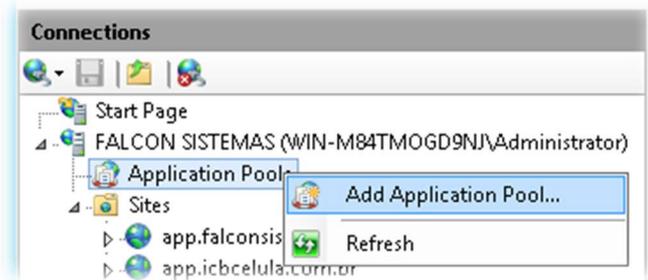


Publicando sua aplicação no IIS – ISAPI Module

Neste passo a passo irei demonstrar como eu hospedo minhas aplicações, porém fica a seu critério a forma que deseja hospedar suas aplicações no IIS.

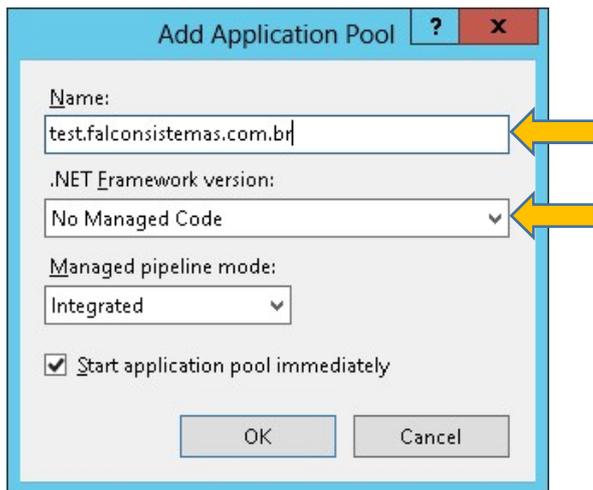
Para este exemplo foi utilizado o IIS8.

Abra o gerenciador do IIS e adicionei primeiramente o Pool de aplicativo, ele é o responsável por gerenciar sua aplicação, sendo uma instância do w3wp.exe:

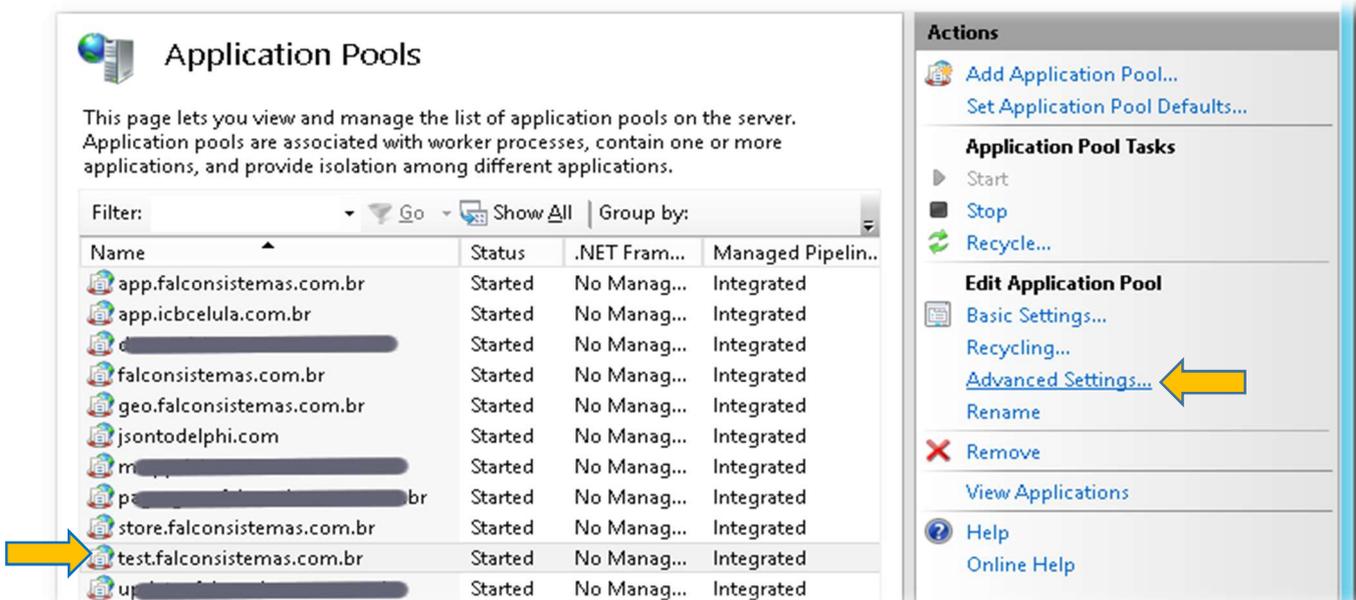


Dica* utilize apenas um único application pool por aplicação.

Por padrão o nome do meu application pool, é o endereço do meu site, e em .NET Framework marque a opção No Managed Code, pois os aplicativos em uniGui não possuem código gerenciado pelo .NET.



Após criado seu application pool você deve ir nas configurações avançadas



Advanced Settings



General

.NET Framework Version	No Managed Code	
Enable 32-Bit Applications	True	
Managed Pipeline Mode	Integrated	
Name	test.falconsistemas.com.br	
Queue Length	1000	
Start Automatically	True	
Start Mode	OnDemand	

CPU

Limit (1/1000 of %)	0
Limit Action	NoAction
Limit Interval (minutes)	5
Processor Affinity Enabled	False
Processor Affinity Mask	4294967295
Processor Affinity Mask (64-bit option)	4294967295

Process Model

Generate Process Model Event Log Entry	
Identity	administrator
Idle Time-out (minutes)	0
Load User Profile	False
Maximum Worker Processes	1
Ping Enabled	True
Ping Maximum Response Time (seconds)	90
Ping Period (seconds)	30
Shutdown Time Limit (seconds)	90
Startup Time Limit (seconds)	90

Process Orphaning

Enabled	False
Executable	
Executable Parameters	

Rapid-Fail Protection

"Service Unavailable" Response Type	HttpLevel
Enabled	True
Failure Interval (minutes)	5
Maximum Failures	5
Shutdown Executable	
Shutdown Executable Parameters	

Recycling

Disable Overlapped Recycle	True
Disable Recycling for Configuration Changes	True
Generate Recycle Event Log Entry	
Private Memory Limit (KB)	0
Regular Time Interval (minutes)	0
Request Limit	0
Specific Times	TimeSpan[] Array
Virtual Memory Limit (KB)	0

.NET Framework Version

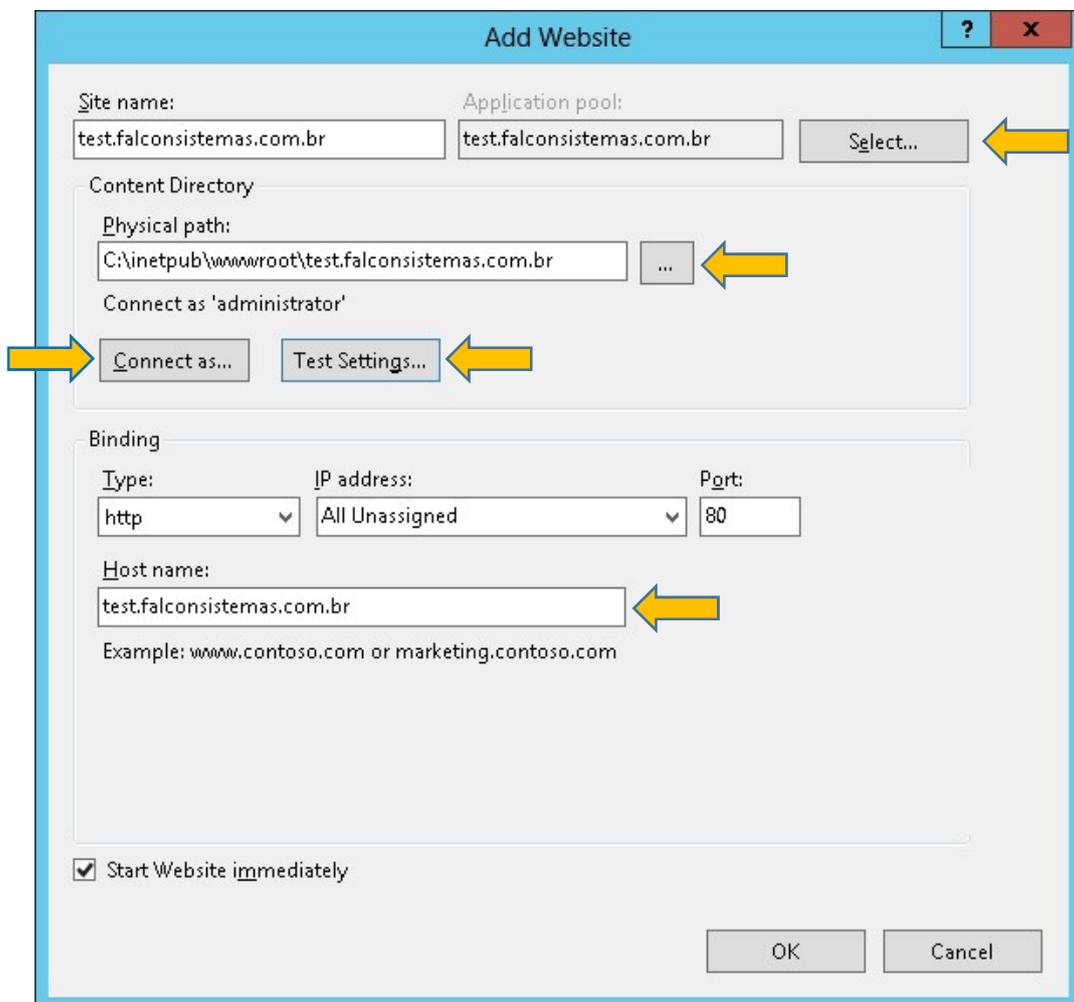
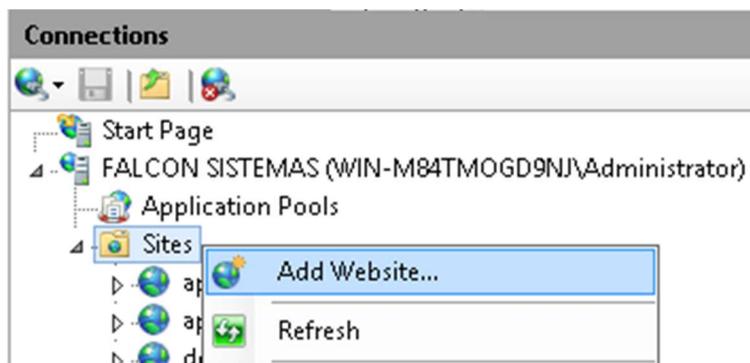
[managedRuntimeVersion] Configures the application pool to load a specific version of the .NET Framework. Selecting "No Managed Code" will cause all ASP.NET requests to fail.

OK

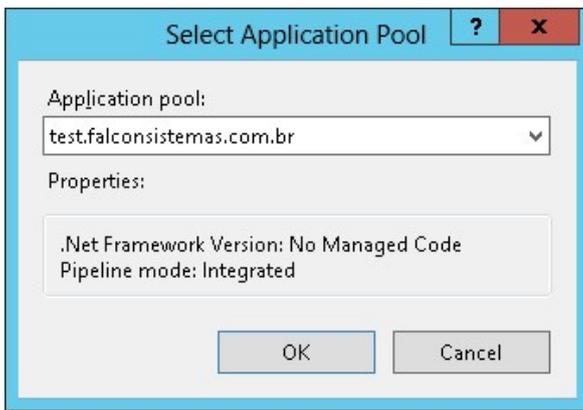
Cancel

- Em **Enable 32-Bit Applications**, você somente ira marcar false caso você tenha desenvolvido seu aplicativo em 64 Bit.
- Em **identity** você deverá adicionar um usuário com acesso a recursos do servidor, neste caso estou adicionando o administrator que tem acesso completo, mas isso depende de sua política de segurança.
- Em **Idle Time-out (minutes)**, deverá ser sempre 0, por padrão ele vem com 20 minutos, isso quer dizer, caso seu aplicativo ficar ocioso (não tiver nenhuma requisição de acesso) ele será reiniciado a cada 20 minutos, definindo 0 ele nunca será reiniciado pelo IIS, quem controla o tempo de sessão é o próprio uniGui.
- Em **Disable Overlapped Recycle** e **Disable Recl yng Configuration Changes**, as opções deverão estar como True, pois caso esteja como False, toda e qualquer alteração que seu application pool tiver, seu aplicativo será reiniciado.
- Em **Regular Time Interval (minutes)**, também deverá ser 0 (nunca recicla o application pool), por padrão ele vem 1740, isso quer dizer que a cada 1740 minutos sua aplicação será reciclada (reiniciada).

Após ter configurado o Application pool vamos configurar o site, diferente do manual do uniGui, aqui nós vamos criar um novo site para nossa aplicação (aqui também eu utilizo para cada aplicação um novo site):

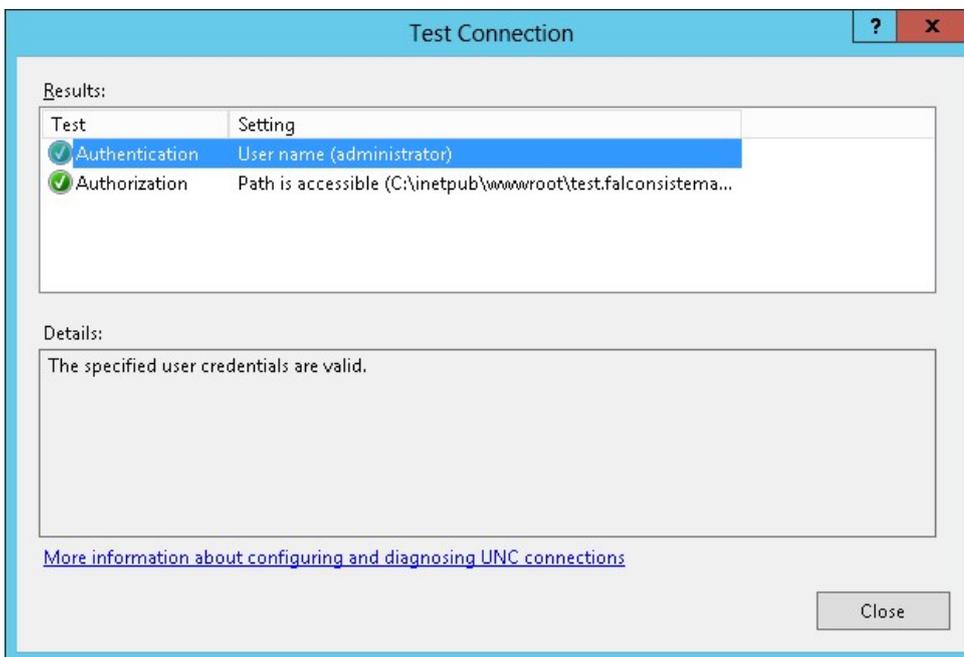


No Botão **Select...** você irá selecionar o application pool que você acabou de criar:



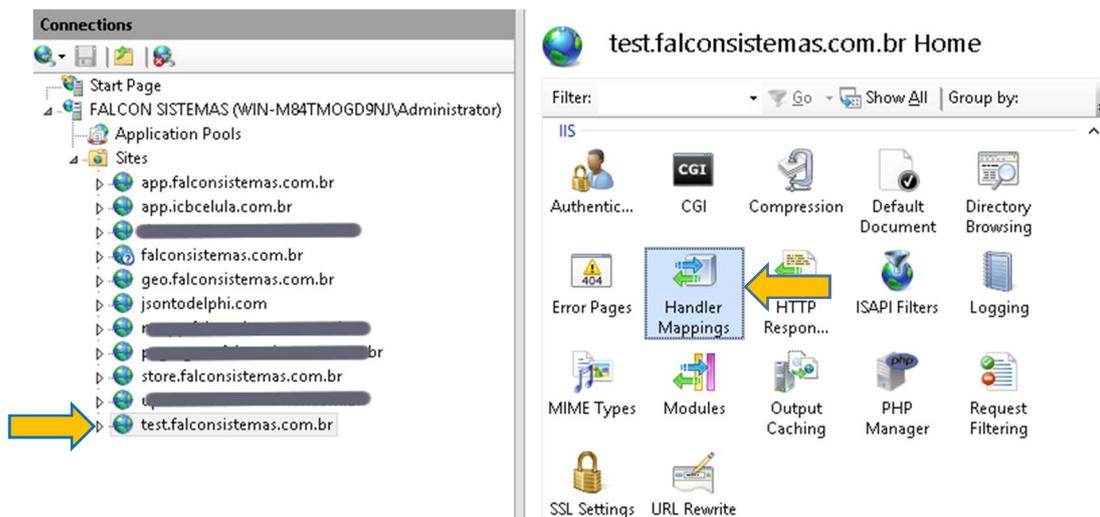
Em **Physical Path**, você irá adicionar o diretório onde sua aplicação se encontra, por padrão as aplicações são criadas em C:\inetpub\wwwroot, porém você pode definir outro diretório.

No Botão **Connect as...** Você irá definir um usuário que tenha acesso aos recursos do servidor, exemplo acesso as pastas. Logo após você irá certificar se o usuário informado passou nos testes de configurações em **Test Settings...**



Por padrão eu sempre uso a porta padrão para minhas aplicações, Porta 80, e para não haver conflito desta porta com minhas outras aplicações que também utilizam a porta 80 eu defino **Host Name**. Com isso eu consigo utilizar todas as minhas aplicações na porta 80, (quem gerencia este controle é o IIS).

Após finalizado esta etapa, selecione seu site, vamos agora habilitar o módulo ISAPI em seu site:





Handler Mappings

Use this feature to specify the resources, such as DLLs and managed code, that handle responses for specific request types.

Group by: State

Name	Path	State	Pa
Disabled			
ISAPI-dll	*.dll	Disabled	Fi
Enabled			
CGI-exe	*.exe		
PHP_via_FastCGI1	*.php		
PHP_via_FastCGI	*.php		
PHP53_via_FastCGI	*.php		
TRACEVerbHandler	*		
OPTIONSVerbHandler	*		
StaticFile	*		

- Add Managed Handler...
- Add Script Map...
- Add Wildcard Script Map...
- Add Module Mapping...
- Edit...
- Rename
- Remove
- Edit Feature Permissions...
- Revert To Parent
- View Ordered List...
- Help
- Online Help

Edit Feature Permissions

Permissions:

- Read
- Script
- Execute

OK Cancel

Depois que você habilitou o módulo ISAPI-dll, duplo clique para abrir esta tela:

Edit Module Mapping

Request path:
*.dll
Example: *.bas, wsvc.axd

Module:
IsapiModule

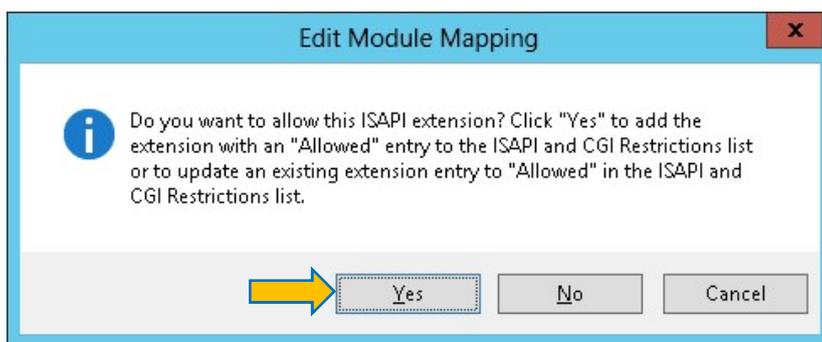
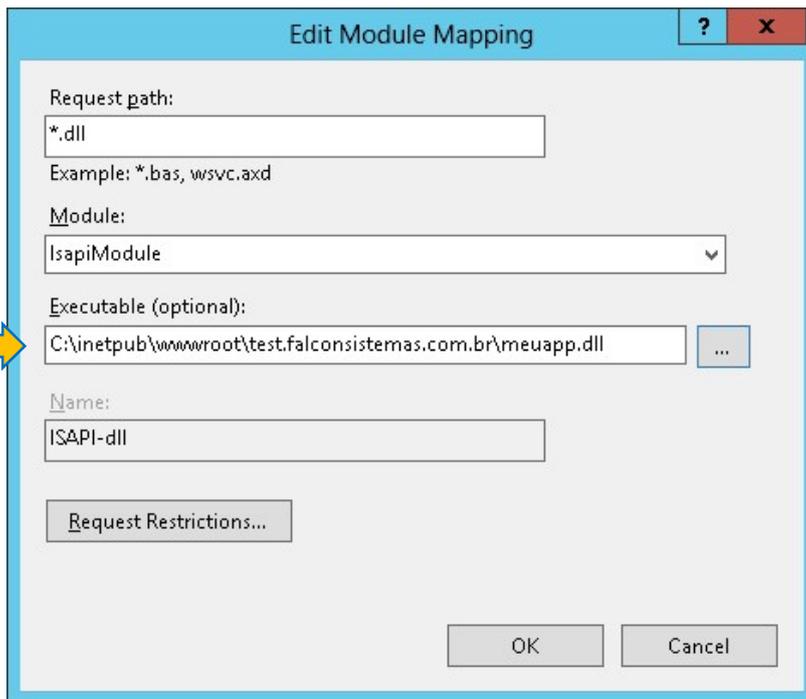
Executable (optional):
...

Name:
ISAPI-dll

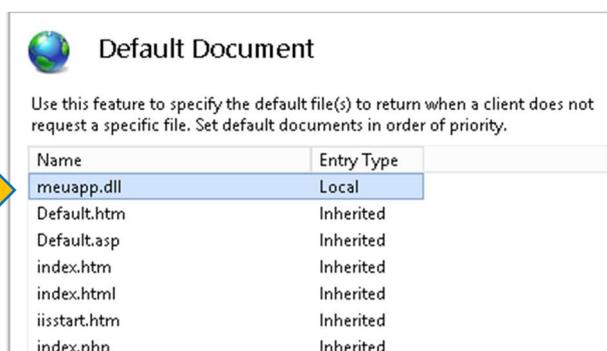
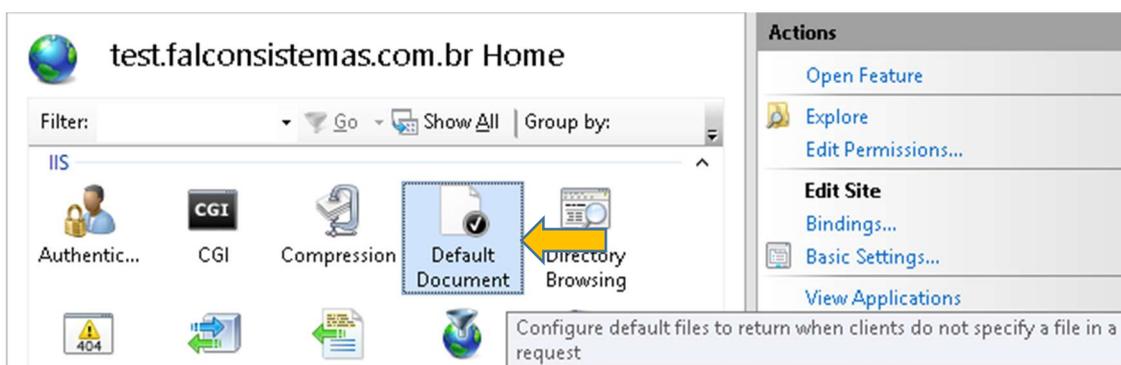
Request Restrictions...

OK Cancel

Um ponto importante que eu sempre faço, é adicionar a minha dll exclusivamente para este site em **Executable (optional)**:



Vamos agora adicionar o nome da dll por default em nosso site, isso evita ter que ficar adicionando o nome da dll na url para acessar:



Com isso finalizamos nossa configuração, basta acessar o endereço: <http://test.falconsistemas.com.br>

