OPC FOUNDATION CLIENT-SERVER SOLUTION

This document contains the I+D technical details of the MVC solution that integrates Hangfire and OPC Foundation

Contents

Requirements....2
Overview....3
Scope....5
RQ-01 | MVC Hangfire Management Application....6
RQ-02 | OPC Foundation Client-Server and DB Model Layer....8
RQ-03 | OPC Foundation Core....10
Deployment....11
Software life cycle management....14
Annexes....15
Improvements and performance....15
Hangfire application always running on IIS....15
Jobs chain of responsibility pattern....15
Additional tools used for testing....15
UaExpert....15

Doc. Version	Date	Author	Description
1.0	2025-10-09	Iker Elg. Alzaa	First version of the release

Requirements

The next table shows the requirements for the solution described.

Requirement	Description
RQ-001 | MVC application with Hangfire	MVC application integrating Hangfire (including a console and Windows Server project for Hangfire.
RQ-002 | OPC Foundation client-server and DB model layer	Custom client-server OPC libraries and jobs, as well as the model and database access.
RQ-003 | OPC Foundation core	Projects with the source code containing the SDK & Stack of OPC Foundation.

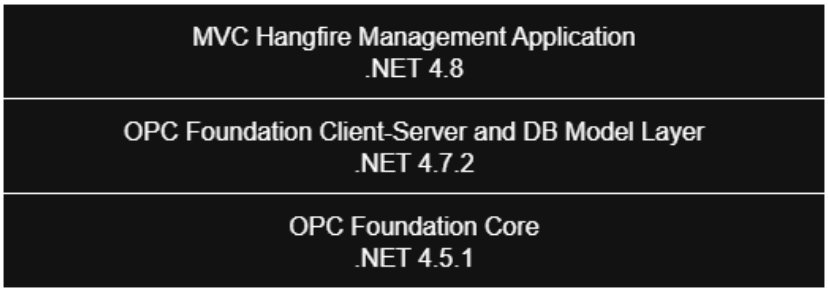
Overview

This document contains the technical details of the MVC application that integrates Hangfire 1.8.21 and OPC Foundation, running on IIS/SQL Server.

Solutions:

- **Hangfire-OPC-Labs_.NET-4.8:** Contains the MVC application with Hangfire (including a console and Windows Server project for Hangfire).
- **OPC-Foundation-Labs-Server-Client-.Net-4.8:** Containing the projects that make up the custom client-server OPC libraries and jobs, as well as the model and database access.
- **OPC-Foundation-Labs-.Net-4.8:** The OPC Foundation projects and core code (SDK & Stack).

The next image shows solution hierarchy and management relation.



The application provides the next capabilities:

- Start/stop a background job that runs an OPC server, where an XML configuration file tells it which nodes it should manage.
- Start/stop a background job that runs an OPC client (connecting to the server), where an XML configuration file tells it which nodes it should subscribe to in a monitored manner. As soon as it is notified of a change from the server, it dumps the new values for those nodes into the database on the server (with an "idProcess" flag).
- Add a recurring job that executes an OPC client (connecting to the server) where an XML configuration file indicates the server nodes to which it should write values.
- Add a recurring job that executes an OPC client (connecting to the server) where an XML configuration file indicates the server nodes from which it should read values.
- View logs in real time using a text buffer.

  NOTE: The XML configuration file used for jobs is the OPC one.

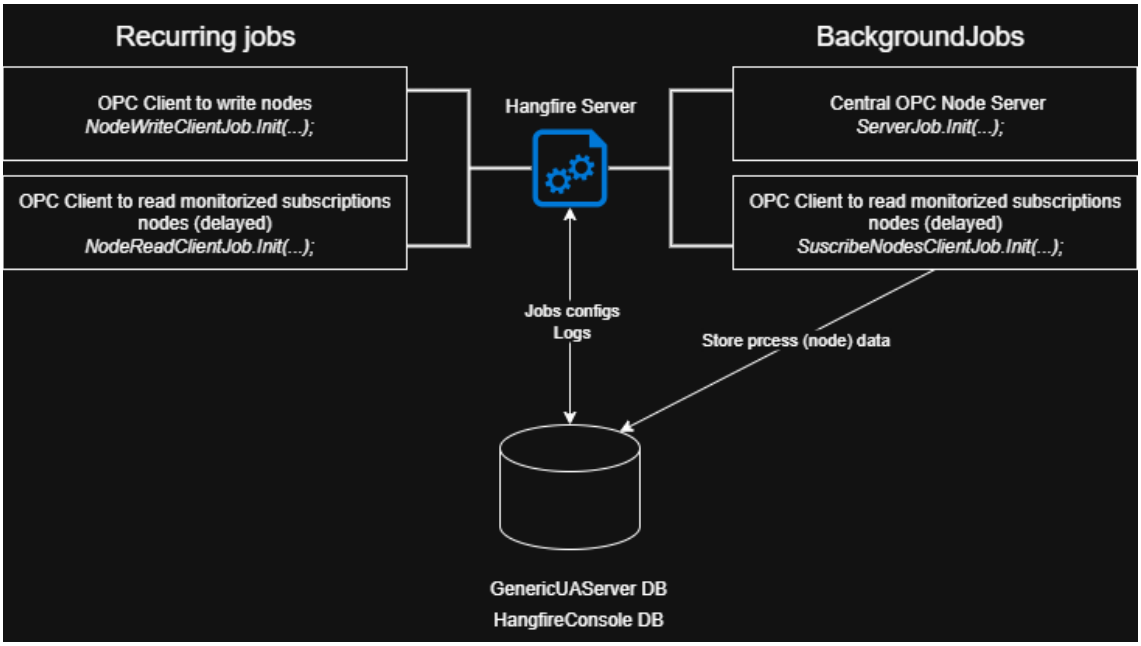
This latest version of the Hangfire server includes:

- Cancellation Tokens to stop jobs in the background.
- All jobs are implemented using the chain of responsibility pattern and can be intercepted using action filters

Basically, the solution creates four jobs:

- First, two BackgroundJob jobs: one for the OPC server (ServerJob) and the other for the client for monitored subscriptions (SubscribeNodesClientJob).
- Then, it creates two more RecurringJob jobs: one to simulate the infrastructure with random values (NodeWriteClientJob) and the other to read the values directly without subscriptions (NodeReadClientJob).

Here the diagram with the solution architecture overvie.



Scope

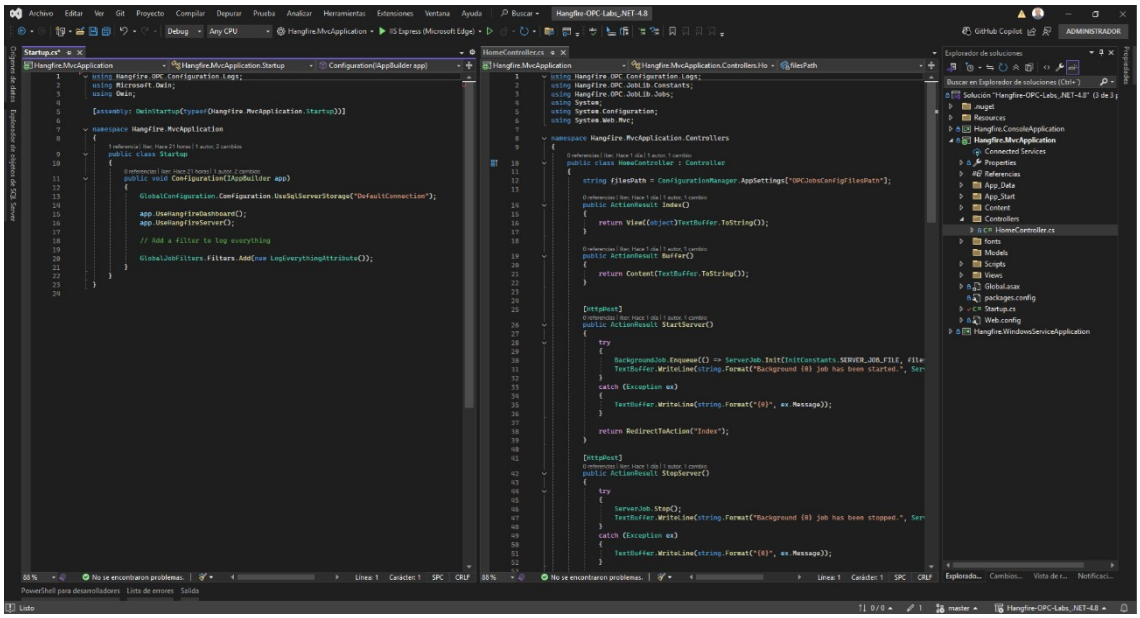
Hangfire is an open-source library for .NET applications that facilitates the execution of background jobs. It allows you to schedule jobs to run at a specific time, recurringly, or immediately, all without the need for external services, and with an administration panel to monitor their status.

OPC technologies are created to allow information to be easily and securely exchanged between diverse platforms from multiple vendors and to allow seamless integration of those platforms without costly, time-consuming software development. This frees engineering resources to do the more important work of running your business.

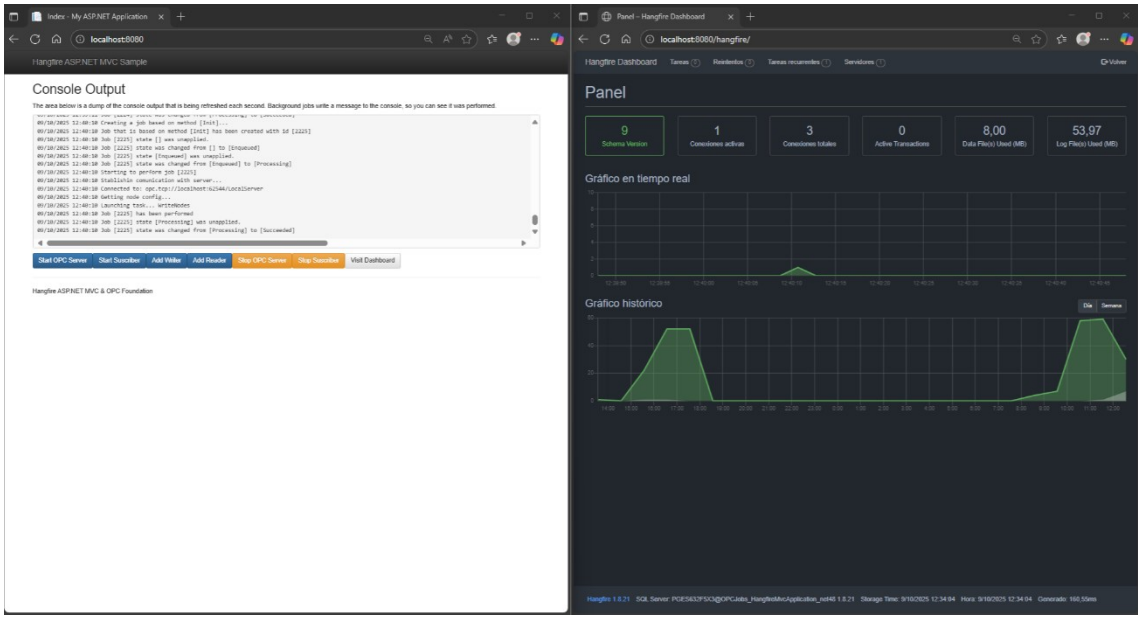
Using the MVC application, it is possible to provide a web-accessible UI for managing OPC jobs and viewing logs.

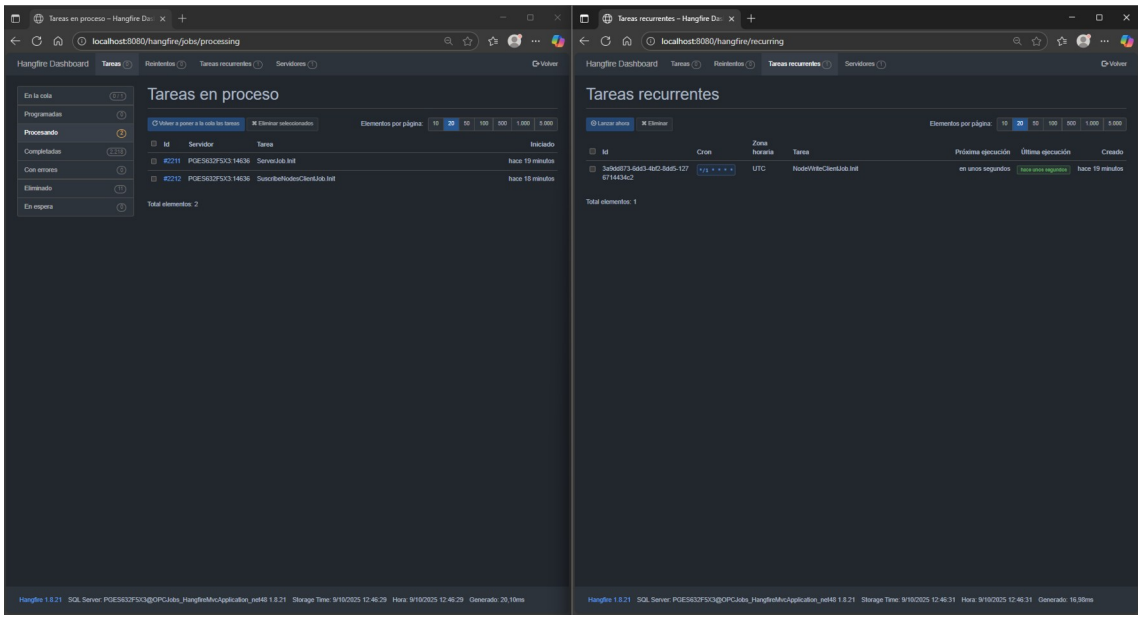
RQ-01 | MVC Hangfire Management Application

This solution contains a standard .NET 4.8 MVC project by which the Hangfire server is managed. It includes two projects, a console application for tests and a windows service project.



Here the screens of the application:



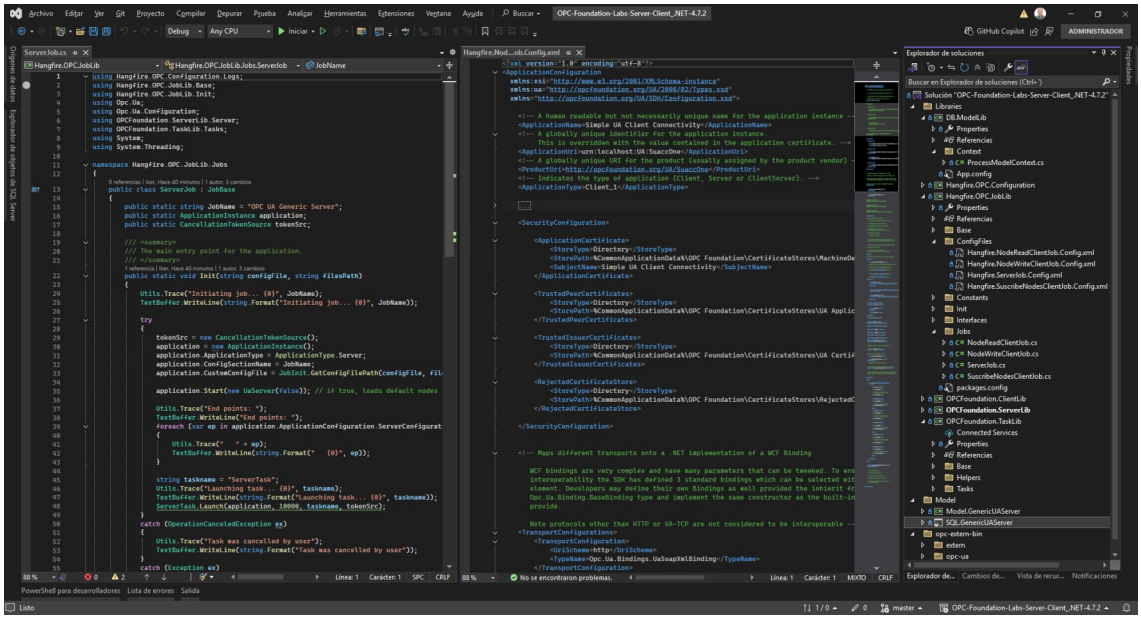

Tareas en proceso – Hangfire Da
localhost:8080/hangfire/jobs/processing
Hangfire Dashboard
Tareas
Reintentos
Tareas recurrentes
Servidores
Volver
En la cola
Programadas
Procesando
Completadas
Con errores
Eliminado
En espera
Tareas en proceso
Volver a poner a la cola las tareas
Eliminar seleccionados
Elementos por página:
Id
Servidor
Tarea
Iniciado
#2211
PGES632FSX3:14636
ServerJob.Init
hace 19 minutos
#2212
PGES632FSX3:14636
SuscribeNodesClientJob.Init
hace 18 minutos
Total elementos: 2
Hangfire 1.8.21
SQL Server: PGES632FSX3@OPCJobs_HangfireMvcApplication_net48 1.8.21
Storage Time: 9/10/2025 12:46:29
Hora: 9/10/2025 12:46:29
Generado: 20.10ms
Tareas recurrentes – Hangfire Da
localhost:8080/hangfire/recurring
Hangfire Dashboard
Tareas
Reintentos
Tareas recurrentes
Servidores
Volver
Tareas recurrentes
Lanzar ahora
Eliminar
Elementos por página:
Id
Cron
Zona horaria
Tarea
Próxima ejecución
Última ejecución
Creado
3a9dd873-6dd3-4bf2-8dd5-1276714434c2
UTC
NodeWriteClientJob.Init
en unos segundos
hace unos segundos
hace 19 minutos
Total elementos: 1
Hangfire 1.8.21
SQL Server: PGES632FSX3@OPCJobs_HangfireMvcApplication_net48 1.8.21
Storage Time: 9/10/2025 12:46:31
Hora: 9/10/2025 12:46:31
Generado: 16.98ms

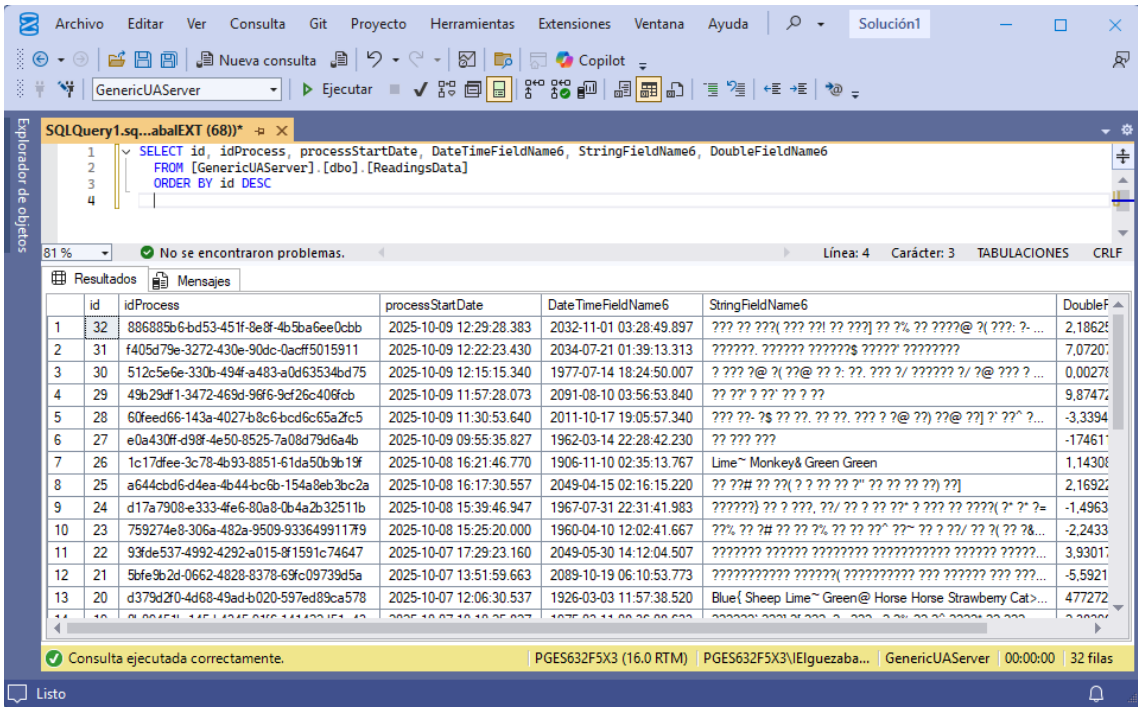
RQ-02 | OPC Foundation Client-Server and DB Model Layer

This solution contains a standard .NET 4.7.2 projects regarding the client-server layer to interact with OPC Foundation together with the database and model projects to store data.

The solution is divided in the next modules or assemblies:

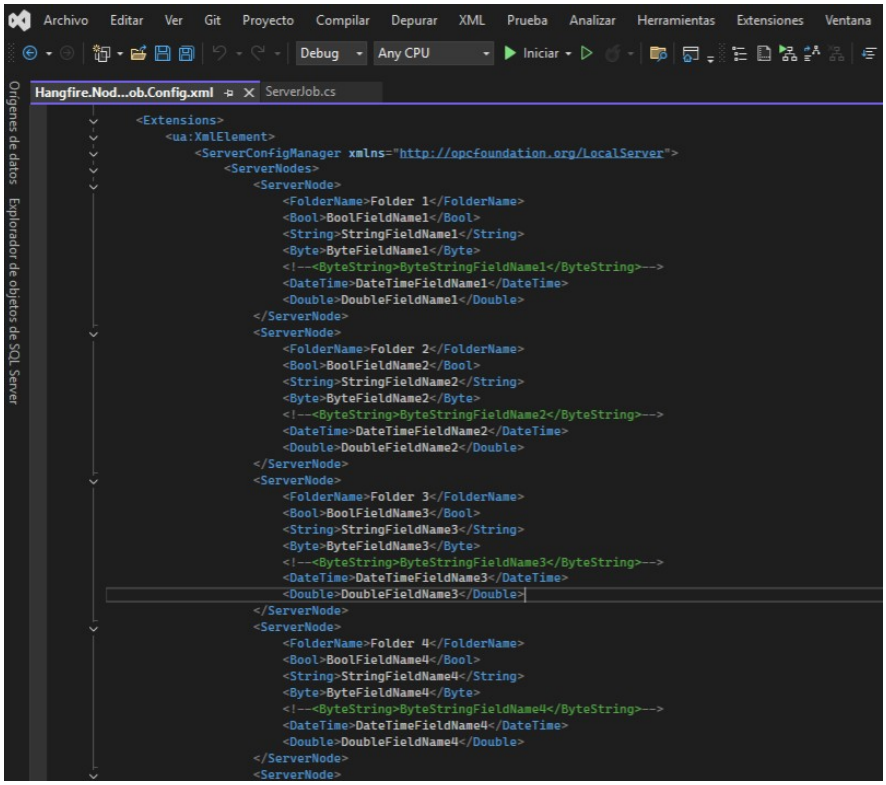
- Client-server layer
  - DB.ModelLib → Database interaction context (EntityFramework)
  - Hangfire.OPC.Configuration → Hangfire default Job configuration
  - Hangfire.OPC.JobLib → Job definitions managed by the Hangfire process
  - OPCFoundation.ClientLib → Opc client procedures and clases to interact with the SDK / Stack
  - OPCFoundation.ServerLib → Opc server procedures and clases to interact with the SDK / Stack
  - OPCFoundation.TaskLib → Tasks managed by the jobs
- Database and model
  - Model.GenericUAServer → GenericUAServer DB model to store the Job process and related data
    Note: Hangfire tables are created automatically
  - SQL.GenericUAServer→ Database to store node data





	id	idProcess	processStartDate	DateTimeFieldName6	StringFieldName6	DoubleF
1	32	886885b6-bd53-451f-8e8f-4b5ba6ee0cbb	2025-10-09 12:29:28.383	2032-11-01 03:28:49.897	??? ?? ???(??? ??! ?? ???] ?? ?% ?? ????@ ?(???: ?- ...	2,1862
2	31	f405d79e-3272-430e-90dc-0acff5015911	2025-10-09 12:22:23.430	2034-07-21 01:39:13.313	??????. ?????? ??????$????? ????????	7,0720
3	30	512c5e6e-330b-494f-a483-a0d63534bd75	2025-10-09 12:15:15.340	1977-07-14 18:24:50.007	? ??? ?@ ?(??@ ?? ?: ??. ??? ?/ ?????? ?/ ?@ ??? ? ...	0,0027
4	29	49b29df1-3472-469d-96f6-9cf26c406fcb	2025-10-09 11:57:28.073	2091-08-10 03:56:53.840	?? ?? ? ?? ?? ? ??	9,8747
5	28	60feed66-143a-4027-b8c6-bcd6c65a2fc5	2025-10-09 11:30:53.640	2011-10-17 19:05:57.340	??? ??- ?$?? ??. ?? ??. ??? ? ?@ ??) ??@ ??] ?` ??^ ?...	-3,3394
6	27	e0a430ff-d98f-4e50-8525-7a08d79d6a4b	2025-10-09 09:55:35.827	1962-03-14 22:28:42.230	?? ??? ???	-17461
7	26	1c17dfee-3c78-4b93-8851-61da50b9b19f	2025-10-08 16:21:46.770	1906-11-10 02:35:13.767	Lime~ Monkey& Green Green	1,1430
8	25	a644cbd6-d4ea-4b44-bc6b-154a8eb3bc2a	2025-10-08 16:17:30.557	2049-04-15 02:16:15.220	?? ??# ?? ??(? ? ?? ?? ?" ?? ?? ?? ??) ??]	2,1692
9	24	d17a7908-e333-4fe6-80a8-0b4a2b32511b	2025-10-08 15:39:46.947	1967-07-31 22:31:41.983	??????} ?? ? ???. ??/ ?? ? ?? ??* ? ??? ?? ????(?* ?* ?=	-1,4963
10	23	759274e8-306a-482a-9509-933649911ff9	2025-10-08 15:25:20.000	1960-04-10 12:02:41.667	??% ?? ?# ?? ?? ??% ?? ?? ??^ ??~ ?? ? ??/ ?? ?(?? ?&...	-2,2433
11	22	93fde537-4992-4292-a015-8f1591c74647	2025-10-07 17:29:23.160	2049-05-30 14:12:04.507	??????? ?????? ???????? ??????????? ?????? ?????...	3,9301
12	21	5bfe9b2d-0662-4828-8378-69fc09739d5a	2025-10-07 13:51:59.663	2089-10-19 06:10:53.773	??????????? ??????(?????????? ??? ?????? ??? ???...	-5,5921
13	20	d379d2f0-4d68-49ad-b020-597ed89ca578	2025-10-07 12:06:30.537	1926-03-03 11:57:38.520	Blue{ Sheep Lime~ Green@ Horse Horse Strawberry Cat>...	477272

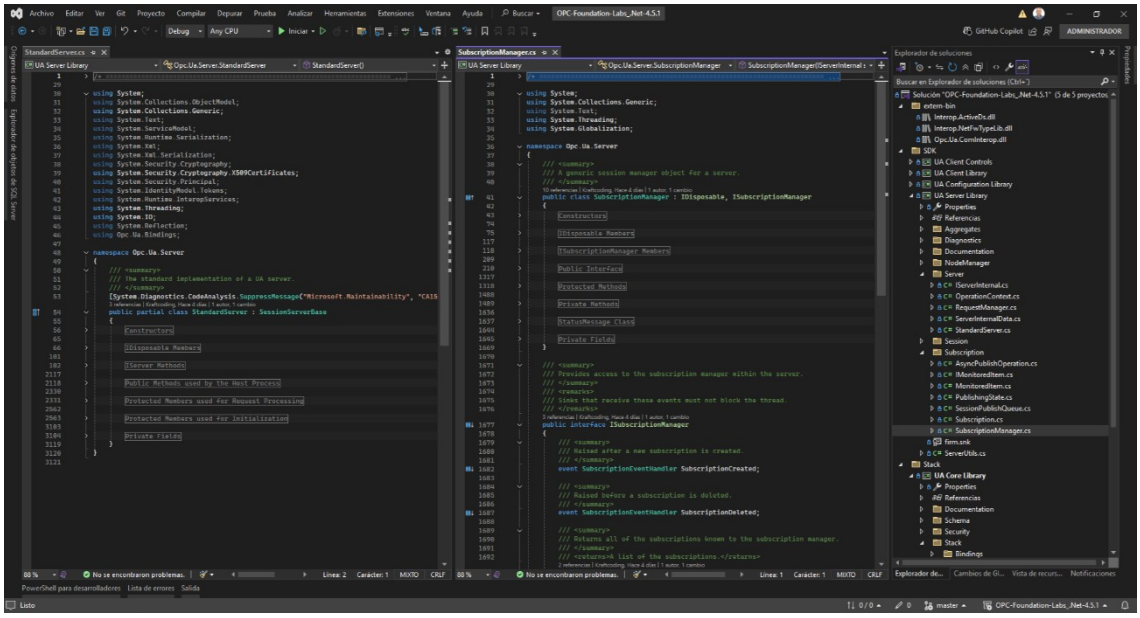
The OPC nodes are defined like this.



RQ-03 | OPC Foundation Core

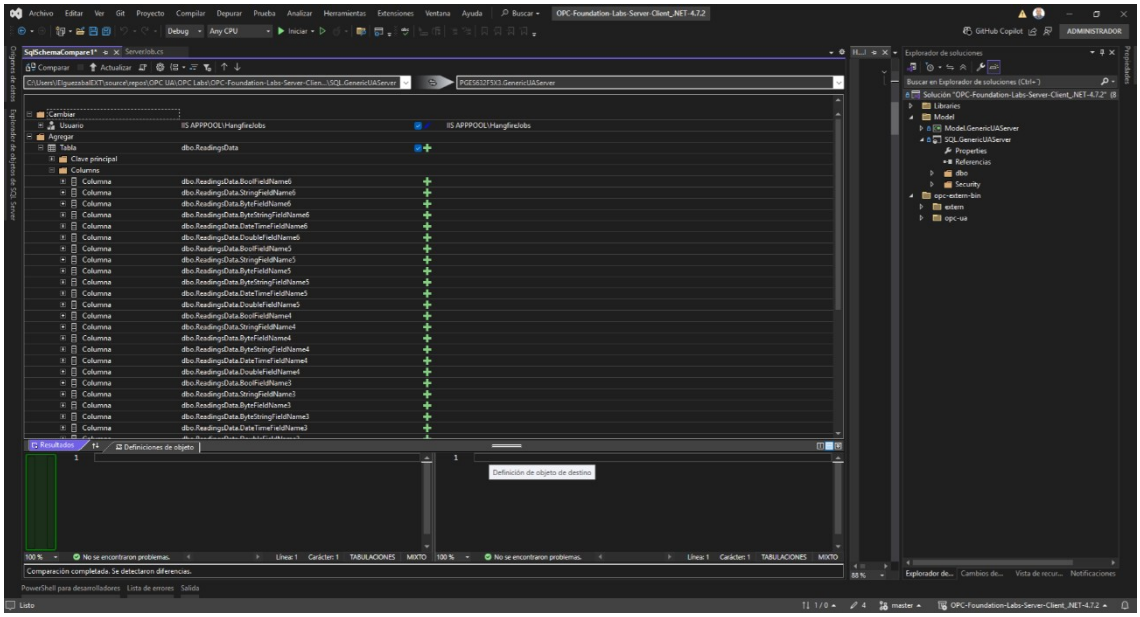
This solution contains a standard .NET 4.5.1 projects with the fundamental source code of OPC Foundation.

Contains the SDK and Stack, as we can see in the next image.

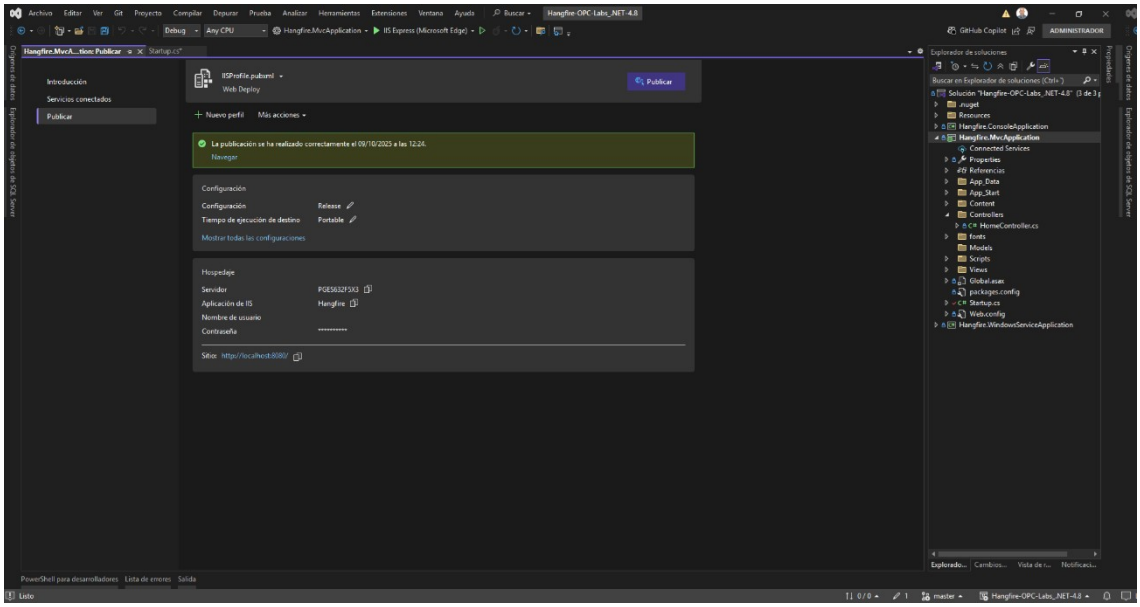


Deployment

To deploy the solution first launch the compare schema.



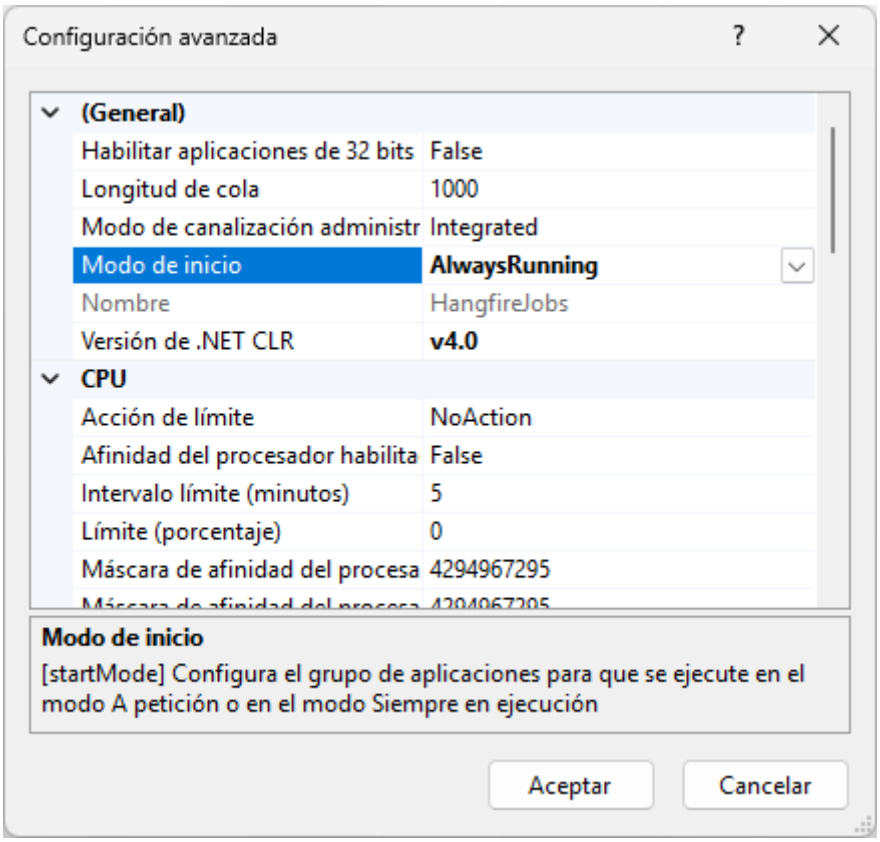
Then publish the MVC application.

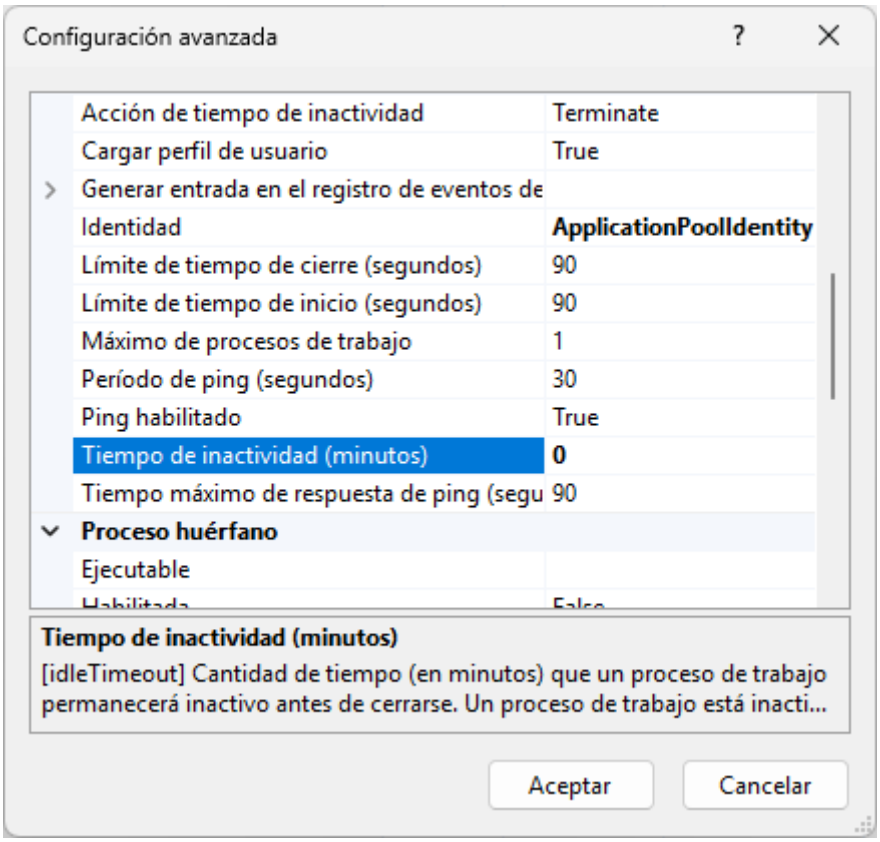


It is needed to give permissions to the IIS app-pool process to use TCP.

- netsh http add urlacl url=http://+:62543/ user=Everyone
- netsh http add urlacl url=http://+:62543/LocalServer/ user=IIS APPPOOL\ HangfireJobs

IIS and app-pool configuration should be like this.





Application Pool Set "Preload Enabled". To set the application pool to "Preload Enabled" in IIS, follow these steps:

- Open the applicationHost.config file located in the %WINDIR%\system32\inetsrv\config\applicationHost.config directory.
- Locate the application pool setting and set preloadEnabled to true.
- Restart IIS to apply the changes.
- Optionally, configure warm-up requests to ensure all necessary components are loaded during the initialization phase.

This configuration allows IIS to send a "fake" request to the application when the associated application pool starts up, ensuring that the application is warmed up before any real traffic is received. This is particularly useful for applications with heavy startup processes, such as those requiring multiple service initializations or database connections.

Software life cycle management

TO DO.

Annexes

Improvements and performance

Hangfire application always running on IIS

Look up if there is any way to ensure that Hangfire application is always running on IIS.

Jobs chain of responsibility pattern

Must be look up if further integration could be attained between the OPC Jobs and Hangfire chain of responsibility patterns.

Additional tools used for testing

UaExpert

For initial steps of development and testing UaExpert client application was used.