



Passo a passo:

Usando o Microsoft Azure para estudantes

Comece a usar o Azure através da nova oferta gratuita da Microsoft para estudantes, aprendendo como desenvolver e submeter soluções para a nuvem, sem qualquer custo!

Introdução

O Microsoft Azure para DreamSpark apresenta os serviços necessários para que você desenvolva aplicativos na nuvem sem custos:

- **Azure App Service Web Apps:** é parte de uma oferta de nuvem totalmente gerenciada que permite compilar e implantar aplicativos Web em segundos. Use HTML5, ASP.NET, Java, PHP, Node.js ou Python.
- **Banco de dados MySQL no ClearDB:** adiciona o poder do banco de dados MySQL a seus aplicativos Web. Com o MySQL no ClearDB, você pode implantar mais tipos de aplicativos Web e soluções CMS.
- **Application Insights:** oferece uma visão 360° sobre disponibilidade, desempenho e uso dos seus serviços ASP.NET.
- **Visual Studio Online:** é a forma mais rápida e fácil de planejar, criar e entregar software em uma variedade de plataformas, utilizando-se de nossa infraestrutura de nuvem.

Iremos desenvolver uma solução completa que basicamente será uma **Azure Web App** gerenciado pelo **Visual Studio Online**. Esta web app também utilizará um **banco de dados MySQL**, além de podermos checar suas estatísticas e dados de desempenho através de um **Application Insights** vinculado ao mesmo.

Há várias maneiras de se submeter suas soluções ao Azure. Tratando-se de web apps, por exemplo, você pode utilizar o Visual Studio, o [GitHub](#), um [cliente FTP](#), entre outras formas. Como dito anteriormente, iremos prosseguir utilizando a IDE do Visual Studio, devido a sua integração nativa com a nuvem da Microsoft.

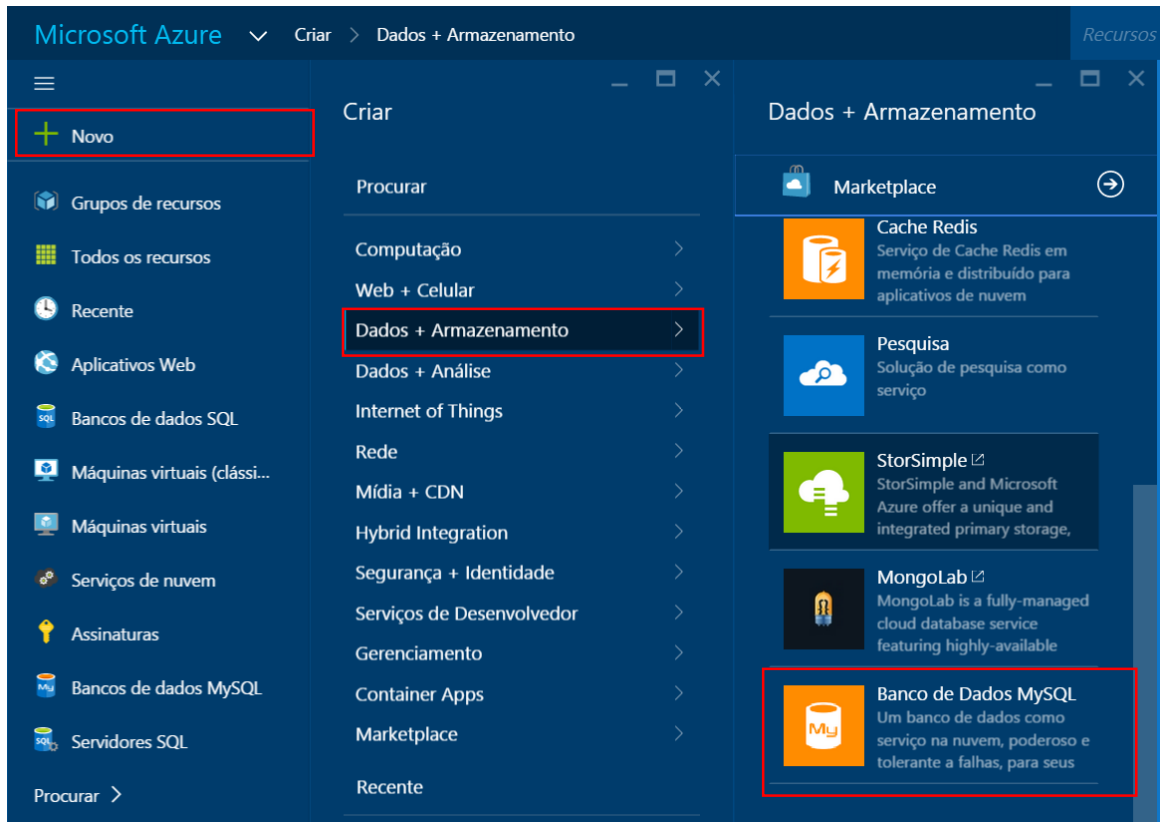
Atenção: Para utilizar a plataforma de nuvem da Microsoft para estudantes, primeiro você precisa criar [uma conta Dreamspark e confirmar sua situação de estudante](#). Em seguida, você precisará [registrar uma conta do Microsoft Azure via Dreamspark](#). Também é preciso, para este tutorial, que você tenha instalado em seu computador o [Visual Studio Community 2015](#) para submeter seu trabalho ao Azure, bem como o [MySQL Workbench](#) para manipularmos o bancos de dado associado a ele.

Vamos lá! Mãos à obra:

Passo I.	Criando Banco de dados MySQL no Azure.....	3
Passo II.	Criando tabelas e populando o banco de dados.....	6
Passo III.	Criando um projeto no Visual Studio Online (VSO)	10
Passo IV.	Desenvolvendo e publicando a web app	13
Passo V.	Utilizando o Application Insights.....	25

Passo I. Criando Banco de dados MySQL no Azure

1. Acesse <https://portal.azure.com/> e entre com suas credenciais. Após um login bem-sucedido, clique em **Novo**, depois em **Dados + Armazenamento** e escolha **Banco de Dados MySQL**.



Dentro do bloco que foi aberto logo a direita, escolha um **nome** qualquer (verifique se este nome é válido/não existente) para seu banco de dados. Confira se a assinatura listada é Dreamspark e o local do banco de dados está em **Brazil South**. Atente-se em clicar em **Termos Legais**, ler o texto indicado e clicar em OK.

Novo Banco de Dados MySQL
Cria um novo Banco de Dados MySQL

* nome do banco de dados
db_student_yourname ✓

* Assinatura
DreamSpark ▼

Database Type
Shared ▼

* Location
Brazil South ▼

* Grupo de recursos
Default

[Selecionar Existente](#)

* Tipo de Preço
Mercúrio >

* Termos Legais
Definir configurações necessárias >

☒ Fixar no Quadro inicial

Criar

Novo Banco de Dados MySQL
Cria um novo Banco de Dados MySQL

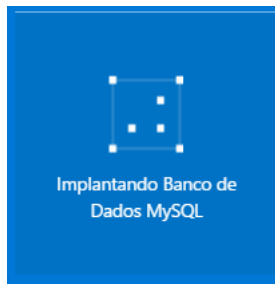
Ao clicar em "OK", eu (a) autorizo a Microsoft a fazer a cobrança mensal do valor indicado com o método de pagamento atual até meu serviço ser cancelado ou encerrado, (b) reconheço que a oferta é fornecida pela SuccessBricks, Inc., não pela Microsoft, e concordo com os termos de uso da SuccessBricks, Inc. [termos de uso](#) e sua [política de privacidade](#), e (c) concordo em compartilhar minhas informações de contato com a SuccessBricks, Inc.

OK

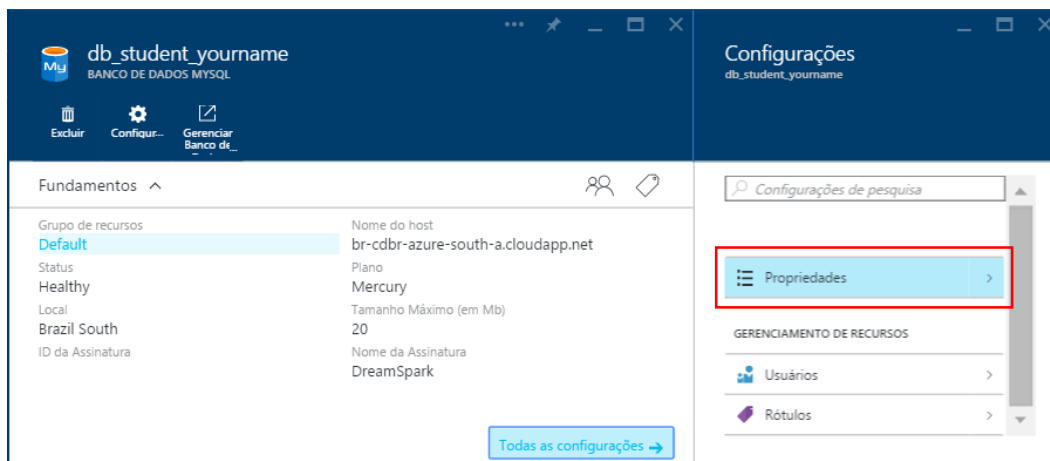
OBS: Os termos legais acima alertam para o preço definido no plano escolhido e que o mesmo será cobrado durante o uso do banco. Mas não se preocupe quanto a isso, o plano disponível que nós estamos utilizando é o **Mercúrio**, cujos detalhes, recursos e limitações podem ser visualizados abaixo:

Mercúrio	
4	Total de conexões
.02	GB
0.01	Nível de Entrada
	Multilocatário
	99.95% SLA
0,00	
USD/MÊS (ESTIMADO)	

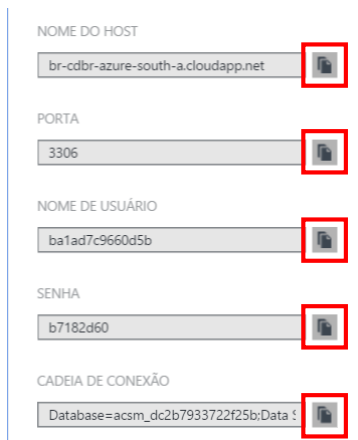
- Logo após esse procedimento, um novo item será adicionado à sua dashboard, representando o banco de dados MySQL.



Após alguns instantes, uma janela contendo os dados de seu recém-criado abrirá. Caso a mesma não apareça automaticamente, clique no ícone de seu banco na dashboard. Logo então, clique, dentro da janela de **Configurações**, em **Propriedades**.

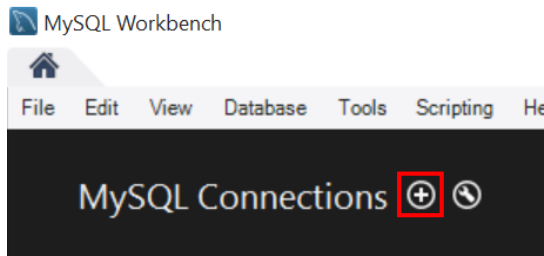


Na janela de Propriedades, note que podem ser visualizados muitos dados importantes com relação à conexão com seu banco de dados. Por agora, atente-se em guardar o **Nome do Host**, **Porta**, **Nome de Usuário**, **Senha** e **Cadeia de Conexão**, clicando no ícone de **copiar**, logo ao seu lado. Por fim, feche todas as janelas abertas.

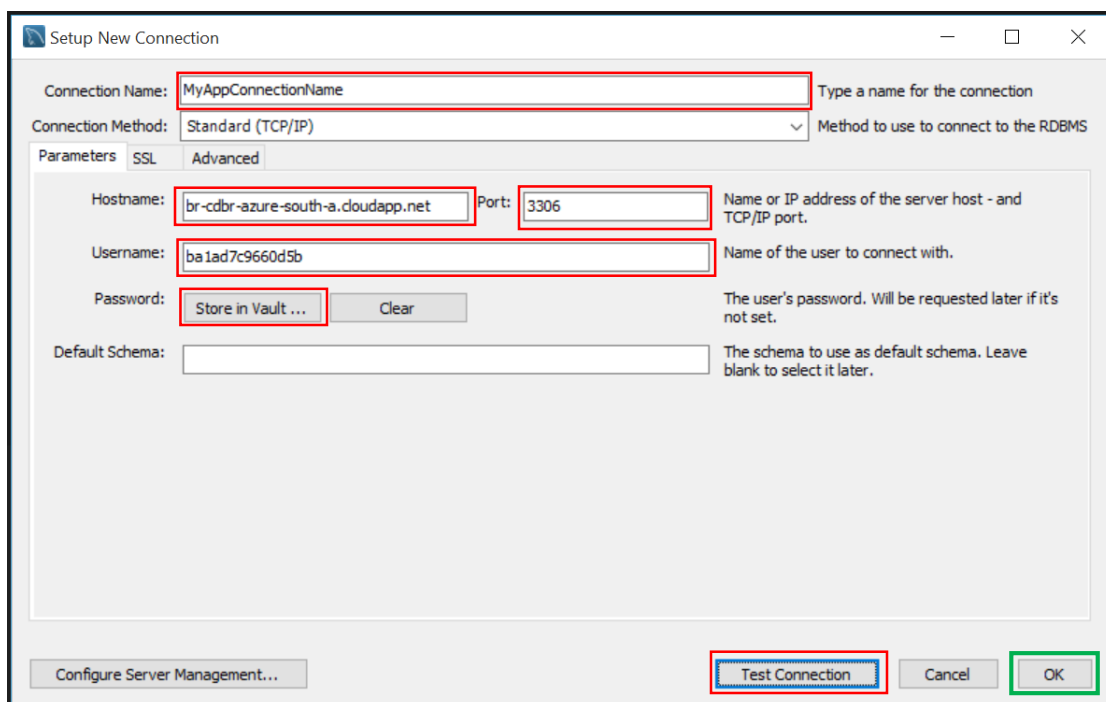


Passo II. Criando tabelas e populando o banco de dados

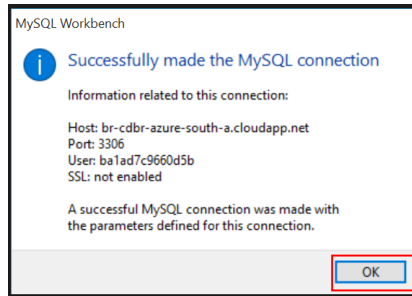
1. Abra o **MySQL Workbench** e clique no ícone de +, logo ao lado de MySQL Connection, para adicionar uma nova conexão MySQL:



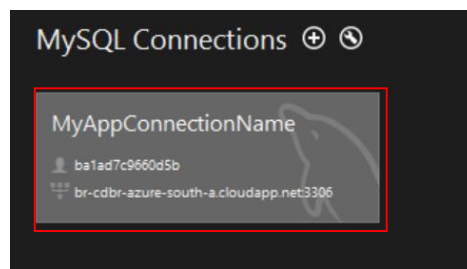
Dê um **nome** qualquer à conexão que está sendo criada, além de preencher todos os dados de conexão obtidos no passo anterior. Atente-se que para adicionar a respectiva senha do banco deve-se clicar no botão **Store in Vault**. Após o preenchimento dos dados, este formulário deve estar similar a figura a seguir:



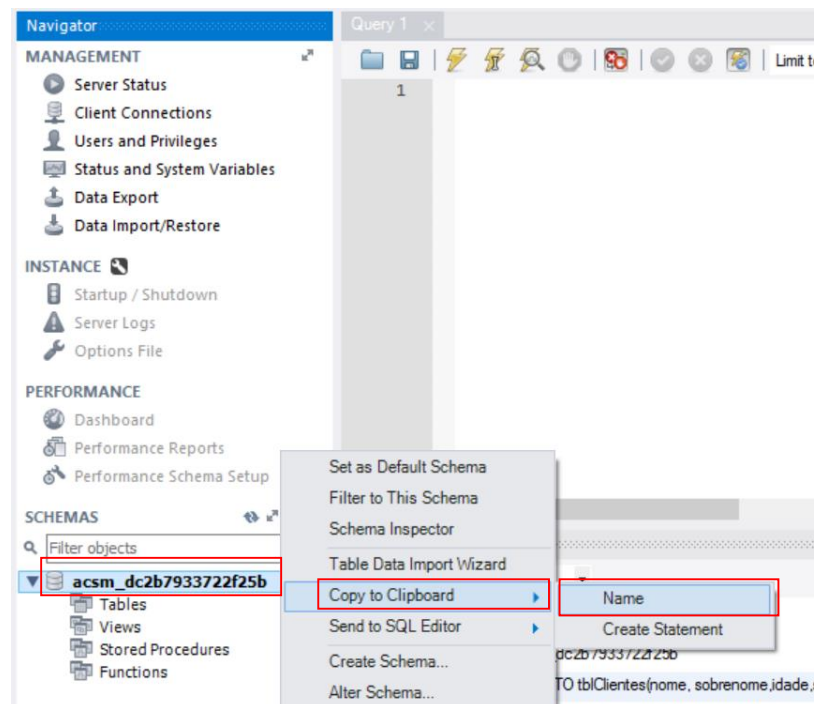
Logo então, clique em **Test Connection** para verificar se os dados foram inseridos de maneira correta. Se sim, a seguinte mensagem será disparada pelo programa:



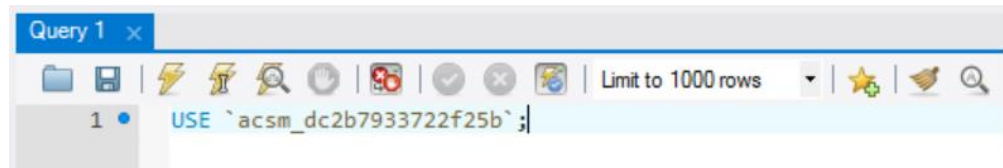
Feche a mensagem clicando em OK. Em seguida clique em OK na janela de nova conexão para salvar a respectiva conexão. A sua conexão será adicionada à dashboard do MySQL Workbench, clique nela para abrir o ambiente de programação.



2. Neste novo ambiente que foi aberto, localize à esquerda, dentro do painel **Navigator**, a categoria **Schemas**. Em seguida, **clique com o direito** em cima do banco de dados, navegue até **Copy to Clipboard** e em seguida **Name**; uma vez que precisamos do nome do banco de dados para realizarmos alterações no mesmo.



Dentro da janela de queries, digite o comando **USE**, em seguida de um espaço e **cole o valor** copiado anteriormente, adicionando por fim o caracter de **ponto e vírgula** (;). Logo, seu código deve estar semelhante a este:



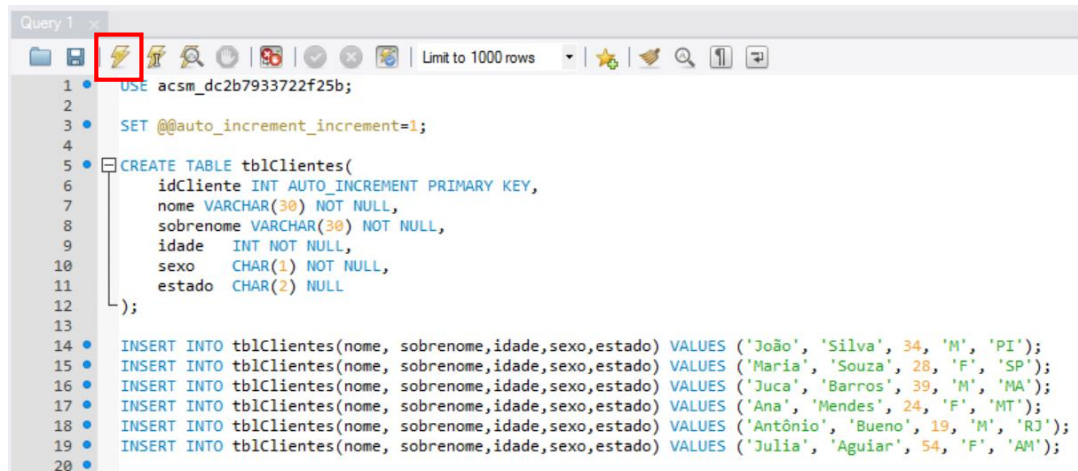
Pule uma linha e adicione o seguinte o código SQL, cuja função é criar uma tabela de clientes com alguns campos, bem como popular a mesma com alguns registros. Sinta-se livre para adicionar mais linhas:

```
SET @@auto_increment_increment=1;

CREATE TABLE tblClientes(
    idCliente INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(30) NOT NULL,
    sobrenome VARCHAR(30) NOT NULL,
    idade INT NOT NULL,
    sexo CHAR(1) NOT NULL,
    estado CHAR(2) NULL
);

INSERT INTO tblClientes(nome, sobrenome, idade, sexo, estado) VALUES ('João', 'Silva', 34, 'M', 'PI');
INSERT INTO tblClientes(nome, sobrenome, idade, sexo, estado) VALUES ('Maria', 'Souza', 28, 'F', 'SP');
INSERT INTO tblClientes(nome, sobrenome, idade, sexo, estado) VALUES ('Juca', 'Barros', 39, 'M', 'MA');
INSERT INTO tblClientes(nome, sobrenome, idade, sexo, estado) VALUES ('Ana', 'Mendes', 24, 'F', 'MT');
INSERT INTO tblClientes(nome, sobrenome, idade, sexo, estado) VALUES ('Antônio', 'Bueno', 19, 'M', 'RJ');
INSERT INTO tblClientes(nome, sobrenome, idade, sexo, estado) VALUES ('Julia', 'Aguiar', 54, 'F', 'AM');
```

Neste momento, seu código deve estar similar ao da seguinte figura:

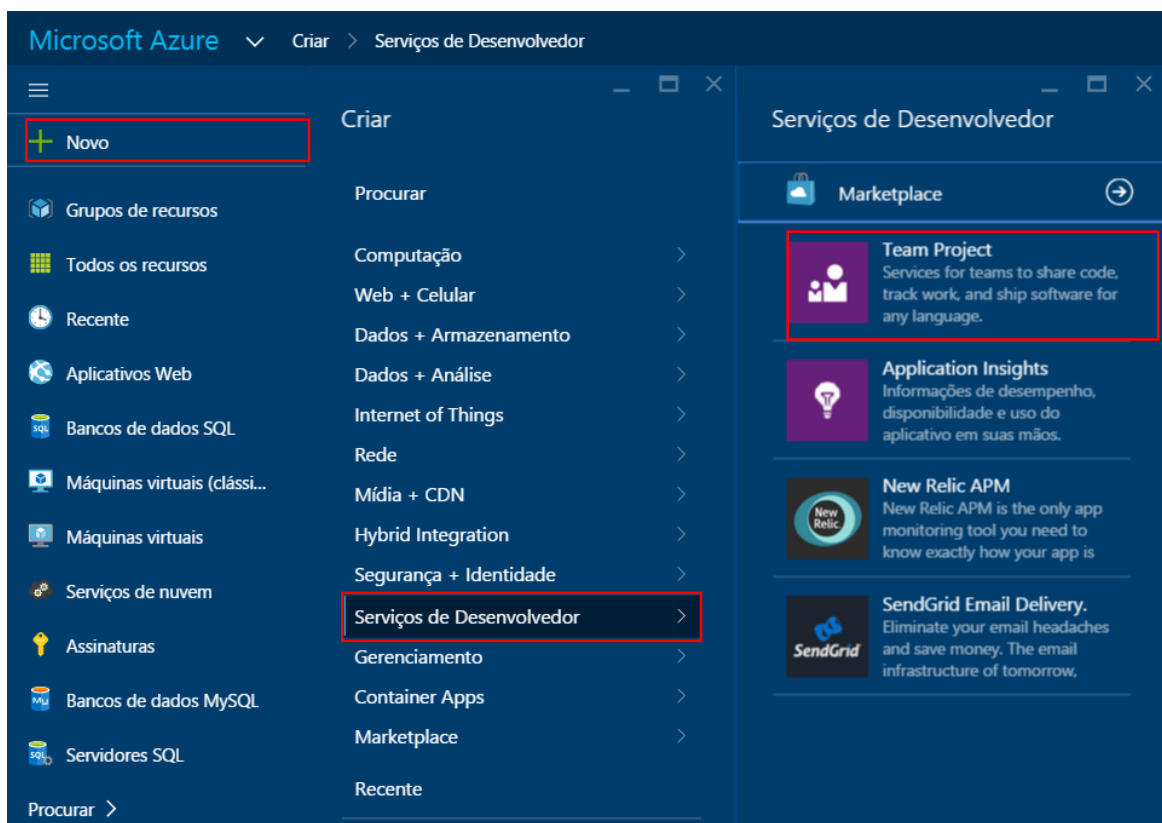


Conforme a figura anterior, clique no ícone do raio para executar o código. Caso não haja qualquer problema, a janela Output deve surgir, indicando o seguinte resultado de sucesso:

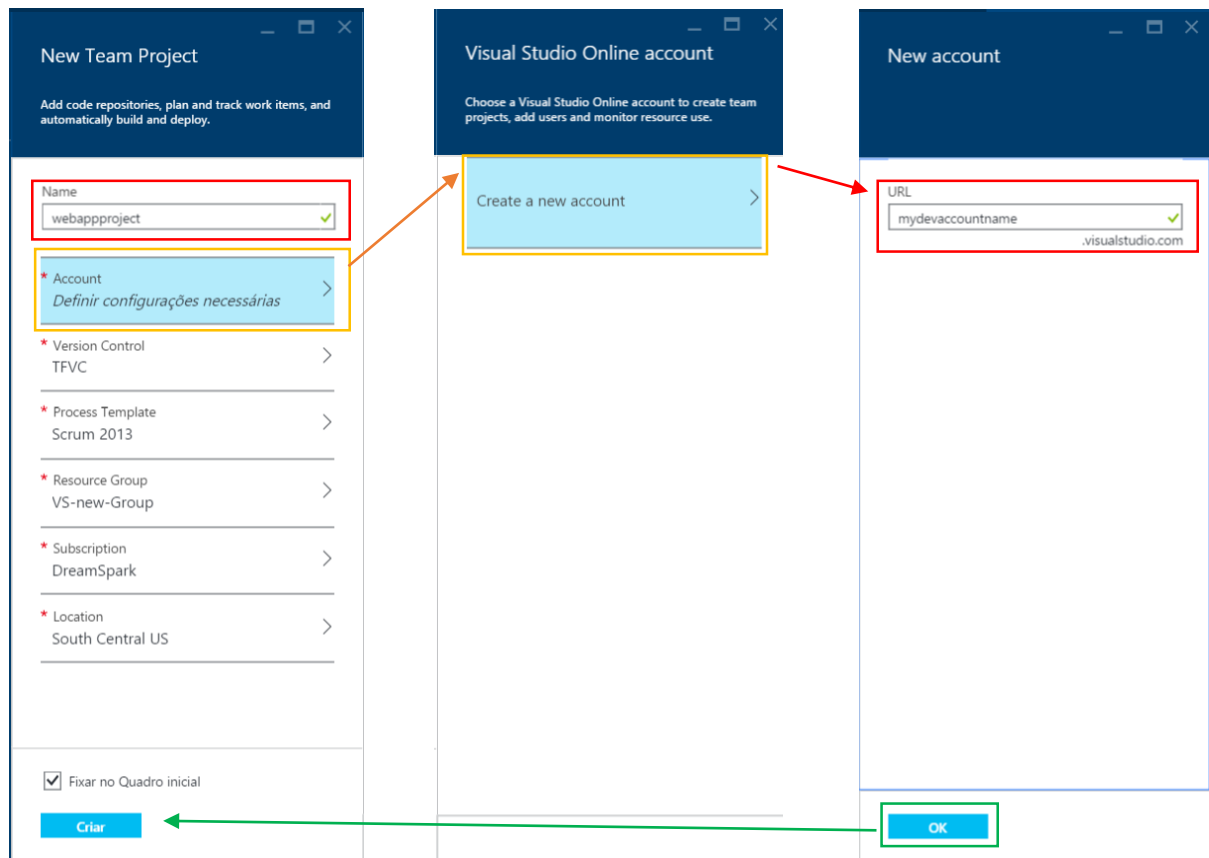
Output			
Action Output			
	Time	Action	Message
✓	1 17:30:57	USE acsm_dc2b7933722f25b	0 row(s) affected
✓	2 17:30:57	SET @@auto_increment_increment=1	0 row(s) affected
✓	3 17:30:57	CREATE TABLE tblClientes(idCliente INT AUTO_INCREMENT PRIMARY KEY, nom...	0 row(s) affected
✓	4 17:30:57	INSERT INTO tblClientes(nome, sobrenome, idade, sexo, estado) VALUES ('João', 'Silv...	1 row(s) affected
✓	5 17:30:57	INSERT INTO tblClientes(nome, sobrenome, idade, sexo, estado) VALUES ('Maria', 'So...	1 row(s) affected
✓	6 17:30:57	INSERT INTO tblClientes(nome, sobrenome, idade, sexo, estado) VALUES ('Juca', 'Barr...	1 row(s) affected
✓	7 17:30:58	INSERT INTO tblClientes(nome, sobrenome, idade, sexo, estado) VALUES ('Ana', 'Men...	1 row(s) affected
✓	8 17:30:58	INSERT INTO tblClientes(nome, sobrenome, idade, sexo, estado) VALUES ('Antônio', '...	1 row(s) affected
✓	9 17:30:58	INSERT INTO tblClientes(nome, sobrenome, idade, sexo, estado) VALUES ('Julia', 'Águi...	1 row(s) affected

Passo III. Criando um projeto no Visual Studio Online (VSO)

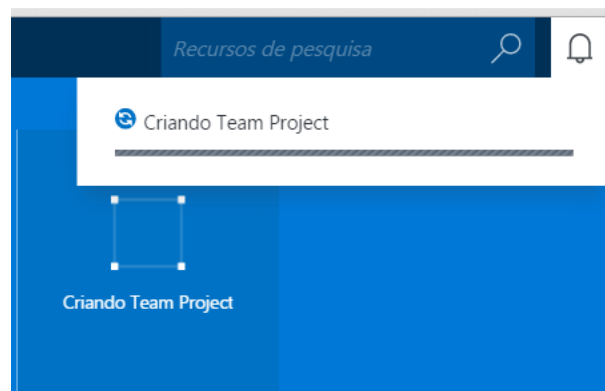
1. Voltando ao portal do Azure, clique em **Novo, Serviços do Desenvolvedor** e em seguida em **Team Project**. Na nova janela que irá abrir, dê um nome para o projeto de nossa web app. Logo então, clique em **Definir configurações necessárias**, seguido de **Create a new account**. Nesta nova janela, escolha um nome para sua conta no VSO e clique em OK. Não se preocupe pelo VSO estar alocado no Sul dos EUA, contas do VSO apenas estão disponíveis neste país.



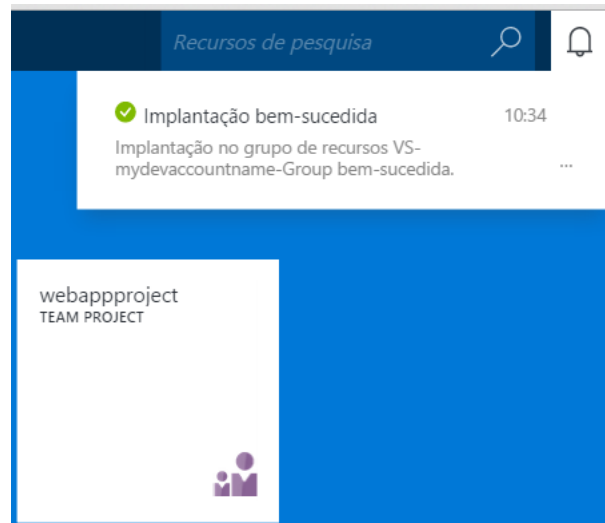
Passo a passo: Usando o Microsoft Azure para estudantes



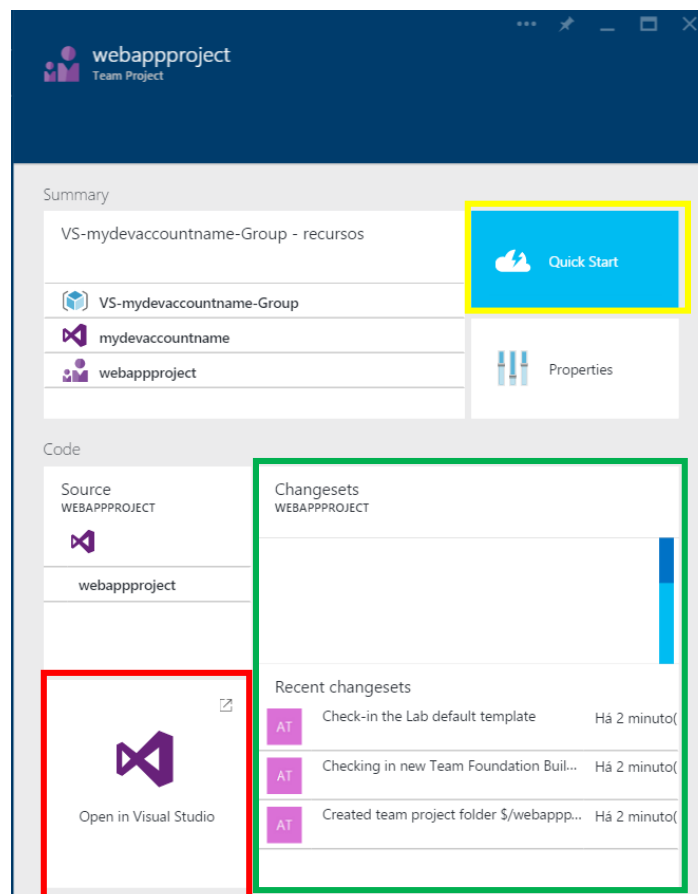
Em seguida, um ícone e uma notificação relatarão que sua conta e projeto estão sendo criados, por favor aguarde o processo terminar, que geralmente leva entre 1 e 3 minutos.



Após a conclusão, a seguinte mensagem e ícone em sua dashboard aparecerão:

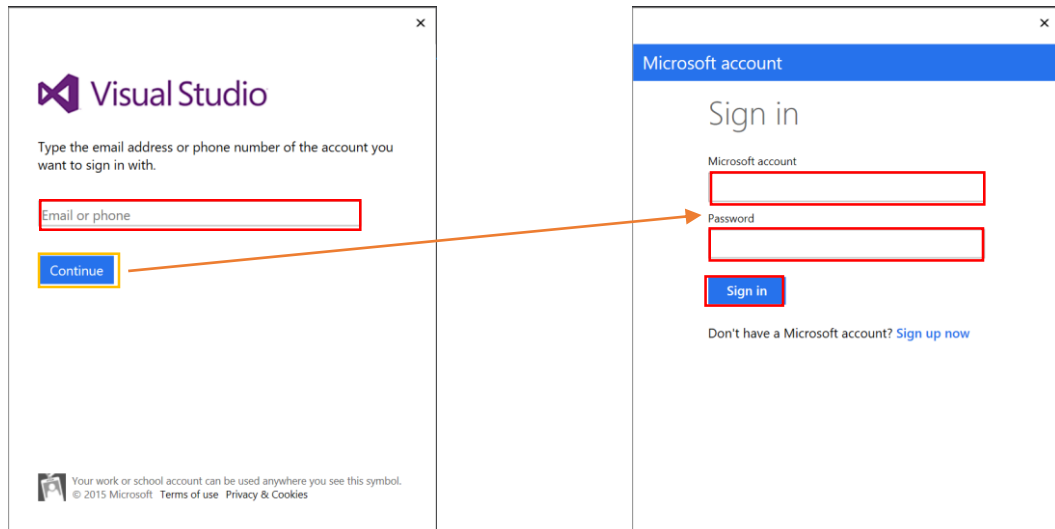


2. Clique no ícone do projeto do VSO criado anteriormente, de modo que uma janela de gerenciamento do projeto será aberta. Nesta janela, pode-se adicionar usuários que irão colaborar em seu projeto (em amarelo, na figura a seguir), visualizar recentes mudanças e submissões feitas no código bem como o usuário que as fez (em verde, na figura a seguir), entre outros recursos. Estamos interessados, por hora, em abrir este projeto, portanto clique em **Open in Visual Studio**.

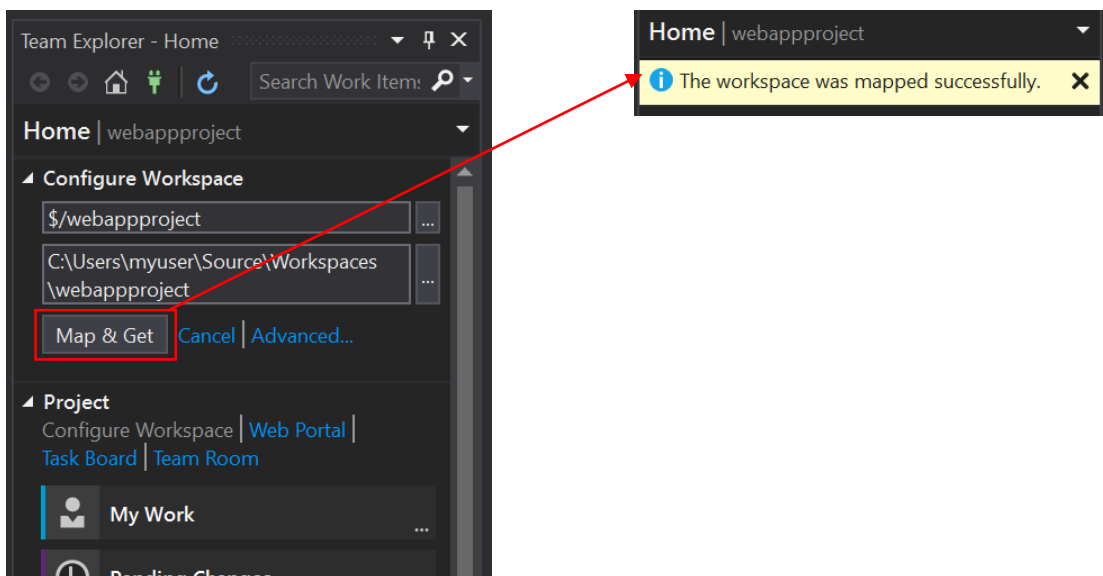


Passo IV. Desenvolvendo e publicando a web app

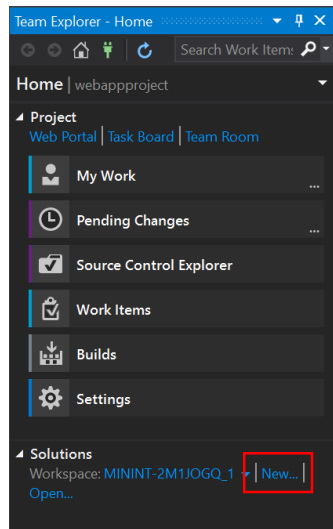
1. Após o Visual Studio abrir, forneça o endereço de e-mail da conta que estamos trabalhando, em seguida escolha a categoria da mesma (conta Microsoft ou de estudante/trabalho). Por fim, entre com sua senha e clique em **Sign in**.



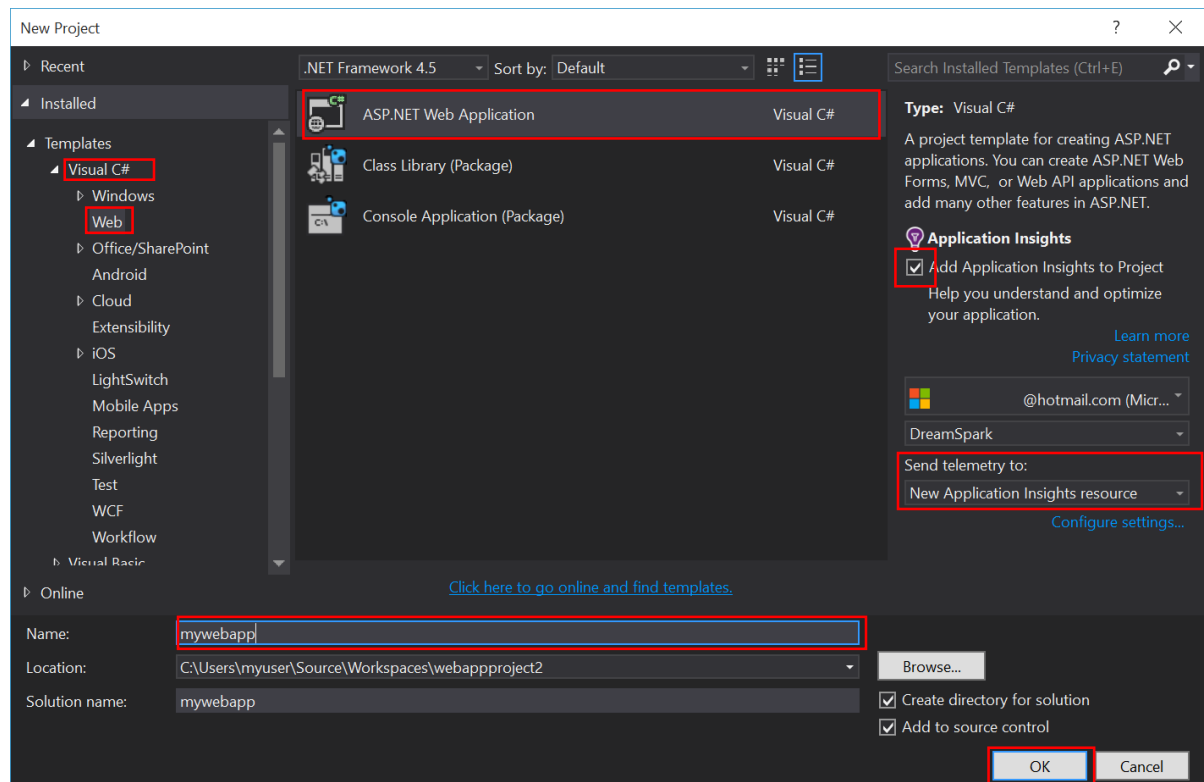
Após um login bem-sucedido note que, na direita de sua tela, há um painel chamado **Team Explorer**. É através deste menu que se realiza todo o gerenciamento do trabalho colaborativo de seu projeto. Pode-se realizar ambos check-in (submeter seu código para a nuvem) quanto check-out (receber a versão mais recente do projeto), bem como revisar ou aprovar mudanças propostas em geral. Dentro da sessão Configure **Workspace**, clique no botão **Map & Get**, para associar um diretório local com o espaço de trabalho online.



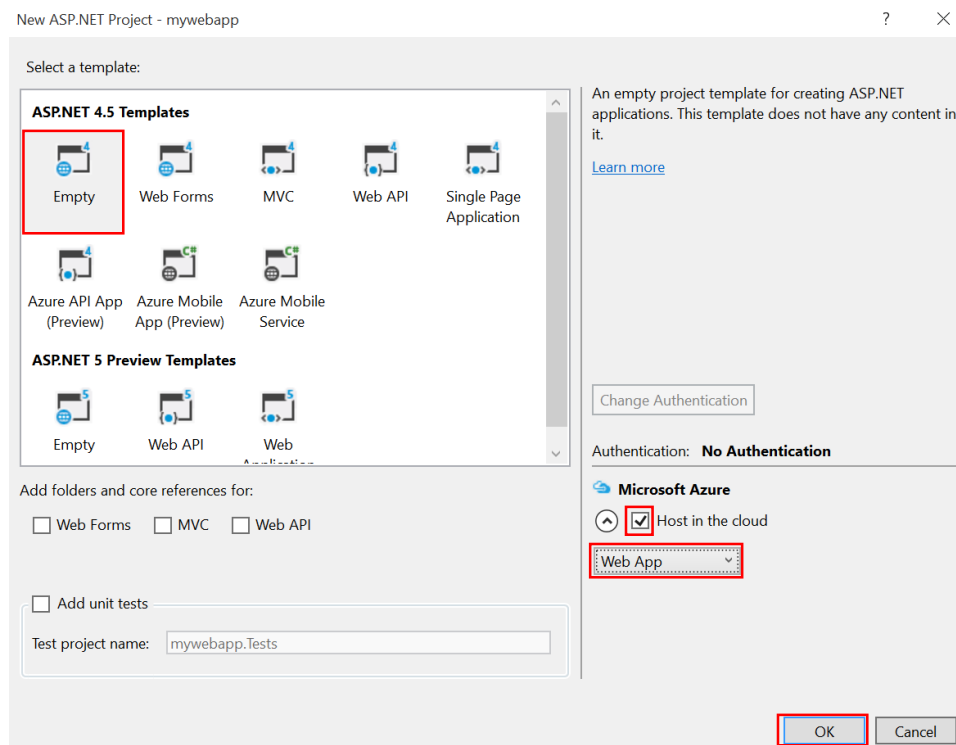
2. Após o ambiente de desenvolvimento for configurado e a mensagem de sucesso ser exibida, ainda no painel de Team Explorer clique, dentro da sessão Solutions, em **New**.



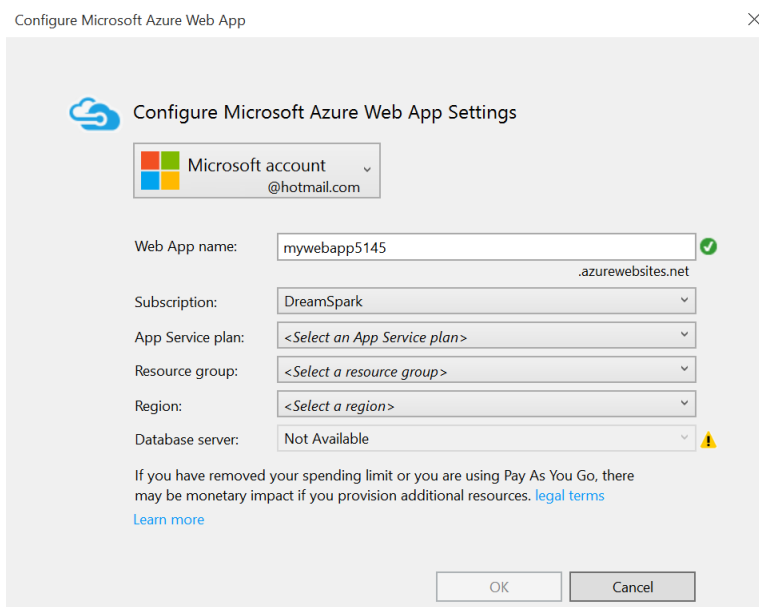
Preste bastante atenção aos detalhes nesta janela que se abriu. Dentro de **Visual C#**, navegue até a categoria **Web**, escolhendo o template de projeto **ASP.NET Web Application**. Dê um **nome** de sua preferência ao projeto. Repare que o local do projeto é o mesmo que foi setado na etapa anterior. Atente-se também, no canto direito, em **chegar a caixa de Applications Insights**. Confira também se a **opção Send telemetry to** indica a opção **New Application Insights resource**.



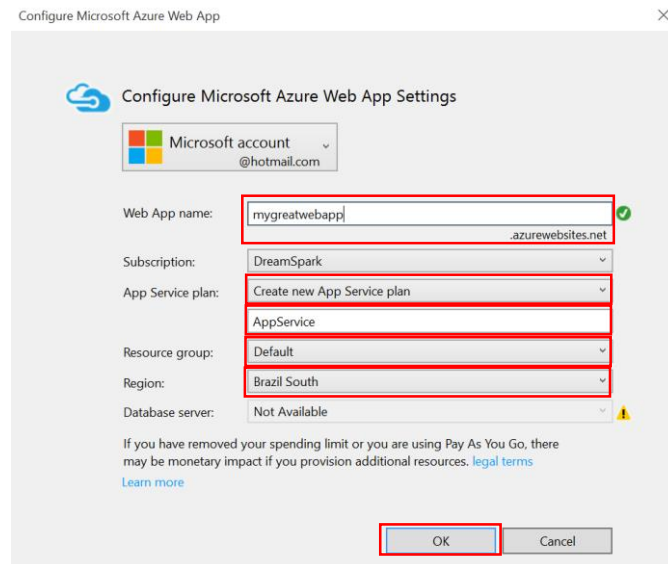
Em seguida a janela de um novo projeto ASP.NET será aberto, opte pelo template **Empty** dentro de **ASP.NET 4.5**. Atente-se em **marcar a caixa Host in the cloud**, bem como a opção **Web App**, ambos dentro da região **Microsoft Azure**. Por fim, clique em OK.



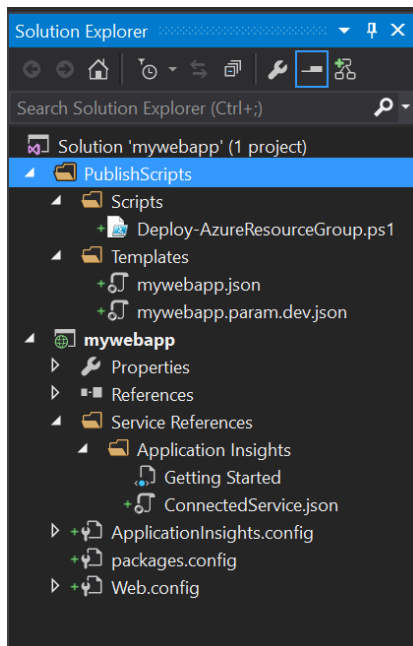
O próximo passo envolve o input de informações a respeito da publicação da web app no Azure. Primeiro certifique-se que a conta mostrada é a que possui a assinatura do Dreamspark. Note que por padrão, ele buscou um nome válido e único para nosso o endereço da web app.



Dê um **nome** de sua preferência a sua web app, em que note também o domínio que ela terá, azurewebsites.net. Caso você possua um domínio próprio, no portal de gerenciamento do Azure, [poderá setá-lo como padrão](#). Selecione a opção **Create new App Service plan**, bem como dê um **nome** para este plano de serviços. Certifique-se de escolher a opção **Default** para **Resource Group** (Criamos tal grupo quando criamos o banco de dados MySQL). Atente-se também em escolher a **região de Brazil South**. Não se preocupe com o ícone de alerta ao lado de Database server, isto apenas significa que você não consegue criar bancos de dados SQL Server e linka-los diretamente a seu site. Após tais configurações, clique em **OK**.



Logo então, uma pequena janela indicando a criação de sua solução, que automaticamente incluirá referências à web app no Azure quanto o respectivo Applications Insights associado.



Após a criação da solução, um projeto web será adicionado a ele, além de uma pasta contendo alguns scripts de publicação.

A fim de podermos utilizar o banco de dados MySQL com nosso projeto ASP.NET é necessário que se instale os respectivos pacotes contendo as bibliotecas de acesso ao banco. Para tal, iremos utilizar o gerenciador de pacotes NuGet e automatizarmos a instalação. Vá até o **menu Tools**, depois **NuGet Package Manager**, clicando depois em **Packager Manager Console**.

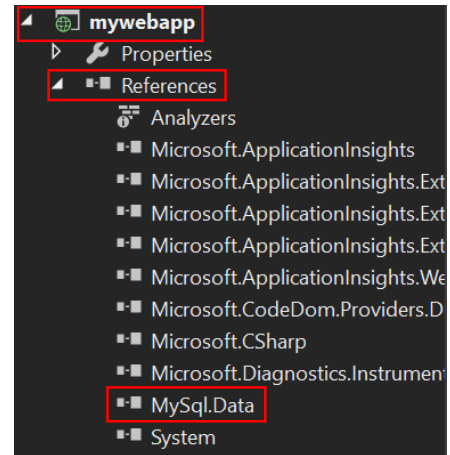
Quando o console do NuGet aparecer, digite o comando:

```
PM> Install-Package MySql.Data
```

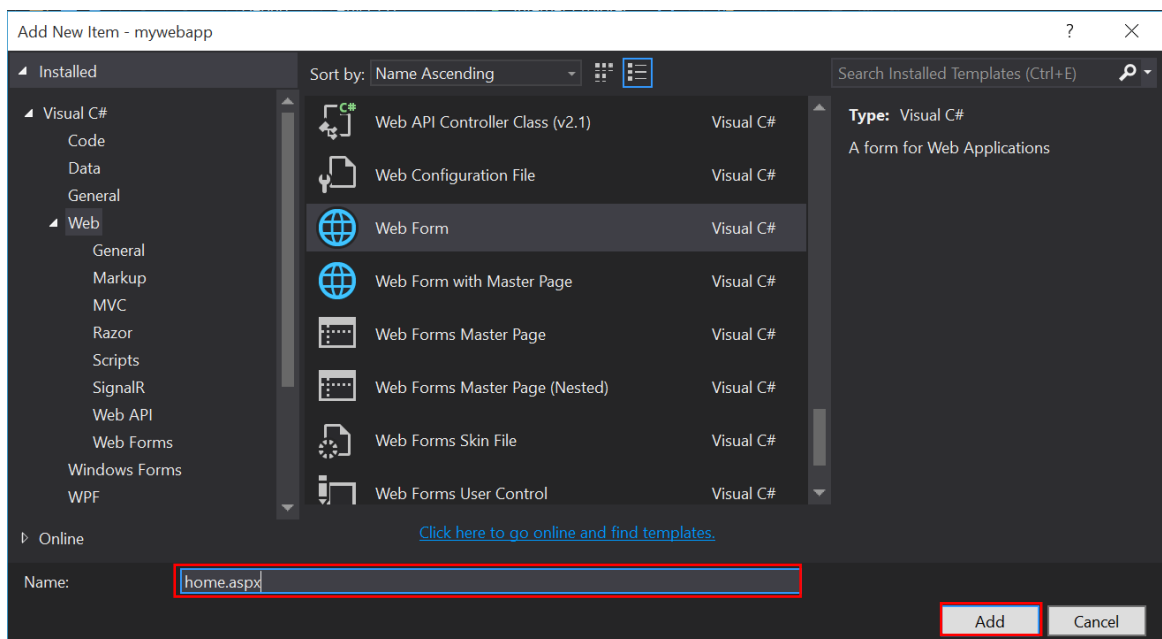
A instalação logo começará e algumas mensagens de controle serem escritas na janela de output.

Após a instalação bem-sucedida dos pacotes envolvidos, o readme.txt das referências utilizadas aparecerá em sua tela.

Caso queira ter certeza que a biblioteca foi adicionada em seu projeto, dentro do painel **Solution Explorer**, vá até o **projeto web** associado. Em seguida, expanda o nó **References**, você deve encontrar um namespace chamado **MySql.Data**.



3. Vamos iniciar agora o desenvolvimento do app propriamente dito, adicionando uma página ASP.NET ao mesmo. **Clique com o direito** em cima de seu projeto web, navegue até **Add**, depois **New Item**. Expanda o nó **Visual C#** e depois clique em **Web**. Escolha o template **Web Form**. Dê o nome **home.aspx** e clique em **Add**.

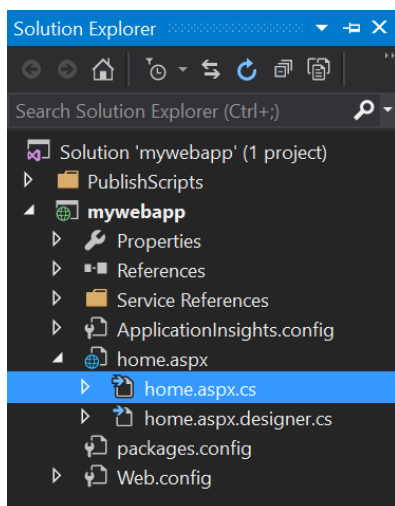


Logo após a página ter sido adicionada ao projeto, o arquivo .aspx que contém algumas tags HTML iniciais aparecerá. Em nosso projeto, queremos mostrar ao usuário final um formulário de cadastro de clientes com os campos Nome, Sobrenome, Idade, Sexo e Estado; além de uma tabela contendo todos os registros já presentes no banco de dados. Para isto, apague todo o conteúdo do arquivo e adicione as marcações a seguir:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="home.aspx.cs"
Inherits="mywebapp.home" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>My Page!</title>
</head>
<body>
    <h1>Web App</h1>
    <h2>Cadastrar cliente:</h2>
    <form runat="server">
        <label for="txtNome">Nome:</label>
        <input runat="server" type="text" name="Nome" id="txtNome" />
        <br />
        <label for="txtSobrenome">Sobrenome:</label>
        <input runat="server" type="text" name="Sobrenome" id="txtSobrenome" />
        <br />
        <label for="txtIdade">Idade:</label>
        <input runat="server" type="number" name="Idade" id="txtIdade" />
        <br />
        <label for="cmbSexo">Sexo:</label>
        <select runat="server" name="Sexo" id="sltSexo">
            <option value=""></option>
            <option value="F">F</option>
            <option value="M">M</option>
        </select>
        <br />
        <label for="txtEstado">Estado:</label>
        <input runat="server" type="text" name="Estado" id="txtEstado" />
        <br />
        <asp:Button runat="server" Text="Cadastrar" OnClick="OnClick"/>
    </form>
    <p runat="server" ID="msg"></p>
    <h2>Dados:</h2>
    <asp:table ID="tblDados" runat="server"/>
</body>
</html>
```



Precisamos agora adicionar a lógica de code-behind de nossa web app, acrescentando a programação necessária para que nossa página se conecte ao banco de dados e obtenha os registros, bem como seja capaz de inserir novos dados no banco.

Para tal, no painel **Solution Explorer**, expanda o nó do arquivo de nome **home.aspx**, de modo que um arquivo com nome **home.aspx.cs** aparecerá. Clique duplo neste item para abrir o código em C# associado a essa página.

Atente-se as primeiras linhas deste arquivo, elas contêm referências as bibliotecas que podem ser utilizadas pelo

desenvolvedor. Porém, iremos precisar apenas de apenas 3, que são as a seguir. Apague as outras e troque-as por estas:

```
using System;  
using System.Web.UI.WebControls;  
using MySql.Data.MySqlClient;
```

Necessitamos informar a nossa web app como esta irá se conectar ao banco de dados. Para isso será preciso da [cadeia de conexão](#) de nosso banco de dados, obtida no Passo I deste guia. Portanto, **logo acima da linha que contém o método Page_Load**, insira a seguinte linha, trocando o texto **SUA STRING DE CONEXÃO AQUI** por sua cadeia de conexão:

```
MySqlConnection connection = new MySqlConnection("SUA STRING DE CONEXÃO AQUI");
```

Dentro do método Page_Load, iremos realizar uma checagem a fim de evitar que a página realize processamento desnecessário quando a página atualizar, em nosso caso particular. Portanto, o método Page_Load deve estar exatamente assim:

```
protected void Page_Load(object sender, EventArgs e)  
{  
    if (!Page.IsPostBack)  
        atualizar();  
}
```

O trecho a seguir, que responderá ao evento de **clique** do botão de nossa página web, é responsável por interagir com o banco de dados inserindo um novo registro ao mesmo. Inicialmente ele define e instancia um objeto capaz de se comunicar com o banco de dados MySQL, em que logo em seguida é definido os comandos a serem executados durante essa interação. Conforme haja sucesso nesta operação, a página será atualizada, o formulário de cadastro será limpo e uma mensagem de sucesso aparecerá na tela. Do contrário, um erro amigável será mostrado ao usuário. Adicione-o **logo abaixo** do método Page_Load:

```
protected void OnClick(object sender, EventArgs e)
{
    try
    {
        string cmd = "";
        MySqlCommand command = new MySqlCommand();
        command.Connection = connection;

        cmd += "SET @@auto_increment_increment=1;";
        cmd += "INSERT INTO tblClientes(nome, sobrenome, idade, sexo, estado) VALUES (@nome, @sobrenome, @idade, @sexo, @estado);";

        command.CommandText = cmd;

        command.Parameters.AddWithValue("@nome", txtNome.Value);
        command.Parameters.AddWithValue("@sobrenome", txtSobrenome.Value);
        command.Parameters.AddWithValue("@idade", txtIdade.Value);
        command.Parameters.AddWithValue("@sexo", sltSexo.Value);
        command.Parameters.AddWithValue("@estado", txtEstado.Value);

        connection.Open();
        command.ExecuteNonQuery();
        connection.Close();

        atualizar();
        limpar();
        msg.InnerText = "Sucesso!";
    }
    catch (Exception ex)
    {
        msg.InnerText = "Erro:" + ex.Message;
    }
}
```

Precisamos criar o método **atualizar**, responsável por consultar nosso banco de dados, ler os registros e inserir em uma tabela presente na página web que estamos trabalhando. Adicione, portanto, o seguinte trecho de código:

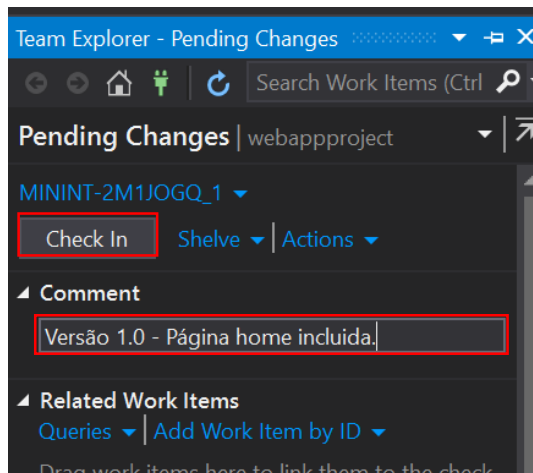
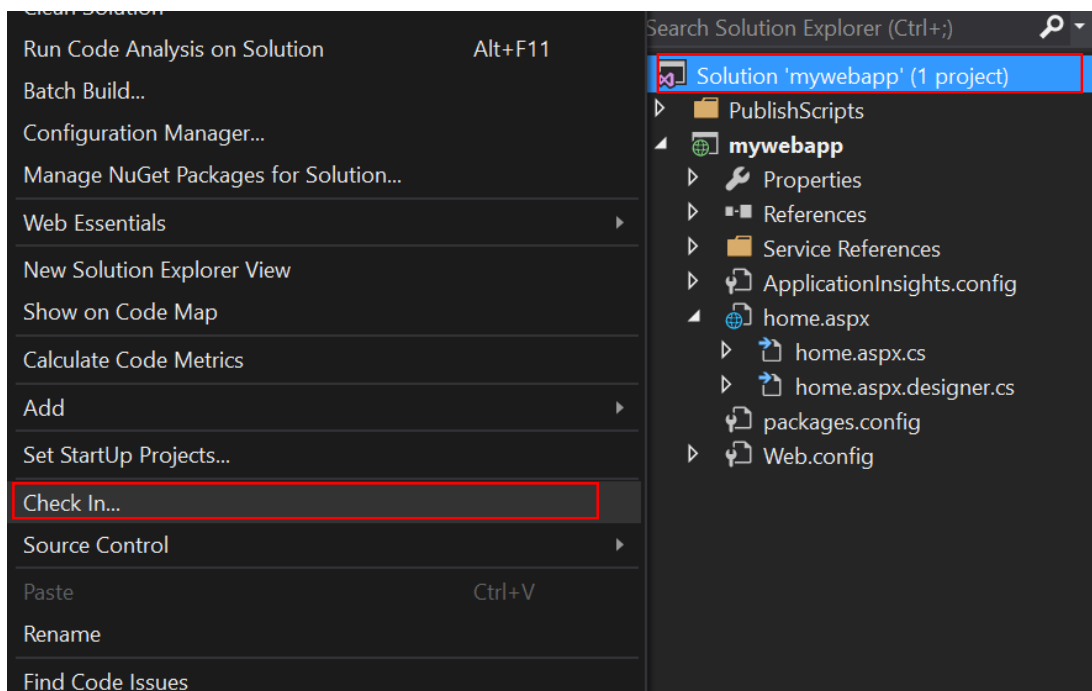
```
private void atualizar()
{
    try
    {
        tblDados.Rows.Clear();
        MySqlCommand command = new MySqlCommand("SELECT * FROM tblClientes;",
connection);
        connection.Open();

        MySqlDataReader reader = command.ExecuteReader();
        while (reader.Read())
        {
            var row = new TableRow();
            TableCell[] cells = new TableCell[6];
            for (int i = 0; i < 6; i++)
            {
                cells[i] = new TableCell();
                cells[i].Text = reader[i].ToString();
            }
            row.Cells.AddRange(cells);
            tblDados.Rows.Add(row);
        }
        reader.Close();
        connection.Close();
    }
    catch (Exception ex)
    {
        msg.InnerText = "Erro:" + ex.Message;
    }
}
```

Por fim, adicione ao código-fonte o método **limpar**:

```
private void limpar()
{
    txtNome.Value = "";
    txtSobrenome.Value = "";
    txtIdade.Value = "";
    sltSexo.Value = "";
    txtEstado.Value = "";
}
```

Uma vez que isto todos os trechos acima tenham sido corretamente adicionados ao arquivo em C#, está quase na hora de vermos o resultado final! Antes disto, clique com o **botão direito** sobre a página **home.aspx** e escolha a opção **Set As Start Page**. Agora, submeta seu código ao Visual Studio Online, através do processo de Check-in. Através disso, você poderá continuar a programar sua web app de qualquer computador, além de convidar outros colaboradores. Para tal, clique com o botão direito sobre a solução, dentro do painel **Solution Explorer**, clicando em seguida em **Check-In**. Caso seja solicitado, clique em salvar.



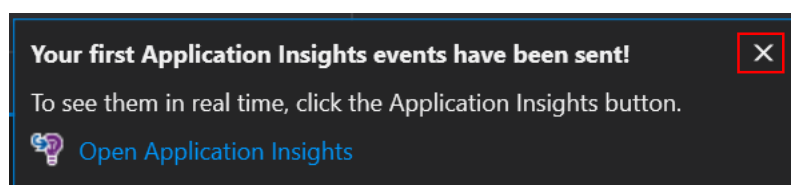
O seguinte cenário ao lado irá surgir, dentro do painel **Team Explorer**. Dentro da caixa **Comment**, escreva algo que descreva o que há de novo ou algo relevante nesta versão que está sendo submetida para o VSO. Em seguida, clique em **Check In**.

Confirme quando for questionado se deseja fazer o Check In de um número x de itens para sua conta do VSO. Após um carregamento que tipicamente é bem rápido, mas que pode demorar um pouco mais dependendo da qualidade do seu link de internet, temos que uma mensagem de sucesso em amarelo irá

surgir.

Hora de testar! Aperte F5 para visualizar sua web app pronta! Teste para ver se tudo está funcionando corretamente.

Uma mensagem do Applications Insights irá aparecer, indicando que a coleta de dados, por mais que sua web app esteja hospedada localmente, irá iniciar. Feche a mensagem, iremos tratar melhor deste assunto no próximo passo.



Web App

Cadastrar cliente:

Nome:

Sobrenome:

Idade:

Sexo:

Estado:

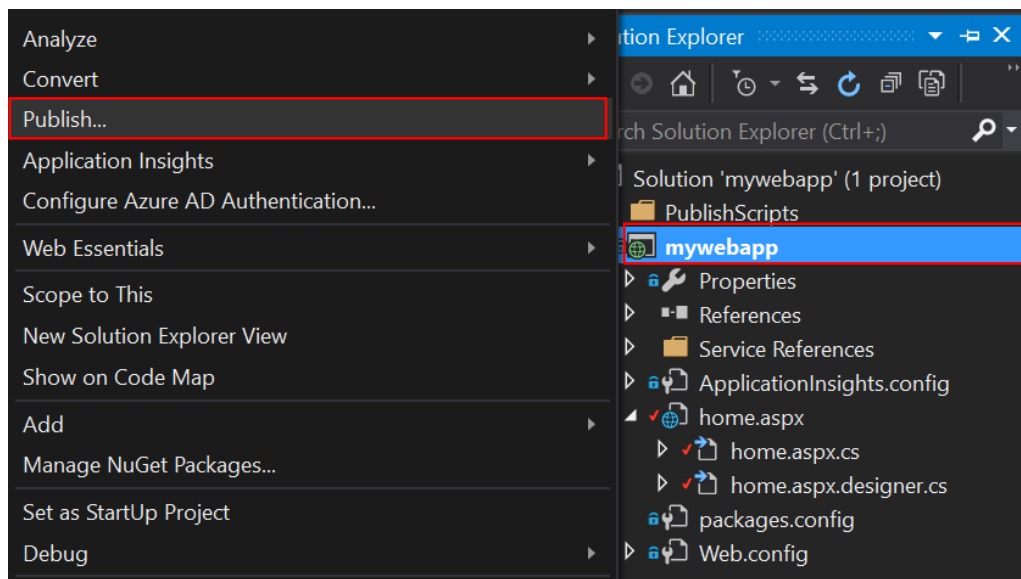
Dados:

1	João	Silva	34	M	PI
2	Maria	Souza	28	F	SP
3	Juca	Barros	39	M	MA
4	Ana	Mendes	24	F	MT
5	Antônio	Bueno	19	M	RJ
6	Julia	Aguiar	54	F	AM

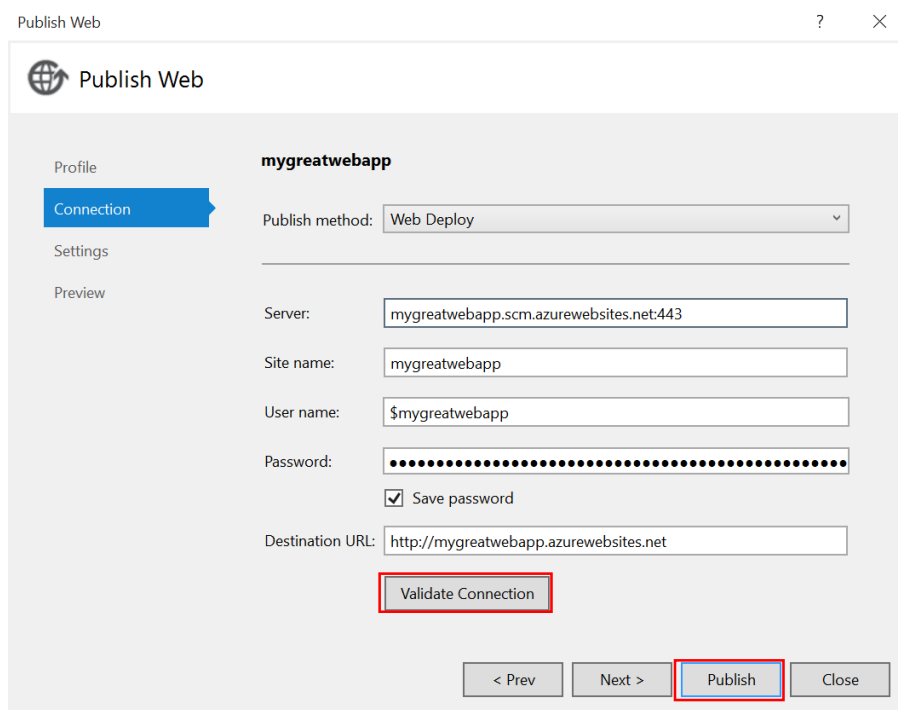
4. Por fim, mandaremos nossa web app para o Azure, para que ela fique disponível para qualquer pessoa ao redor do mundo!

Há várias maneiras de se realizar este processo de deploy, como a publicação feita diretamente do Visual Studio, upload de arquivos via FTP, usando um arquivo de configurações fornecido pelo Azure, entre muitos outros cenários. Iremos optar, como já dito, pela integração nativa fornecida pelo Visual Studio.

Caso seu app ainda esteja em execução, **pare** através da barra de ferramentas (ícone de um quadrado vermelho). **Clique com o direito** sobre o projeto web, depois vá em **Publish**.



Em seguida, uma janela contendo os dados de conexão aparecerá. Uma vez que estamos usando a integração nativa do Visual Studio, todos os campos serão automaticamente preenchidos. Para verificar se a conexão está sendo feita sem quaisquer problemas, clique em **Validate Connection**. Após um ícone de sucesso aparecer logo ao lado do botão de validação, clique sobre o botão **Publish**.



Publish Web

Profile

mygreatwebapp

Connection

Settings

Preview

Publish method: Web Deploy

Server: mygreatwebapp.scm.azurewebsites.net:443

Site name: mygreatwebapp

User name: \$mygreatwebapp

Password:

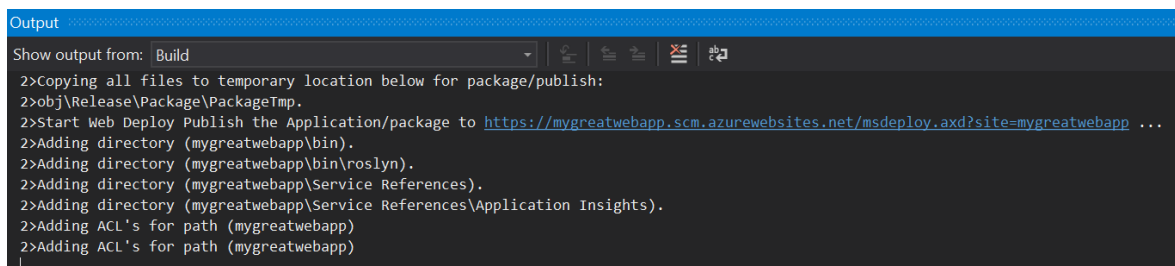
☒ Save password

Destination URL: http://mygreatwebapp.azurewebsites.net

Validate Connection

< Prev Next > Publish Close

Logo então, o processo de submissão da web app ser iniciará. Você notará que a janela de output irá informar em tempo real quais procedimentos da submissão estão sendo realizado, como configurações, upload de arquivos, entre outros.

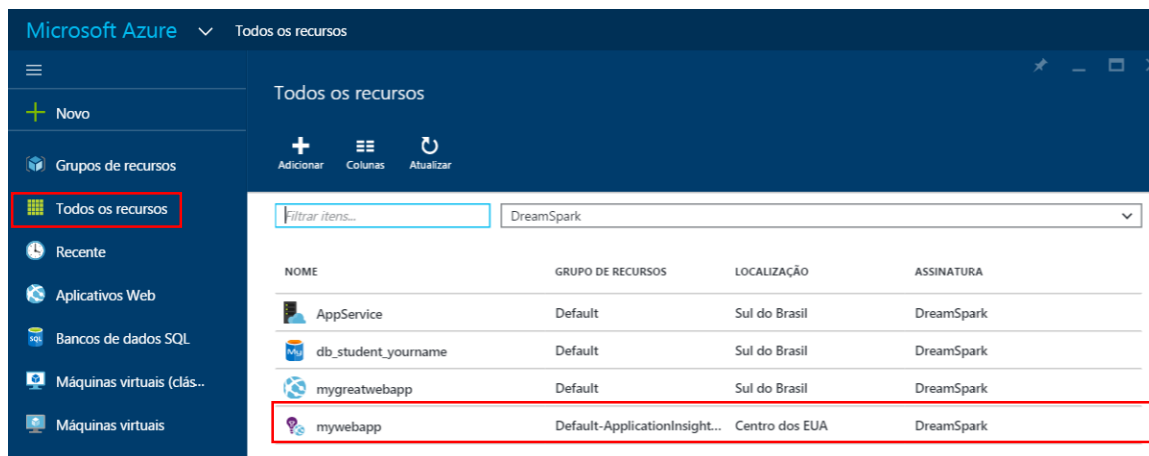


```
Output
Show output from: Build
2>Copying all files to temporary location below for package/publish:
2>obj\Release\Package\PackageImp.
2>Start Web Deploy Publish the Application/package to https://mygreatwebapp.scm.azurewebsites.net/msdeploy.axd?site=mygreatwebapp ...
2>Adding directory (mygreatwebapp\bin).
2>Adding directory (mygreatwebapp\bin\roslyn).
2>Adding directory (mygreatwebapp\Service References).
2>Adding directory (mygreatwebapp\Service References\Application Insights).
2>Adding ACL's for path (mygreatwebapp)
2>Adding ACL's for path (mygreatwebapp)
```

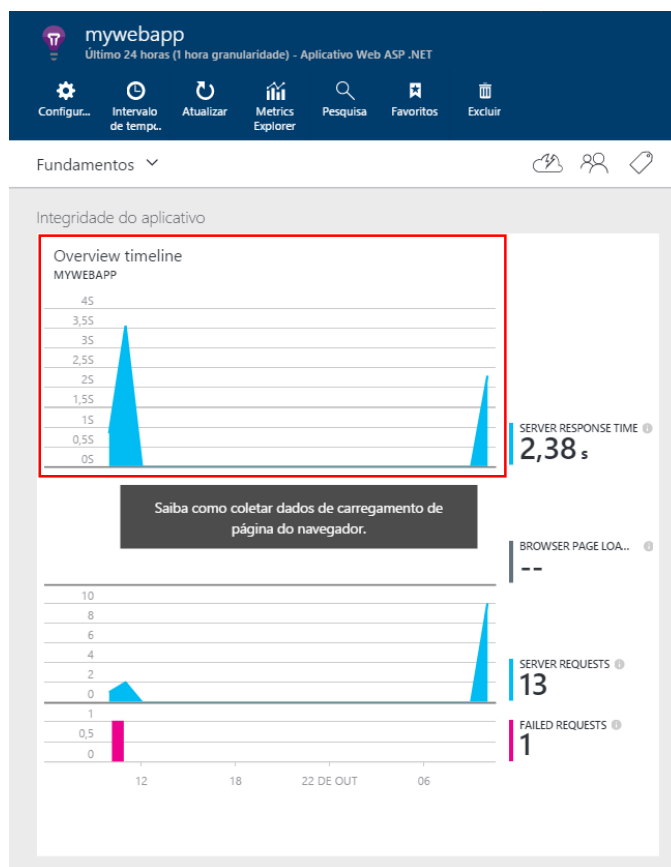
Uma vez que o processo tenha sido finalizado, sua web app abrirá em seu seu browser padrão!

Passo V. Utilizando o Application Insights

1. Acesse o **portal** do Azure e, caso não esteja logado, faça-o. Vá em seguida até **Todos os recursos** e depois clique sobre o **Application Insights** relacionado à sua web app.



2. Logo então, a página principal de seu Application Insights irá surgir, contendo dados sobre tempo de resposta, requisições ao servidor e possíveis falhas!



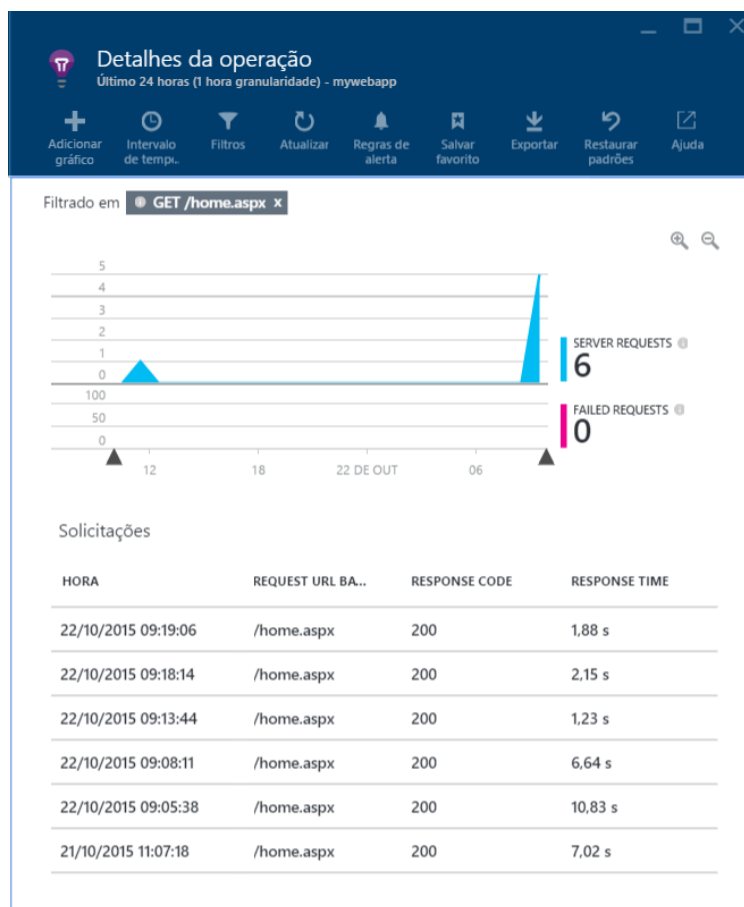
3. Clique sobre o primeiro gráfico, por exemplo, para abrir mais detalhes sobre o mesmo. Na nova janela de detalhes que irá abrir, pode-se, dentre outras possibilidades, verificar quais

páginas são mais solicitadas em sua web app, bem como o tempo médio de resposta das páginas/operações HTTP.

Média de Server response time por Operation name





OPERATION NAME	MÉDIA	CONTAGEM	DISPOSITI...
GET /home.aspx	4,96 s	6	3,48 s
GET /WebForm1.aspx	843,67 ms	1	0 ms
GET /_browserLink/req...	108,02 ms	1	0 ms
GET /_browserLink/req...	103,1 ms	1	0 ms
GET /_browserLink/req...	69,17 ms	1	0 ms
GET /_browserLink/req...	67,53 ms	1	0 ms
GET /_browserLink/req...	24,29 ms	1	0 ms
GET /_browserLink/req...	6,58 ms	1	0 ms

- Ao clicar em um dos itens/páginas acima, você poderá obter mais detalhes sobre cada requisição feita ao servidor, filtrando os resultados desejados.



Parabéns! Você terminou este guia!

E assim termina nosso guia de como utilizar os serviços da nuvem ofertados pela Microsoft sem qualquer custo a você, estudante! Durante este guia, aprendemos como usar e integrar os 4 serviços disponíveis: banco de dados MySQL, Web App, Visual Studio Online e Application Insights. Esperamos que tenha gostado!

db_student_yourname MYSQL DATABASE Healthy 	webappproject TEAM PROJECT 
mygreatwebapp WEB APP Running 	mywebapp APPLICATION INSIGHTS 

Continue seu aprendizado:



Referências

- *Portal do Microsoft Azure:*
<https://portal.azure.com/>
- *Programa Dreamspark:*
<https://www.dreamspark.com/Student/Default.aspx>
- *Download VS 2015 Community:*
<https://www.visualstudio.com/products/visual-studio-community-vs>
- *MySQL Workbench:*
<http://dev.mysql.com/downloads/workbench/>
- *NuGet e MySql.Data:*
<https://www.nuget.org/packages/MySql.Data/>