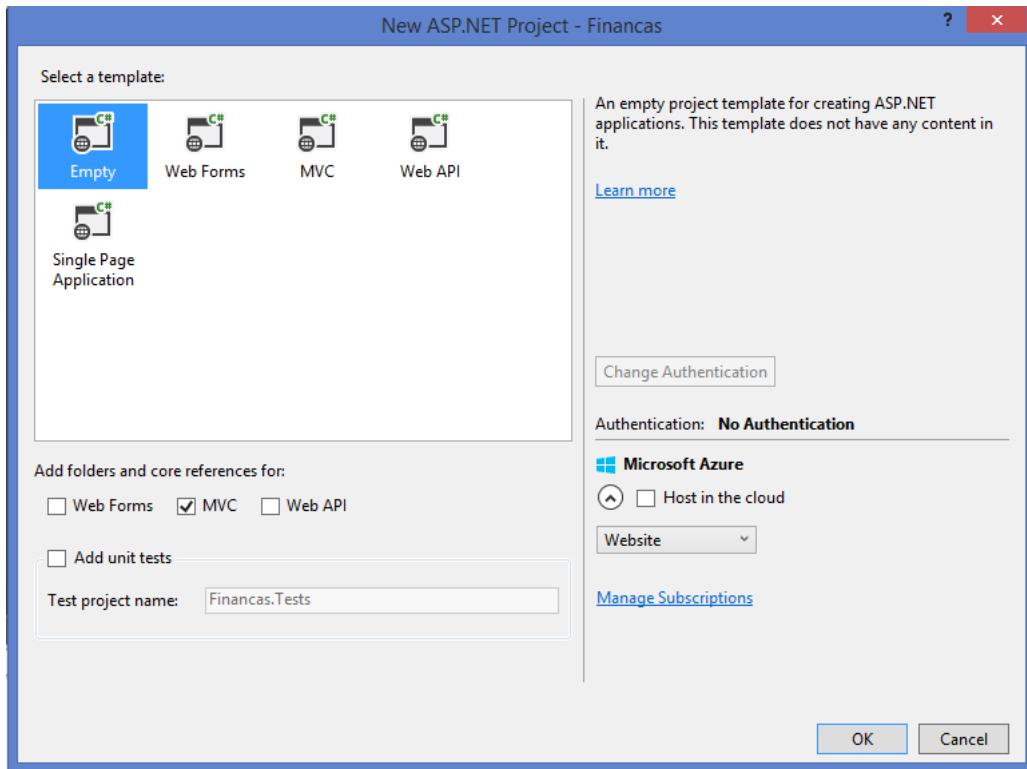
Criando o projeto no Visual Studio

Vamos então iniciar nosso novo projeto no Visual Studio 2013 Express for Web. Nesse curso utilizaremos o update 3 do Visual Studio que pode ser baixado nesse endereço: <http://go.microsoft.com/?linkid=9842996>.

Para criarmos o projeto, vamos novamente utilizar o comando `Ctrl + Shift + N` para abrir a janela do assistente de novo projeto. Dentro dessa janela, escolheremos a opção `Asp.Net Web Application`. O nome desse novo projeto será `Financas`:



Na próxima janela, escolheremos o projeto vazio (Empty) com as bibliotecas do Asp.Net MVC:



Agora que o projeto está criado, vamos configurar o banco de dados que será utilizado na aplicação. Clique com o botão direito na pasta `App_Data` e escolha a opção `Add > New Item` e dentro da janela de novo item, escolha a opção `Service-based Database`. Chame o banco de dados de `financas.mdf`.

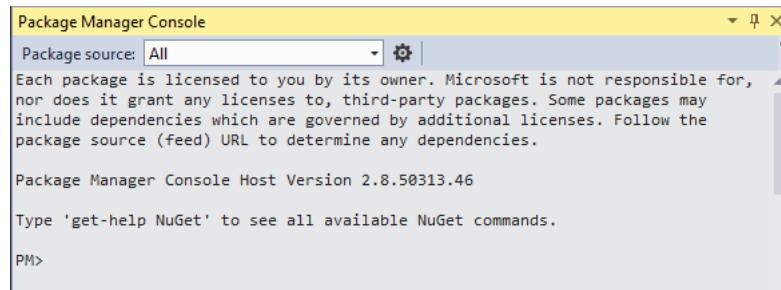
Agora precisamos configurar a string de conexão que será utilizada na aplicação. Para isso, abra o arquivo de configuração do projeto, o `Web.Config` (o que fica fora da pasta `Views`) e dentro da tag `configuration` declare a tag `connectionStrings`:

```
<configuration>
    <!-- outras configurações -->
    <connectionStrings>
        <add name="FinancasContext"
            providerName="System.Data.SqlClient"
            connectionString="Data Source=(LocalDB)\v11.0;
                AttachDbFilename=|DataDirectory|financas.mdf;
                Integrated Security=True"
        />
    </connectionStrings>
</configuration>
```

(Quem usar o Visual Studio 2015 deve substituir `Data Source=(LocalDB)\v11.0;` por `Data Source=(LocalDB)\MSSQLLocalDB;`)

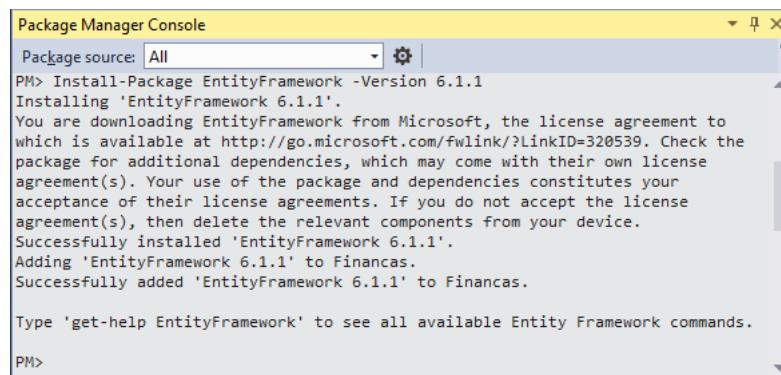
Agora que terminamos de configurar o banco de dados, vamos fazer a instalação do Entity Framework no projeto. Para fazer a instalação dessa biblioteca, assim como no curso de Entity Framework, nós utilizaremos o NuGet, porém para que todos utilizem a mesma versão do EntityFramework, faremos a instalação através do console do Nuget.

Dentro do Visual Studio, entre no menu `Tools > Nuget Package Manager > Package Manager Console`. Isso abrirá o `Console do Nuget` dentro do Visual Studio, o `Package Manager Console`:



No projeto utilizaremos o Entity Framework versão 6.1.1 que pode ser instalado utilizando o comando abaixo no Package Manager Console:

`Install-Package EntityFramework -Version 6.1.1`



Agora que o Entity Framework está instalado, vamos criar as entidades que serão utilizadas no projeto. Dentro do projeto crie uma nova pasta chamada `Entidades` e nela crie duas classes: `Movimentacao` e `Usuario`:

```

public class Usuario
{
    public int Id { get; set; }

    [Required]
    public string Nome { get; set; }

    [Required,EmailAddress]
    public string Email { get; set; }
}

public class Movimentacao
{
    public int Id { get; set; }

    public decimal Valor { get; set; }

    public DateTime Data { get; set; }

    public Tipo Tipo { get; set; }

    public int UsuarioId { get; set; }

    public virtual Usuario Usuario { get; set; }
}

public enum Tipo

```

```
{
    Entrada, Saída
}
```

Agora que criamos as entidades, vamos mapeá-las utilizando o Entity Framework, para isso, crie uma nova pasta no projeto chamada `DAO` e dentro dela, crie a classe `FinancasContext`:

```
public class FinancasContext : DbContext
{
    public DbSet<Usuario> Usuarios { get; set; }

    public DbSet<Movimentacao> Movimentacoes { get; set; }

    protected override void OnModelCreating(DbModelBuilder builder)
    {
        builder.Entity<Movimentacao>().HasRequired(m => m.Usuario);
    }
}
```

Para criar as tabelas que serão utilizadas pelo Entity Framework no banco de dados, criaremos uma nova migração. No Package Manager Console, digite o comando abaixo para habilitar migrações:

```
Enable-Migrations
```

E agora vamos criar a nova migração que criará as tabelas iniciais do banco de dados:

```
Add-Migration TabelasIniciais
```

Para executar a migração criada, utilizamos o seguinte comando:

```
Update-Database
```

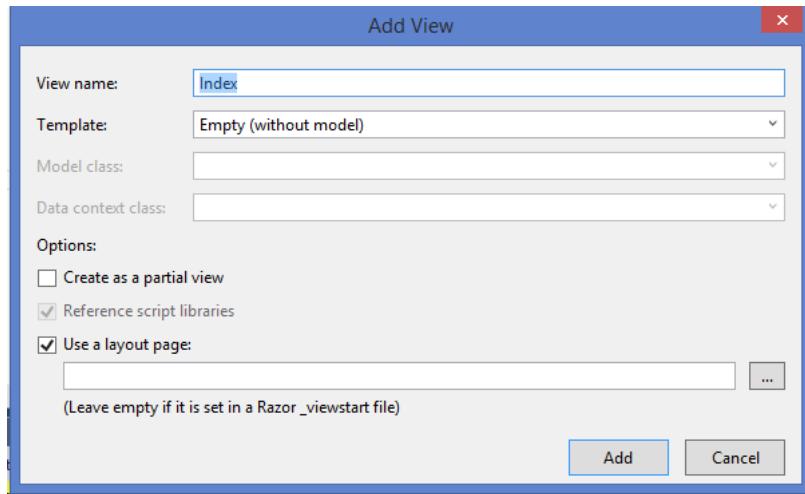
Com isso terminamos a configuração do Entity Framework e das entidades que serão gravadas no banco de dados.

Para terminarmos esse capítulo, vamos criar a página inicial para a aplicação, para isso, dentro da pasta `Controllers` do projeto, crie um novo controller chamado `HomeController` e dentro dele, vamos implementar o método `Index` que simplesmente envia o usuário para a camada de visualização:

```
public class HomeController : Controller
{
    public ActionResult Index()
    {
        return View();
    }
}
```

Agora vamos criar a view para essa action que acabamos de desenvolver. Clique com o botão direito no método `Index` que acabamos de criar e selecione a opção `Add View...` dentro da janela de nova view e dentro da janela `Add View`,

escolha deixe a opção Use a layout page marcada:

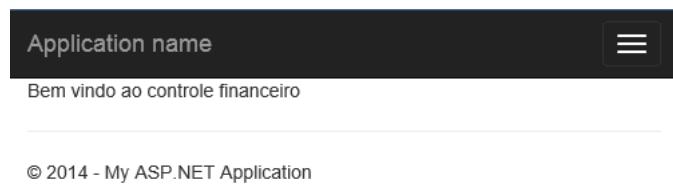


Com isso o assistente do Visual Studio criará a view e também já deixará preparado um layout page para a aplicação.

Dentro do código da view criada, vamos simplesmente adicionar a mensagem de boas vindas:

```
Bem vindo ao controle financeiro
```

Depois disso execute a aplicação para visualizar a página inicial:



A página inicial está pronta! No próximo capítulo implementaremos o layout da aplicação junto com a lógica de cadastro de usuários.