

TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE APIZACO

NOMBRE DEL PROYECTO

**SISTEMA WEB DE GENERACIÓN DE DICTÁMENES A PARTIR
DE ANÁLISIS VOLUMÉTRICOS DE DATOS**

**TITULACIÓN INTEGRAL: INFORME DE
RESIDENCIA PROFESIONAL**

**PARA OBTENER EL TÍTULO PROFESIONAL DE:
INGENIERIA EN TECNOLOGÍAS DE LA INFORMACIÓN Y
COMUNICACIONES**

PRESENTA

ADÁN LOERA SÁNCHEZ

NOMBRE DEL ASESOR

PERFECTO MALAQUIAS QUINTERO FLORES

APIZACO, TLAX, ABRIL 2026



Educación
Secretaría de Educación Pública



TECNOLÓGICO
NACIONAL DE MÉXICO



Instituto Tecnológico de Apizaco

Sistemas y Computación

Tzompantepec, Tlaxcala, **24/marzo /2025**

OFICIO No: SC/OPV/0094/25

ASUNTO: Liberación de proyecto
para titulación integral

ISRAEL JORGE SÁNCHEZ SÁNCHEZ
JEFE DEL DEPTO. DE DIVISIÓN DE ESTUDIOS PROFESIONALES
P R E S E N T E.

POR ESTE MEDIO LE INFORMO QUE HA SIDO LIBERADO EL SIGUIENTE PROYECTO PARA LA TITULACIÓN INTEGRAL.

NOMBRE DEL EGRESADO:	LOERA SANCHEZ ADAN
CARRERA:	INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES
NO. DE CONTROL:	18370962
NOMBRE DEL PROYECTO:	Sistema Web de Generación de Dictámenes a partir de Análisis Volumétricos de Datos.
PRODUCTO U OPCIÓN:	TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL

AGRADEZCO DE ANTEMANO SU VALIOSO APOYO EN ESTA IMPORTANTE ACTIVIDAD PARA LA FORMACIÓN PROFESIONAL DE NUESTROS EGRESADOS.

ATENTAMENTE

Excelencia en Educación Tecnológica
Pensar para Servir, Servir para Triunfar

MIQUELINA SÁNCHEZ PULIDO
JEFA DEL DEPTO. DE SISTEMAS Y COMPUTACIÓN

c.c.p. División de Estudios Profesionales
c.c.p. Depto. de Servicios Escolares
c.c.p. Archivo
MSP/mrol*



SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLÓGICO NACIONAL
DE MÉXICO
INSTITUTO TECNOLÓGICO DE APIZACO
**DEPARTAMENTO DE
SISTEMAS Y COMPUTACIÓN**

PERFECTO MALAQUÍAS QUINTERO FLORES
ASESOR INTERNO



2025
Año de
**La Mujer
Indígena**

Av. Instituto Tecnológico No. 418, San Andrés Ahuashuatepec,
Municipio de Tzompantepec, Tlaxcala, Mex. C.P. 90491 Tel. (241)
4172010 Ext. 144, e-mail: sistemas@apizaco.tecnm.mx | tecnm.mx |
apizaco.tecnm.mx





Tzompantepec, Tlaxcala, 12/Marzo/2026

ADÁN LOERA SÁNCHEZ
EGRESADO(A) DE LA CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES
CON NÚMERO DE CONTROL 18370962.
P R E S E N T E.

Por este medio, me permito informar a usted que por aprobación de la comisión revisora asignada para valorar el trabajo que mediante la opción **TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL** y cuyo tema es: **"SISTEMA WEB DE GENERACIÓN DE DICTÁMENES A PARTIR DE ANÁLISIS VOLUMÉTRICOS DE DATOS"**, conforme a lo establecido en el procedimiento para la obtención del Título Profesional en el Instituto Tecnológico, la División de Estudios Profesionales a mi cargo le otorga la:

AUTORIZACIÓN DE IMPRESIÓN

Debiendo entregar **1 USB** del mismo, apropiadamente empastados, a la brevedad, para presentar su Acto de Recepción Profesional.

Sin otro particular por el momento me despido.

ATENTAMENTE

Excelencia en Educación Tecnológica®

Pensar para Servir, Servir para Triunfar®

CECILIA YARELY ARREGUÍN LARA
JEFA DE LA DIVISIÓN DE ESTUDIOS PROFESIONALES



SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLÓGICO NACIONAL
DE MÉXICO
INSTITUTO TECNOLÓGICO DE APIZACO
**DIVISIÓN DE
ESTUDIOS PROFESIONALES**

ccp. Archivo
CYAL/Gvm*



Agradecimientos

Quiero expresar mi más profundo agradecimiento a las personas que hicieron posible la culminación de mi carrera. A mi papá, Manuel, por ser mi mayor ejemplo de esfuerzo y perseverancia, cuya guía y apoyo incondicional en cada paso de este camino fueron la base que me permitió seguir adelante; a mi mamá, Susana, por su amor infinito, y por creer en mí siempre, y a mi pareja, Melanie, por ser mi compañera incondicional, brindarme su apoyo constante y celebrar a mi lado cada pequeño logro. Gracias a los tres por ser el pilar fundamental que me permitió cumplir este gran sueño.

Índice	
Resumen.....	2
Introducción	3
Problemas a Resolver.....	5
Objetivo general	7
Objetivos específicos.....	7
Justificación	8
Marco Teorico	9
Desarrollo	43
Resultados y conclusiones	58
Referencias	59

Resumen

El proyecto "Sistema Web para la generación de Dictámenes a partir de Análisis Volumétricos de Datos" es una propuesta de solución por parte de la empresa ATISoft, ante la nueva Norma gubernamental de Controles Volumétricos. Durante el desarrollo de este proyecto se hace énfasis en la creación de un Sistema Web capaz de cubrir los requerimientos propuestos, en donde empresas que se dediquen a la producción, almacenamiento, transporte y comercialización de hidrocarburos y petrolíferos, lleven un registro y control de todas las actividades afines. Para lograr esto, se han seguido diferentes etapas en el desarrollo del proyecto, como la obtención de requerimientos, la configuración y adecuación de entidades, la creación de módulos de registro para cada caso de uso y la implementación de una base de datos para almacenar los registros generados por el usuario. Además, se han presentado los módulos y su funcionalidad tanto en el back-end como en el front-end.

Introducción

El Gobierno Mexicano emitió en 2020 una nueva norma sobre Controles Volumétricos de Hidrocarburos y Petrolíferos, la cual entró en vigor en toda la República Mexicana el 1 de enero de 2022. La norma tiene como objetivo garantizar la transparencia y legalidad en las transacciones de hidrocarburos y petrolíferos en México, mediante la implementación de medidas para controlar las cantidades de estos productos que se producen, almacenan, transportan y comercializan en el país.

Para ello, la norma establece que todas las empresas que se dedican a la producción, almacenamiento, transporte y comercialización de hidrocarburos y petrolíferos deben llevar un registro detallado de las cantidades que manejan, así como instalar equipos de medición y control volumétrico. El incumplimiento de esta norma podría agravar la problemática de la falta de control sobre las cantidades de hidrocarburos y petrolíferos que se producen, almacenan, transportan y comercializan en México, lo

que a su vez tendría consecuencias negativas en la economía, la sociedad y el Estado en general, como el incremento de la evasión fiscal, la afectación a la seguridad energética y el aumento de la corrupción.

Ante esta problemática, se propone crear una plataforma web que permita llevar un control correspondiente a los dictámenes emitidos por cada empresa involucrada en este giro comercial. Asimismo, se deberá crear sistemas de medición capaces de generar registros de volumen de manera automática, encriptada y con una interfaz de comunicación que permita llevar un control volumétrico adecuado.

Descripción de empresa

ATISoft es una empresa especializada en el manejo de datos en tiempo real, en donde la integración de la tecnología de Industria 4.0 permite crear plataformas tecnológicas adaptadas a los requerimientos de sus clientes. Su trabajo se respalda detrás de la tecnología de OSIsoft, Alteryx, Microsoft, Esri y Capella Space. Los principales mercados de la empresa ATISoft se basan en la Minería, Manufactura, Gas y Petróleo, Energía y Farmacéutica. ATISoft además fija su estrategia de crecimiento en cuatro ramas, como lo son; Industria IoT, Conected Worker, Location of Things y Operational Gis, lo que les permite acceso a diversas herramientas que automatizan los procesos operativos. Las certificaciones que avalan a la compañía ATISoft son ISO/IEC 29110-4 y NMX-I-059/02/MOPROSOFT.



Ubicacion

Montecito 38 piso 28-16 Col. Napoles. Benito Juarez CDMX.

Y

Av. Xicohtencatl #407 int. 2 esquina 5 de mayo, CP.90300 Apizaco, Tlaxcala.

Correo

contacto@atisoft.com.mx

Telefono

+52 (55) 5448 5874

Problemas a Resolver

En México, las empresas del sector de hidrocarburos están obligadas a implementar controles volumétricos para registrar con precisión los volúmenes de entrada, salida y existencias de hidrocarburos y petrolíferos, conforme a los Anexos 30, 31 y 32 de la Resolución Miscelánea Fiscal OMAWWW.SAT.GOB.MX. Este proceso implica desafíos significativos debido al manejo de grandes volúmenes de datos y la necesidad de cumplir con estrictas normativas fiscales.

Los principales problemas a abordar son:

Automatización del Procesamiento de Datos: Actualmente, la recopilación y análisis de datos volumétricos se realizan de manera manual, lo que es laborioso y propenso a errores. Se busca desarrollar un sistema que automatice la extracción, importación y procesamiento de estos datos, facilitando la generación de dictámenes precisos y oportunos.

Cumplimiento Normativo: Las empresas deben asegurarse de que sus sistemas de control volumétrico cumplan con las especificaciones técnicas y legales establecidas por el Servicio de Administración Tributaria (SAT) OMAWWW.SAT.GOB.MX. Esto incluye la correcta operación de los equipos y programas informáticos utilizados para llevar los registros volumétricos.

Facilidad de Uso: Es esencial que el sistema sea intuitivo y accesible para los usuarios, permitiendo una rápida adaptación y minimizando la curva de aprendizaje.

Gestión de Roles y Permisos: Implementar un esquema de roles con permisos definidos garantizará que cada usuario acceda únicamente a las funciones correspondientes a su perfil, mejorando la seguridad y eficiencia operativa.

Optimización del Rendimiento: El sistema debe ser capaz de manejar grandes volúmenes de datos sin comprometer el rendimiento del servidor, evitando la saturación y asegurando tiempos de respuesta adecuados.

Objetivo general

Implementar un sistema de control para los registros de Controles Volumétricos de Hidrocarburos y Petrolíferos en empresas mexicanas para garantizar la transparencia y legalidad en las transacciones de estos productos.

Objetivos específicos

Diseño de la Plataforma Web

- Crear una interfaz intuitiva y accesible que permita a los usuarios registrar en tiempo real las cantidades de hidrocarburos y petrolíferos.
- Implementar un sistema de almacenamiento de datos seguro y confiable para garantizar la integridad de la información.
- Asegurar que la plataforma cumpla con las normativas vigentes en México para el manejo de hidrocarburos y petrolíferos.

Desarrollo de Sistemas de Medición y Control

- Diseñar e instalar dispositivos de medición que automaticen el registro de volúmenes de hidrocarburos y petrolíferos.
- Implementar un sistema de encriptación de datos para proteger la información durante su transmisión.
- Crear una interfaz de comunicación que permita el acceso a la información de manera transparente y que facilite auditorías.

Capacitación del Personal

- Desarrollar un programa de capacitación integral para el personal de las empresas que utilizarán el sistema.
- Incluir módulos sobre el manejo correcto de los controles volumétricos y la importancia de la legalidad en las transacciones.
- Evaluar el aprendizaje del personal mediante pruebas y retroalimentación para asegurar la correcta aplicación de los conocimientos.

Evaluación y Mejora del Sistema

- Establecer un calendario de evaluaciones periódicas para revisar el funcionamiento del sistema de control volumétrico.
- Recoger feedback de los usuarios para identificar áreas de mejora y ajustes necesarios en el sistema.
- Implementar cambios basados en las evaluaciones para optimizar la eficiencia y efectividad del control de hidrocarburos y petrolíferos.

Justificación

El presente proyecto busca la necesidad de implementar un sistema de gestión de Controles Volumétricos de Hidrocarburos y Petrolíferos en empresas mexicanas que se dedican a la producción, almacenamiento, transporte y comercialización de estos productos. La normativa actual establece la obligación de llevar un registro detallado de las cantidades que se manejan y la instalación de equipos de medición y control volumétrico. Sin embargo, se ha detectado una falta de cumplimiento de estas normas por parte de algunas empresas, lo que puede tener consecuencias negativas en la economía, la sociedad y el Estado en general. Por ello, se plantea la necesidad de implementar una plataforma web y sistemas de medición que permitan llevar un control adecuado y en tiempo real saber de las

cantidades de hidrocarburos y petrolíferos, de manera encriptada y con una interfaz de comunicación que garantice la transparencia y legalidad en las transacciones. Este proyecto, por tanto, ofrece una alternativa viable para solucionar esta problemática y contribuir a la transparencia y legalidad en el sector energético mexicano.

Marco Teorico

Introducción al Control Volumétrico

El control volumétrico se refiere a la implementación de sistemas y procedimientos destinados a medir y registrar con precisión los volúmenes de hidrocarburos y petrolíferos en las distintas etapas de su cadena de valor, desde la producción hasta la comercialización. Este mecanismo es esencial para garantizar la transparencia, trazabilidad y cumplimiento de las obligaciones fiscales en el sector energético.

2. Marco Normativo en México

En México, la regulación del control volumétrico está fundamentada en diversas disposiciones legales y fiscales que buscan asegurar la correcta gestión y supervisión de los hidrocarburos.

2.1. Código Fiscal de la Federación (CFF)

El artículo 28, fracción I, apartado B del CFF establece la obligación de los contribuyentes que manejen hidrocarburos y petrolíferos de contar con equipos y programas informáticos para llevar controles volumétricos como parte de su contabilidad. Estos sistemas deben registrar los volúmenes de entrada, salida y existencias de los productos, garantizando su precisión y confiabilidad.

2.2. Resolución Miscelánea Fiscal (RMF)

La RMF complementa las disposiciones del CFF mediante anexos específicos que detallan los requisitos técnicos y operativos para la implementación de los controles volumétricos:

Anexo 30: Establece las especificaciones técnicas de funcionalidad y seguridad que deben cumplir los equipos y programas informáticos utilizados para llevar a cabo los controles volumétricos.

KERNOTEK.MX

Anexo 31: Define las características y obligaciones de los proveedores autorizados para ofrecer servicios relacionados con los sistemas de control volumétrico.

Anexo 32: Detalla los lineamientos para la generación y envío de reportes mensuales al Servicio de Administración Tributaria (SAT), asegurando la correcta transmisión de la información registrada.

3. Procesos Clave del Control Volumétrico

La implementación efectiva del control volumétrico implica una serie de procesos interrelacionados que aseguran la integridad y precisión de los datos manejados:

3.1. Instalación de Equipos de Medición

Se requiere la instalación de dispositivos como caudalímetros, sensores de nivel y temperatura en puntos estratégicos de la cadena de suministro. Estos equipos capturan datos en tiempo real sobre los volúmenes manejados, proporcionando información esencial para el registro contable y fiscal.

3.2. Integración de Sistemas Informáticos

Los datos recolectados por los equipos de medición deben ser procesados y almacenados mediante programas informáticos que cumplan con las especificaciones técnicas establecidas en el Anexo 30 de la RMF. Estos sistemas garantizan la integridad, disponibilidad y confidencialidad de la información, facilitando su posterior análisis y reporte.

3.3. Generación y Envío de Reportes

Mensualmente, las empresas están obligadas a generar reportes que reflejen los movimientos y existencias de hidrocarburos y petrolíferos. Estos informes deben ser enviados al SAT en los formatos y plazos establecidos, permitiendo a la autoridad fiscal supervisar el cumplimiento de las obligaciones tributarias y detectar posibles irregularidades.

GOB.MX

4. Importancia del Control Volumétrico

La implementación del control volumétrico en México responde a múltiples objetivos que benefician tanto al Estado como a las empresas del sector:

Transparencia Fiscal: Facilita la correcta determinación y pago de impuestos, reduciendo la evasión fiscal y asegurando que las contribuciones se realicen conforme a la ley.

Seguridad Energética: Permite un monitoreo preciso de los flujos de hidrocarburos, contribuyendo a la detección oportuna de pérdidas, robos o prácticas ilícitas que puedan comprometer el suministro energético del país.

Eficiencia Operativa: Proporciona a las empresas herramientas para optimizar sus procesos internos, mejorar la gestión de inventarios y tomar decisiones informadas basadas en datos precisos y actualizados.

5. Desafíos en la Implementación

A pesar de los beneficios, la adopción de sistemas de control volumétrico enfrenta ciertos desafíos:

Inversión en Infraestructura: La adquisición e instalación de equipos de medición y sistemas informáticos representan una inversión significativa, especialmente para pequeñas y medianas empresas.

Capacitación del Personal: Es esencial que el personal encargado de operar y mantener estos sistemas cuente con la formación adecuada para garantizar su correcto funcionamiento y cumplimiento normativo.

Actualización Tecnológica: La evolución constante de la tecnología y las normativas exige a las empresas mantenerse actualizadas y adaptar sus sistemas conforme a los nuevos requerimientos.

6. Conclusión

El control volumétrico se erige como una herramienta fundamental en la gestión eficiente y transparente de los hidrocarburos y petrolíferos en México. Su correcta implementación no solo asegura el cumplimiento de las obligaciones fiscales, sino que también fortalece la seguridad energética y promueve la competitividad en el sector. Para lograr estos objetivos, es indispensable que las empresas inviertan en

tecnología adecuada, capaciten a su personal y se mantengan informadas sobre las actualizaciones normativas pertinentes.

Instalación de Visual Studio

Visual Studio 2022. En esta versión, es fácil elegir e instalar solo las características que necesita.

Nota

Si necesita instalar Visual Studio mediante programación, consulte [Uso de parámetros de línea de comandos](#).

¿Quieres saber más sobre lo que hay de más novedades en esta versión? Consulte las [notas de la versión](#).

Puede descargar e instalar una versión específica desde la página [Historial de versiones de Visual Studio 2022](#).

¿Listo para instalar? Este artículo le guía por él, paso a paso.

Paso 1: Asegúrese de que el equipo está listo para Visual Studio

Antes de empezar a instalar Visual Studio:

- Compruebe los [requisitos del sistema](#). Estos requisitos le ayudan a saber si el equipo admite Visual Studio 2022.
- Asegúrese de que el usuario que realiza la instalación inicial tiene permisos de administrador en el equipo. Para obtener más información, consulte [Permisos de usuario y Visual Studio](#).
- Aplique las actualizaciones más recientes de Windows. Estas actualizaciones garantizan que el equipo tenga las actualizaciones de seguridad más recientes y los componentes del sistema necesarios para Visual Studio.
- Reanudar. Reiniciar garantiza que las instalaciones o actualizaciones pendientes no impidan la instalación de Visual Studio.
- Liberar espacio. Quite los archivos y aplicaciones innecesarios de la unidad del sistema; por ejemplo, ejecute la aplicación Limpieza de disco.

Puede instalar Visual Studio 2022 en paralelo con otras versiones. Para obtener más información, vea [Compatibilidad y destinatarios de la plataforma de Visual Studio 2022](#) e [Instalación de distintas versiones de Visual Studio en paralelo](#).

Paso 2: Determinación de la versión y edición de Visual Studio que se va a instalar

Decida qué versión y edición de Visual Studio se va a instalar. Las opciones más comunes son:

- La versión más reciente de Visual Studio 2022 hospedada en servidores de Microsoft. Para instalar esta versión, seleccione el botón siguiente y elija la edición que desee. El instalador descarga un pequeño *cargador de arranque* en su carpeta *Descargas*.

Descargar Visual Studio

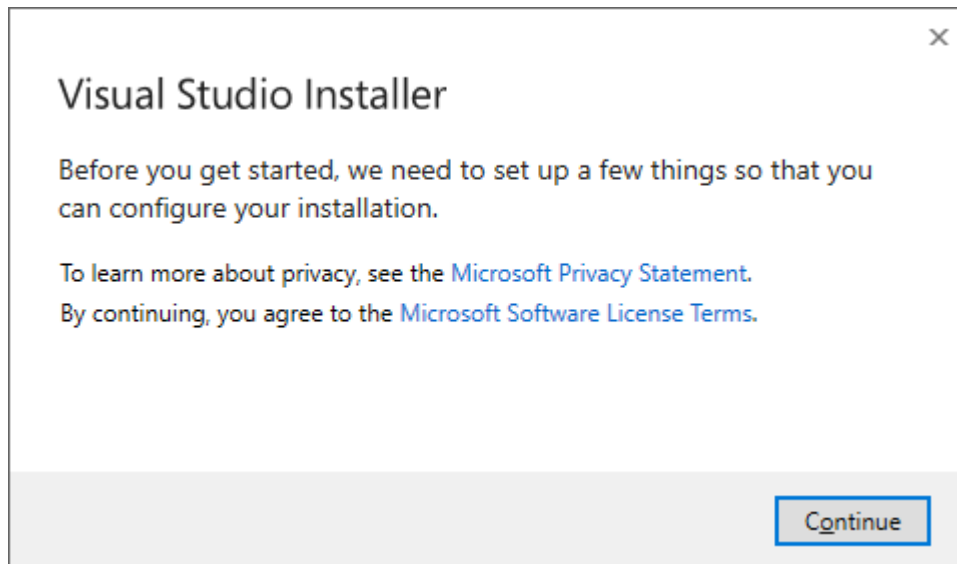
- Si ya tiene Instalado Visual Studio, puede [instalar otra versión simultáneamente](#). Para ello, debe elegir alguna de las que se ofrecen en la pestaña **Disponible** de Instalador de Visual Studio.
- Puede descargar un cargador de arranque para una versión específica en la página [Historial de versiones de Visual Studio 2022](#) y usarlo para instalar Visual Studio.
- El administrador de TI puede apuntar a una ubicación específica desde la que instalar Visual Studio.

Paso 3: Iniciar la instalación

Si descargó un archivo bootstrap, puede usarlo para instalar Visual Studio. Necesita permisos de administrador. El instalador inicial instala la versión más reciente del Instalador de Visual Studio. El instalador es un programa independiente que proporciona todo lo que necesita para instalar y personalizar Visual Studio.

1. En la carpeta 'Descargas' de , haga doble clic en el programa de arranque denominado *VisualStudioSetup.exe* o con un nombre similar a *vs_community.exe* para iniciar la instalación.

2. Si recibe un aviso de Control de cuentas de usuario, elija **Sí**. El cuadro de diálogo le pide que confirme los términos de licencia de Microsoft y la Declaración de privacidad de Microsoft . Elija **Continuar**.

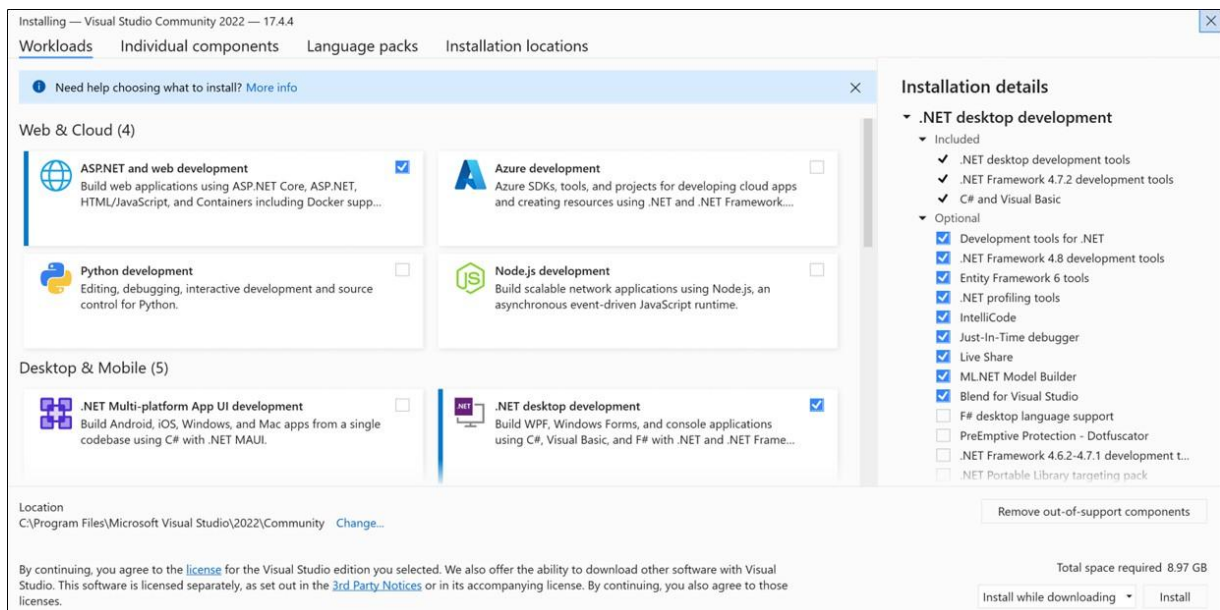


Se abre el Instalador de Visual Studio. También puede instalar cualquier producto que la pestaña [Instaladores disponibles de Visual Studio](#) ofrece.

Paso 4: Elegir cargas de trabajo

Después de instalar el Instalador de Visual Studio, puede usarlo para personalizar la instalación seleccionando los conjuntos de características o *cargas de trabajo*, que desee. Así es como.

1. Seleccione la carga de trabajo que desee en el instalador de Visual Studio .



Revise los resúmenes de carga de trabajo para decidir qué carga de trabajo admite las características que necesita. Por ejemplo, elija la carga de trabajo de **ASP.NET y desarrollo web** para editar páginas web de ASP.NET con Vista Previa en Vivo o crear aplicaciones web responsivas con Blazor. Puede elegir entre las cargas de trabajo de **escritorio, & móvil, o** para desarrollar aplicaciones multiplataforma con C#, o proyectos de C++ que tienen como destino C++20.

2. Después de elegir las cargas de trabajo que desee, seleccione **Instalar**.

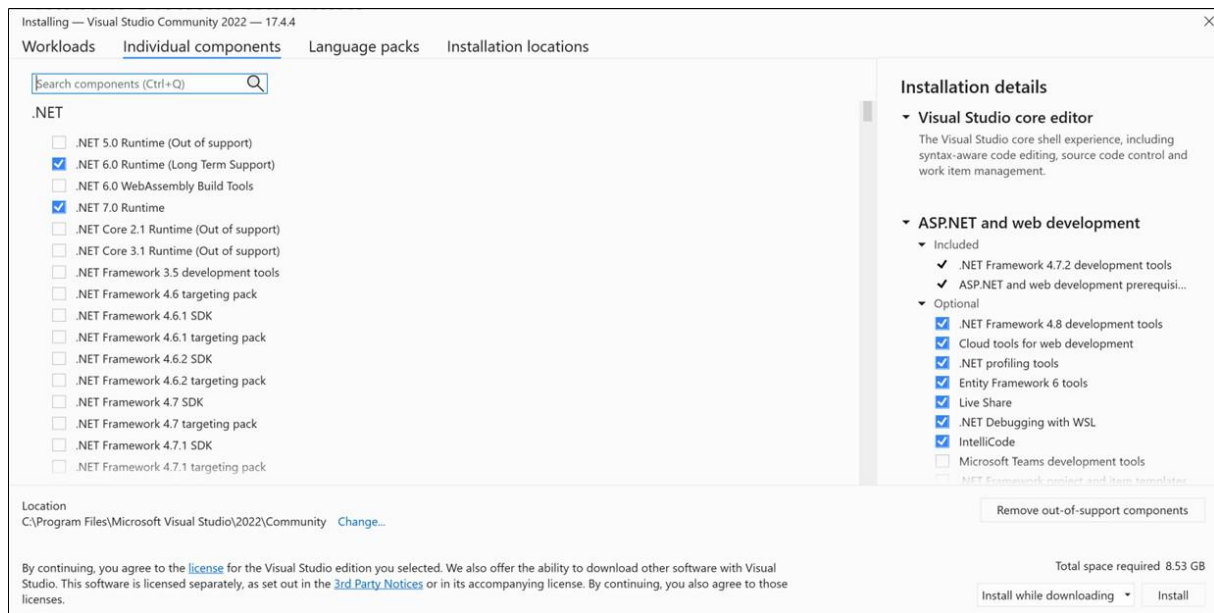
A continuación, aparecen pantallas de estado que muestran el progreso de la instalación de Visual Studio.

Sugerencia

En cualquier momento después de la instalación, puede instalar cargas de trabajo o componentes que no instaló inicialmente. Si tiene Visual Studio abierto, vaya a **Tools>Obtener herramientas y características**, que abre el Instalador de Visual Studio. O bien, abra el Instalador de Visual Studio desde el menú Inicio . Desde allí, puede elegir las cargas de trabajo o los componentes que desea instalar. A continuación, elija **Modificar**.

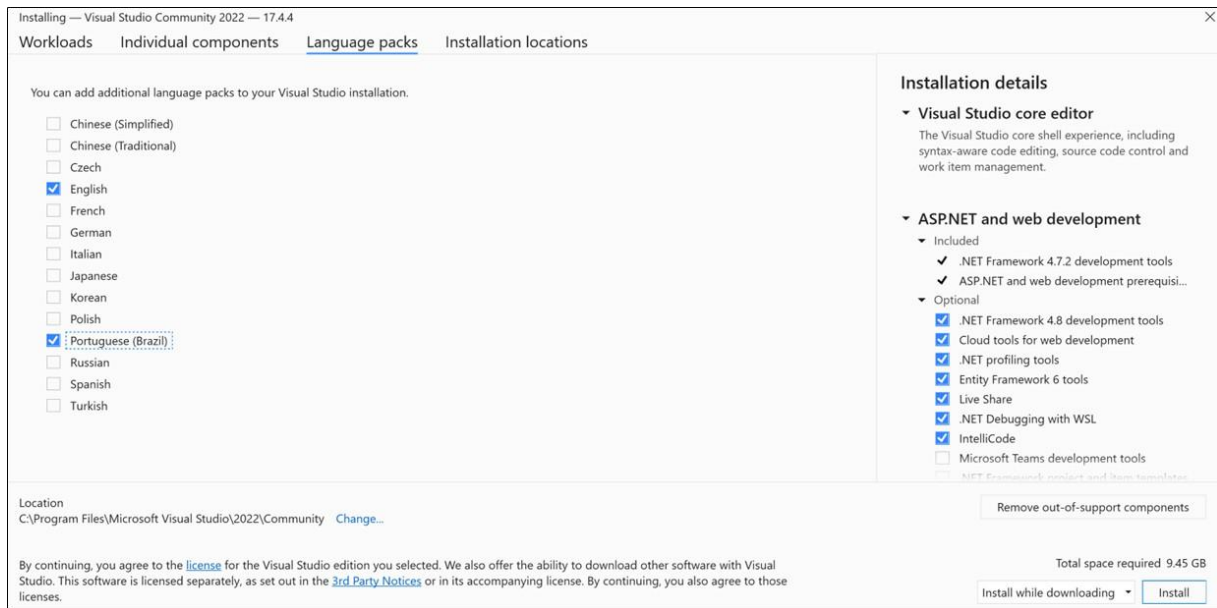
Paso 5: Elegir componentes individuales (opcional)

Si no desea usar la característica Cargas de trabajo para personalizar la instalación de Visual Studio o desea agregar más componentes que las instalaciones de una carga de trabajo, puede instalar o agregar componentes individuales desde la pestaña **Componentes individuales**. Elija lo que desee y, a continuación, siga las indicaciones.



Paso 6: Instalar paquetes de idioma (opcional)

De forma predeterminada, el programa de instalador intenta coincidir con el idioma del sistema operativo cuando se ejecuta por primera vez. Para instalar Visual Studio en un idioma de su elección, elija la pestaña Paquetes de idioma en el Instalador de Visual Studio y, a continuación, siga las indicaciones.



Cambiar el idioma del instalador desde la línea de comandos

Otra manera de cambiar el idioma predeterminado es ejecutar el instalador desde la línea de comandos. Por ejemplo, puede forzar que el instalador se ejecute en inglés mediante el siguiente comando:

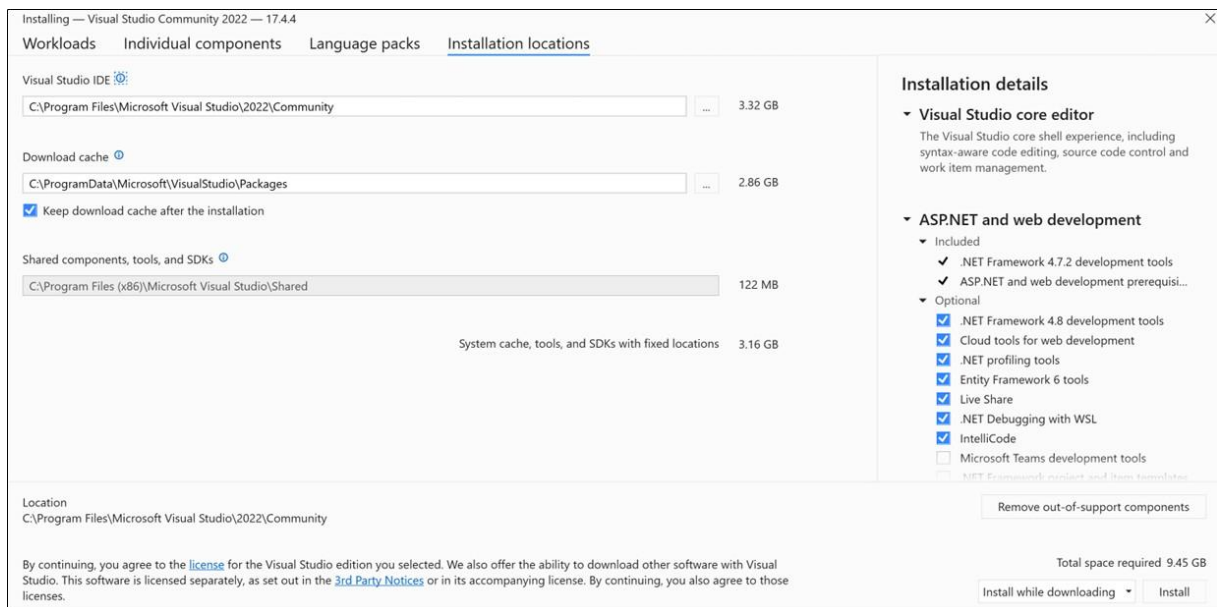
```
shellCopiar
```

```
vs_installer.exe --locale en-US
```

El instalador recuerda esta configuración al volver a ejecutarla. El instalador admite estas configuraciones regionales del idioma : zh-cn, zh-tw, cs-cz, en-us, es-es, fr-fr, de-de, it-it, ja-jp, ko-kr, pl-pl, pt-br, ru-ru y tr-tr.

Paso 7: Seleccionar la ubicación de instalación (opcional)

Puede reducir la superficie de memoria de instalación de Visual Studio en la unidad del sistema. Para obtener más información, consulte [Selección de las ubicaciones de instalación](#).



Importante

Solo puede seleccionar una unidad diferente para el **IDE de Visual Studio** o la **caché de descarga** al instalar Visual Studio por primera vez. Si ya la instaló y quiere cambiar las unidades, debe desinstalar Visual Studio y volver a instalarla.

Si ya ha instalado previamente Visual Studio en el equipo, no podrá cambiar la ruta de acceso de los **componentes, las herramientas y los SDK compartidos**. Aparece atenuado. Todas las instalaciones de Visual Studio comparten esta ubicación.

Paso 8: Iniciar sesión en su cuenta (opcional)

Aunque no tiene que iniciar sesión, hacerlo ofrece muchas [ventajas](#).

Puede evaluar una evaluación gratuita de [Visual Studio Professional o Visual Studio Enterprise](#) durante 30 días. Si inicia sesión, puede ampliar el período de prueba a 90 días. La extensión de prueba de 90 días solo funciona una vez. Para seguir usando Visual Studio después de que finalice un período de prueba, desbloquee con una suscripción en línea de o una clave de producto de .

Visual Studio Community no requiere que inicie sesión. Sin embargo, si la instalación le pide que inicie sesión periódicamente, [inicie sesión](#) para seguir usando Visual Studio Community sin interrupciones).

Paso 9: Empezar a desarrollar

Una vez completada la instalación, puede empezar a desarrollar con Visual Studio.

1. Seleccione el botón **Iniciar**.
2. En la ventana de inicio, elija **Crear un nuevo proyecto**.
3. En el cuadro de búsqueda de plantillas, escriba el tipo de aplicación que desea crear para ver una lista de plantillas disponibles. La lista de plantillas depende de las cargas de trabajo que haya elegido durante la instalación. Para ver diferentes plantillas, elija diferentes cargas de trabajo.

También puede filtrar la búsqueda de un lenguaje de programación específico mediante la lista desplegable **Language**. Además, puede filtrar mediante la lista **Plataforma** y la lista **Tipo de proyecto**.

4. Seleccione **Siguiente**. Proporcione otra información en los cuadros de diálogo siguientes y seleccione **Crear**.

Visual Studio abre el nuevo proyecto y está listo para codificar.

Inicio de la aplicación ABP

Normalmente, se usa el comando CLI de [ABP](#) `abp new` para [comenzar con una de las plantillas de solución de inicio](#) prediseñadas . Al hacerlo, generalmente no es necesario conocer los detalles de cómo se integra ABP con la aplicación ni cómo se configura e inicializa. La plantilla de inicio también incluye los paquetes básicos de ABP y [los módulos de la aplicación](#) preinstalados y configurados.

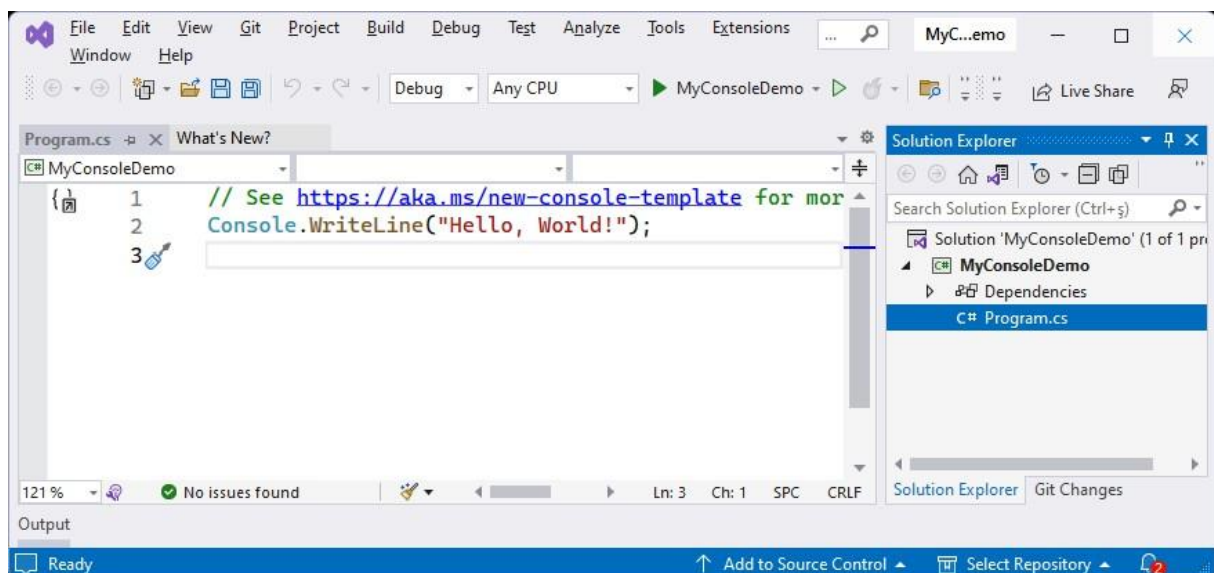
Siempre se recomienda [comenzar con una plantilla de inicio](#) y modificarla según sus necesidades. Lea este documento solo si desea comprender los detalles o si necesita modificar el inicio del ABP.

Si bien ABP cuenta con numerosas funciones e integraciones, está diseñado como un framework ligero y modular. Consta de [cientos de paquetes NuGet y NPM](#) , por lo que solo puede usar las funciones que necesita. Si sigue el documento "[Empezando con una aplicación ASP.NET Core MVC/Razor Pages vacía](#)" , verá lo fácil que es instalar ABP en un proyecto ASP.NET Core vacío desde cero. Solo necesita instalar un único paquete NuGet y realizar algunos cambios menores. Este documento es para quien quiera entender mejor cómo se inicializa y configura el ABP al iniciarse.

Instalación en una aplicación de consola

Una aplicación de consola .NET es la aplicación .NET minimalista. Por lo tanto, es recomendable mostrar la instalación de ABP en una aplicación de consola como ejemplo minimalista.

Si [crea una nueva aplicación de consola con Visual Studio](#) (para .NET 8.0 o posterior), verá la siguiente estructura de solución (llamé la solución como MyConsoleDemo):



Este ejemplo utiliza las [declaraciones de nivel superior](#) , por lo que consta de una sola línea de código.

El primer paso es instalar el paquete NuGet [Volo.Abp.Core](#) , el principal de ABP. Puede instalarlo mediante la CLI de ABP. Ejecute el siguiente comando en la carpeta del archivo .csproj donde desee instalar el paquete:

```
abp add-package Volo.Abp.Core
```

Si aún no lo ha hecho, primero debe instalar la [CLI de ABP](#) . Para ver otras opciones de instalación, consulte [la página de descripción del paquete](#) .

Alternativamente, puede utilizar una terminal de línea de comandos en la carpeta raíz del proyecto (la carpeta que contiene el MyConsoleDemo.csprojarchivo, para este ejemplo):

```
dotnet add package Volo.Abp.Core
```

Intento Copiar

Después de agregar el paquete NuGet, debemos crear una [clase de módulo](#) raíz para nuestra aplicación. Podemos crear la siguiente clase en el proyecto:

```
using Volo.Abp.Modularity;

namespace MyConsoleDemo
{
    public class MyConsoleDemoModule : AbpModule
    {
    }
}
```

Esta es una clase vacía derivada de la AbpModuleclase anterior. Es la clase principal con la que controlará las dependencias de su aplicación e implementará la configuración y la lógica de inicio/apagado. Para más información, consulte el documento [sobre Modularidad](#) .

Como segundo y último paso, cambie Program.cscomo se muestra en el siguiente bloque de código:

```

using MyConsoleDemo;
using Volo.Abp;

// 1: Create the ABP application container
using var application = await AbpApplicationFactory.CreateAsync<MyConsoleDemoModule>();

// 2: Initialize/start the ABP (and all the modules)
await application.InitializeAsync();

Console.WriteLine("ABP has been started...");

// 3: Stop the ABP (and all the modules)
await application.ShutdownAsync();

```

Eso es todo. Ahora, ABP está instalado, integrado, iniciado y detenido en tu aplicación. A partir de ahora, puedes instalar [paquetes ABP](#) en tu aplicación cuando los necesites.

Instalación de un paquete de marco

Si desea enviar correos electrónicos desde su aplicación .NET, puede usar [la clase SmtpClient](#) estándar de .NET . ABP también ofrece un IEmailSenderservicio que simplifica [el envío de correos electrónicos](#) y la configuración de los ajustes de correo electrónico en una ubicación central. Para usarlo, debe instalar el paquete NuGet [Volo.Abp.Emailing](#) en su proyecto:

```
dotnet add package Volo.Abp.Emailing
```

Al agregar un nuevo paquete/módulo ABP, también debe especificar la dependencia del módulo desde su clase. Por lo tanto, modifique la MyConsoleDemoModuleclase como se muestra a continuación:

```

using Volo.Abp.Emailing;
using Volo.Abp.Modularity;

namespace MyConsoleDemo
{
    [DependsOn(typeof(AbpEmailingModule))] // Added the module dependency
    public class MyConsoleDemoModule : AbpModule
    {
    }
}

```

Acabo de añadir un [DependsOn] atributo para indicar que quiero usar el módulo de correo electrónico ABP (AbpEmailingModule). Ahora puedo usar el IEmailSenderservicio en mi Program.cs:

```

using Microsoft.Extensions.DependencyInjection;
using MyConsoleDemo;
using Volo.Abp;
using Volo.Abp.Emailing;

using var application = await AbpApplicationFactory.CreateAsync<MyConsoleDemoModule>();
await application.InitializeAsync();

// Sending emails using the IEmailSender service
var emailsender = application.ServiceProvider.GetRequiredService<IEmailSender>();
await emailsender.SendAsync(
    to: "info@acme.com",
    subject: "Hello World",
    body: "My message body..."
);

await application.ShutdownAsync();

```

Si ejecutas esa aplicación, recibirás un error de ejecución que indica que la configuración de envío de correo electrónico aún no se ha configurado. Puedes consultar el [documento "Envío de correo electrónico"](#) para saber cómo configurarlo.

Eso es todo. Instale un paquete NuGet de ABP, agregue la dependencia del módulo (usando el [DependsOn] atributo) y use cualquier servicio dentro del paquete NuGet.

La [CLI de ABP](#) ya tiene un comando especial para realizar la adición de un NuGet de ABP y también agregar el [DependsOn] atributo a su clase de módulo con un solo comando:

```
abp add-package Volo.Abp.Emailing
```

Le sugerimos que utilice el `abp add-package` comando en lugar de hacerlo manualmente.

Fábrica de aplicaciones Abp

`AbpApplicationFactory` es la clase principal que crea un contenedor de aplicaciones ABP. Proporciona un único método estático `CreateAsync(y, Create)` si no se puede usar programación asíncrona), con múltiples sobrecargas. Analicemos estas sobrecargas para comprender dónde se pueden usar.

La primera sobrecarga obtiene un parámetro de clase de módulo genérico como el que hemos utilizado antes en este documento:

```
AbpApplicationFactory.CreateAsync<MyConsoleDemoModule>();
```

El parámetro de clase genérico debe ser la clase del módulo raíz de la aplicación. Todos los demás módulos se resuelven como dependencias de ese módulo.

La segunda sobrecarga utiliza la clase del módulo como `Type` parámetro, en lugar del parámetro genérico. Por lo tanto, el bloque de código anterior podría reescribirse como se muestra a continuación:

```
AbpApplicationFactory.CreateAsync(typeof(MyConsoleDemoModule));
```

Ambas sobrecargas funcionan exactamente igual. Por lo tanto, puedes usar la segunda si desconoces el tipo de clase del módulo en tiempo de desarrollo y, de alguna manera, lo calculas en tiempo de ejecución.

Si utiliza uno de los métodos anteriores, ABP crea una colección de servicios internos (`IServiceCollection`) y un proveedor de servicios interno (`IServiceProvider`) para configurar el sistema [de inyección de dependencias](#) internamente. Observe que hemos usado la `application.ServiceProvider` propiedad en la sección "*Instalación de un paquete de Framework*" para resolver el `IEmailSenderservicio` desde el sistema de inyección de dependencias.

La siguiente sobrecarga obtiene un `IServiceCollection` parámetro suyo para permitirle configurar usted mismo el sistema de inyección de dependencia o integrarlo con otro marco (como ASP.NET Core) que también configura el sistema de inyección de dependencia internamente.

Podemos cambiar lo `Program.cs` que se muestra a continuación para administrar externamente la configuración de inyección de dependencia:

```
using Microsoft.Extensions.DependencyInjection;
using MyConsoleDemo;
using Volo.Abp;

// 1: Manually created the IServiceCollection
IServiceCollection services = new ServiceCollection();

// 2: Pass the IServiceCollection externally to the ABP
using var application = await AbpApplicationFactory
    .CreateAsync<MyConsoleDemoModule>(services);

// 3: Manually built the IServiceProvider object
IServiceProvider serviceProvider = services.BuildServiceProvider();

// 4: Pass the IServiceProvider externally to the ABP
await application.InitializeAsync(serviceProvider);

Console.WriteLine("ABP has been started...");

await application.ShutdownAsync();
```

En este ejemplo, usamos el contenedor de inyección de dependencias estándar de .NET. La `services.BuildServiceProvider()` llamada crea el contenedor estándar. Sin embargo, ABP ofrece un método de extensión

alternativo, `BuildServiceProviderFromFactory()` que funciona correctamente incluso si se usa otro contenedor de inyección de dependencias:

```
IServiceProvider serviceProvider = services.BuildServiceProviderFromFactory();
```

Puede consultar el documento [de integración de Autofac](#) si desea saber cómo integrar el contenedor de inyección de dependencia [de Autofac](#) con ABP.

Finalmente, el `CreateAsync` método tiene una última sobrecarga que toma el nombre de la clase del módulo como `Type` parámetro y como `IServiceCollection` objeto. Por lo tanto, podríamos reescribir el `CreateAsync` uso del último método como en el siguiente bloque de código:

```
using var application = await AbpApplicationFactory
    .CreateAsync(typeof(MyConsoleDemoModule), services);
```

Todas las `CreateAsync` sobrecargas de métodos tienen `Create` sus contrapartes. Si su tipo de aplicación no puede utilizar la programación asíncrona (es decir, no puede usar la `await` palabra clave), puede usar el `Create` método en su lugar `CreateAsync`.
Opciones de creación de aplicaciones Abp

Todas las `CreateAsync` sobrecargas pueden tener un `Action<AbpApplicationCreationOptions>` parámetro opcional para configurar las opciones utilizadas al crear la aplicación. Vea el siguiente ejemplo:

```
using var application = await AbpApplicationFactory
    .CreateAsync<MyConsoleDemoModule>(options =>
    {
        options.ApplicationName = "MyApp";
    });
```

Hemos pasado un método lambda para configurar la `ApplicationName` opción. Aquí hay una lista de todas las opciones estándar:

- `ApplicationName`: Un nombre legible para la aplicación. Es un valor único para la aplicación.

- Configuration: Se puede usar para configurar la [aplicación](#) cuando el sistema de alojamiento no la proporciona. No es necesaria para ASP.NET Core ni otras aplicaciones alojadas en .NET. Sin embargo, si ha utilizado AbpApplicationFactory un proveedor de servicios interno, puede usar esta opción para configurar cómo se crea la configuración de la aplicación.
- Environment: Nombre del entorno para la aplicación.
- PlugInSourcesLista de fuentes de complementos. Consulta la [documentación de módulos de complementos](#) para saber cómo trabajar con ellos.
- ServicesEl IServiceCollection objeto que se puede usar para registrar dependencias de servicio. Generalmente no es necesario, ya que los servicios se configuran en la [clase del módulo](#) . Sin embargo, se puede usar al escribir métodos de extensión para la AbpApplicationCreationOptions clase.

La opción NombreDeLaAplicación

Como se definió anteriormente, la ApplicationName opción es un nombre legible para la aplicación. Es un valor único para cada aplicación.

ApplicationNameEl ABP lo utiliza en varios lugares para distinguir la aplicación. Por ejemplo, el sistema [de registro de auditoría](#) guarda el [nombre ApplicationName de la aplicación] en cada registro de auditoría escrito por la aplicación relacionada, para que pueda saber qué aplicación creó la entrada. Por lo tanto, si su sistema consta de varias aplicaciones (como una solución de microservicios) que guardan los registros de auditoría en un único punto, debe asegurarse de que cada aplicación tenga un [nombre de la aplicación] diferente ApplicationName.

El ApplicationName valor de la propiedad se establece automáticamente desde el **nombre del conjunto de entrada** (generalmente, el nombre del proyecto en una solución .NET) de forma predeterminada, lo que es adecuado para la mayoría de

los casos, ya que cada aplicación normalmente tiene un nombre de ensamblado de entrada único.

Hay dos maneras de configurar el nombre de la aplicación con un valor diferente. En la primera, puede configurar la propiedad en [la configuración](#) `ApplicationName` de su aplicación. La forma más sencilla es agregar un campo a su archivo: `ApplicationNameappsettings.json`

```
{
  "ApplicationName": "Services.Ordering"
}
```

Como alternativa, puede configurarlo `AbpApplicationCreationOptions.ApplicationName` al crear la aplicación ABP. Puede encontrar la llamada `AddApplication` o `AddApplicationAsync` en su solución (normalmente en el `Program.cs` archivo) y configurar la `ApplicationName` opción como se muestra a continuación:

```
await builder.AddApplicationAsync<OrderingServiceHttpApiHostModule>(options =>
{
    options.ApplicationName = "Services.Ordering";
});
```

Accesor de información de la aplicación

Si necesita acceder `ApplicationName` más tarde en su solución, puede inyectar el `IApplicationInfoAccessor` servicio y obtener el valor de su `ApplicationName` propiedad.

`IApplicationInfoAccessor` También proporciona un `InstanceId` valor GUID aleatorio que se genera al iniciar la aplicación. Puede usar ese valor para distinguir las instancias de la aplicación entre sí.

Aplicación IAbp

AbpApplicationFactory devuelve un IAbpApplication objeto de su método CreateAsync(o Create). IAbpApplication es el contenedor principal de una aplicación ABP. También está registrado en el sistema [de inyección de dependencias](#), por lo que puede inyectarlo IAbpApplication en sus servicios para usar sus propiedades y métodos.

A continuación le presentamos una lista de IAbpApplication propiedades que quizás le interese conocer:

- StartupModuleType: Obtiene el módulo raíz de la aplicación que se utilizó al crear el contenedor de la aplicación (en el AbpApplicationFactory.CreateAsync método).
- Services Una lista de todos los registros de servicios (el IServiceCollection objeto). No se pueden añadir nuevos servicios a esta colección tras la inicialización de la aplicación (aunque sí se pueden añadir, no tendrán ningún efecto).
- ServiceProvider: Una referencia al proveedor de servicios raíz que utiliza la aplicación. No se puede usar antes de inicializar la aplicación. Si necesita resolver servicios no singleton desde ese IServiceProvider objeto, cree siempre un nuevo ámbito de servicio y deséchelo después de usarlo. De lo contrario, su aplicación experimentará fugas de memoria. Consulte la sección "*Liberación/Descarte de Servicios*" del documento [sobre inyección de dependencias](#) para obtener más información sobre los ámbitos de servicio.
- Modules Lista de solo lectura de todos los módulos cargados en la aplicación actual. Como alternativa, puede inyectar el IModuleContainer servicio si necesita acceder a la lista de módulos en el código de su aplicación.

La IAbpApplication interfaz extiende la IApplicationInfoAccessor interfaz, por lo que puede obtener los valores ApplicationName e InstanceId de ella. Sin embargo, si solo

necesita acceder a estas propiedades, inyecte y use el `IAApplicationInfoAccessorservicio`.

`IAbpApplicationEs` desechable. Deséchelo siempre antes de salir de la aplicación.

Entorno de host de `IAbp`

A veces, al crear una aplicación, necesitamos obtener el entorno de alojamiento actual y tomar medidas en consecuencia. En estos casos, podemos usar servicios como [IWebHostEnvironment](#) o [IWebAssemblyHostEnvironment](#), proporcionados por .NET, en la aplicación final.

Sin embargo, no podemos usar estos servicios en una biblioteca de clases, que es utilizada por la aplicación final. ABP proporciona el `IAbpHostEnvironmentservicio`, que permite obtener el nombre del entorno actual cuando se desee. `IAbpHostEnvironmentABP` lo utiliza en varios lugares para realizar acciones específicas del entorno. Por ejemplo, ABP reduce la duración de la caché en...**Desarrollo**entorno para algunos servicios.

`IAbpHostEnvironment`obtiene el nombre del entorno actual según el siguiente orden:

1. Obtiene y establece el nombre del entorno si se especifica en el `AbpApplicationCreationOptions`.
2. Intenta obtener el nombre del entorno de los `IWebHostEnvironmentservicios` `IWebAssemblyHostEnvironment`para aplicaciones ASP.NET Core y Blazor WASM si el nombre del entorno no está especificado en el `AbpApplicationCreationOptions`.
3. Establece el nombre del entorno como **Producción**, si el nombre del entorno no se especifica o no se puede obtener de los servicios.

Puede configurar la `AbpApplicationCreationOptions` [clase de opciones](#) al crear la aplicación ABP y asignarle un nombre de entorno a su `Environment`propiedad. Puede encontrar la llamada `AddApplication` `` AddApplicationAsync` en su solución

(normalmente en el Program.cs archivo) y configurar la Environment opción como se muestra a continuación:

```
await builder.AddApplicationAsync<OrderingServiceHttpApiHostModule>(options =>
{
    options.Environment = Environments.Staging; //or directly set as "Staging"
});
```

Luego, siempre que necesite obtener el nombre del entorno actual o verificar el entorno, puede usar la IAbpHostEnvironment interfaz:

```
public class MyDemoService
{
    private readonly IAbpHostEnvironment _abpHostEnvironment;

    public MyDemoService(IAbpHostEnvironment abpHostEnvironment)
    {
        _abpHostEnvironment = abpHostEnvironment;
    }

    public void MyMethod()
    {
        var environmentName = _abpHostEnvironment.EnvironmentName;

        if (_abpHostEnvironment.IsDevelopment()) { /* ... */ }

        if (_abpHostEnvironment.IsStaging()) { /* ... */ }

        if (_abpHostEnvironment.IsProduction()) { /* ... */ }

        if (_abpHostEnvironment.IsEnvironment("custom-environment")) { /* ... */ }
    }
}
```

Integraciones de host genérico .NET y ASP.NET Core

AbpApplicationFactorySe puede crear un contenedor de aplicaciones ABP independiente sin dependencias externas. Sin embargo, en la mayoría de los casos, se recomienda integrarlo con [el host genérico de .NET](#) o ASP.NET Core. Para estos usos, ABP proporciona métodos de extensión integrados para crear fácilmente un contenedor de aplicaciones ABP bien integrado con estos sistemas.

El documento [Primeros pasos con una aplicación ASP.NET Core MVC / Razor Pages vacía](#) explica claramente cómo crear un contenedor de aplicaciones ABP en una aplicación ASP.NET Core.

También puede [crear una aplicación de consola](#) para ver cómo se integra con .NET Generic Host.

En la mayoría de los casos, creará aplicaciones ABP directamente mediante el newcomando CLI de ABP. Por lo tanto, no necesita preocuparse por estos detalles de integración.

Instalación del gestor de base de datos Dbeaver en Windows

1. Descargar Dbeaver

[Link de descarga Dbeaver](#)

En caso de que el link anterior no funcione, copie y pegue el siguiente link en su navegador, **https://dbeaver.io/files/dbeaver-ce-latest-x86_64-setup.exe**

2. Ejecutamos el archivo descargado **dbeaver-ce-23.2.2-x86_64-setup.exe** y comenzamos la instalación. Seguimos las imágenes que se muestran a continuación.





Bienvenido al Asistente de Instalación de DBeaver Community

Este programa instalará DBeaver Community en su ordenador.

Se recomienda que cierre todas las demás aplicaciones antes de iniciar la instalación. Esto hará posible actualizar archivos relacionados con el sistema sin tener que reiniciar su ordenador.

Presione Siguiente para continuar.

Siguiente >

Cancelar

Acuerdo de licencia

Por favor revise los términos de la licencia antes de instalar DBeaver Community.



Presione Avanzar Página para ver el resto del acuerdo.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

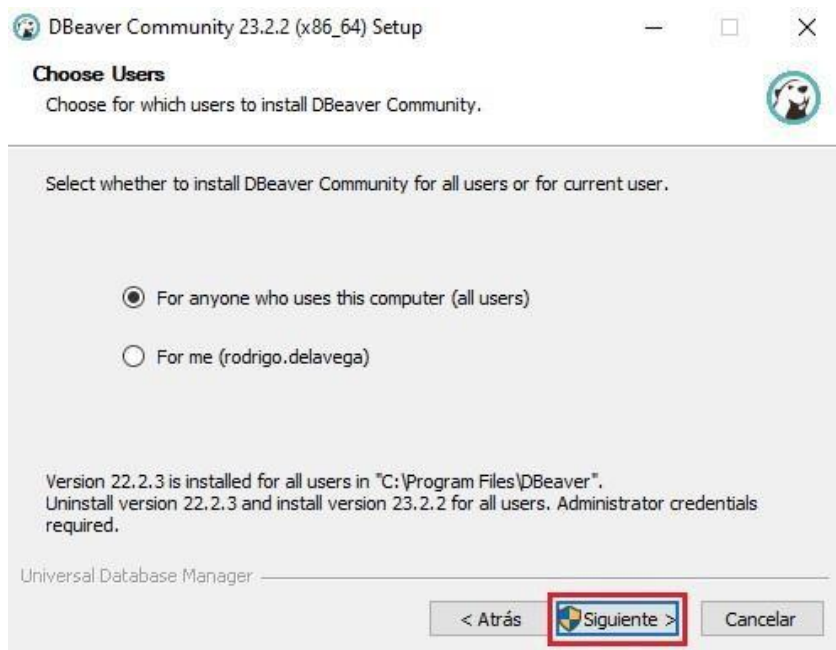
Si acepta todos los términos del acuerdo, seleccione Acepto para continuar. Debe aceptar el acuerdo para instalar DBeaver Community.

Universal Database Manager

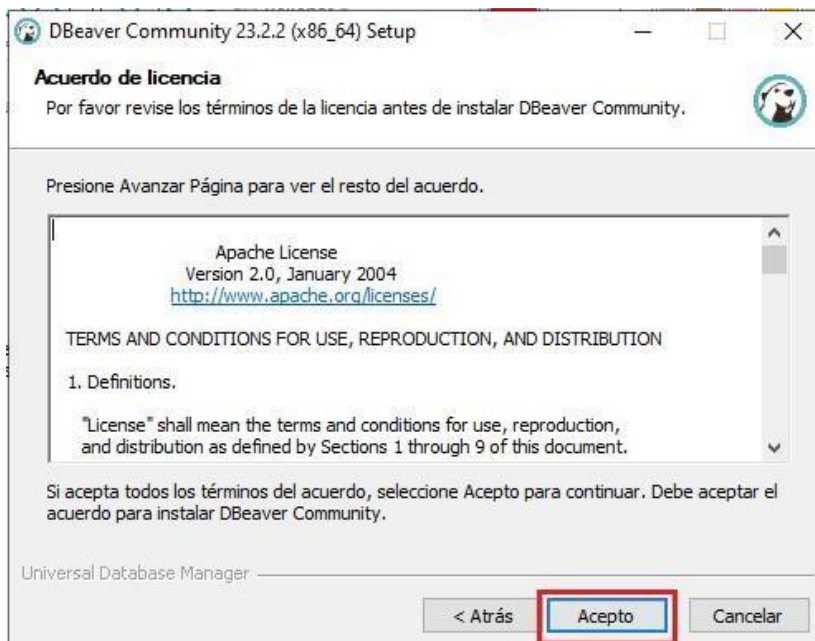
< Atrás

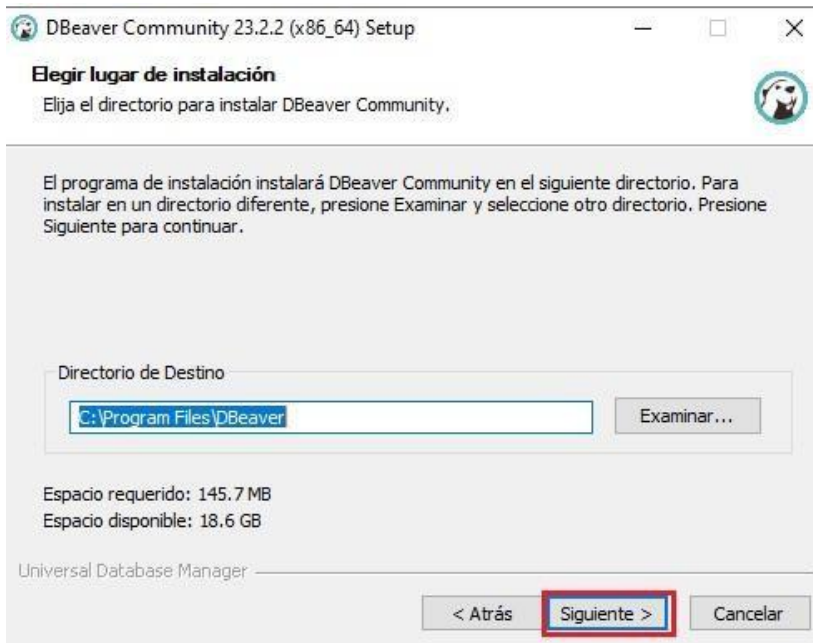
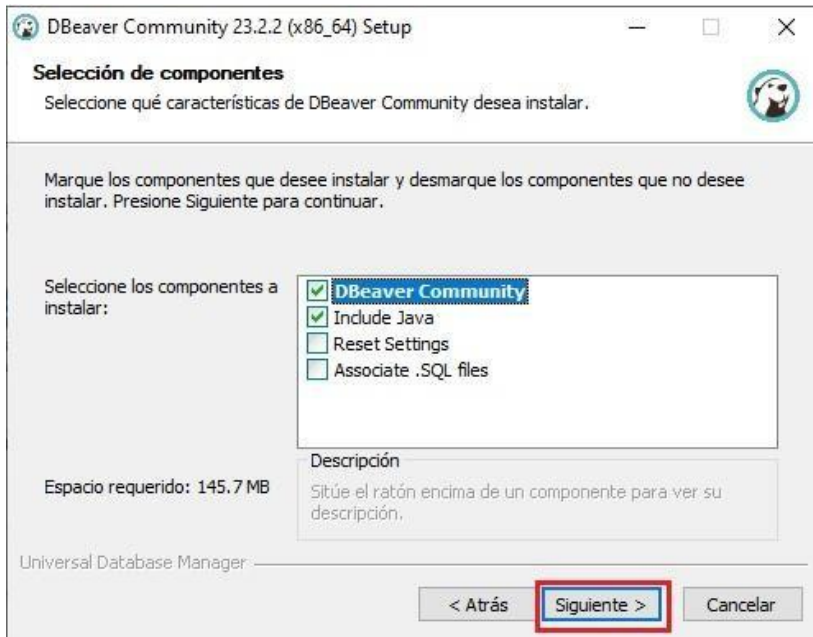
Acepto

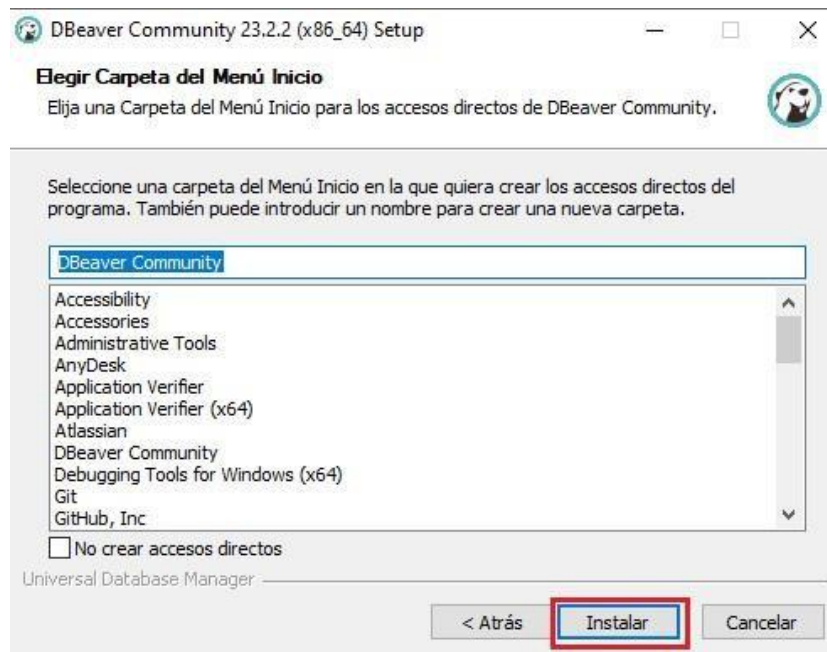
Cancelar



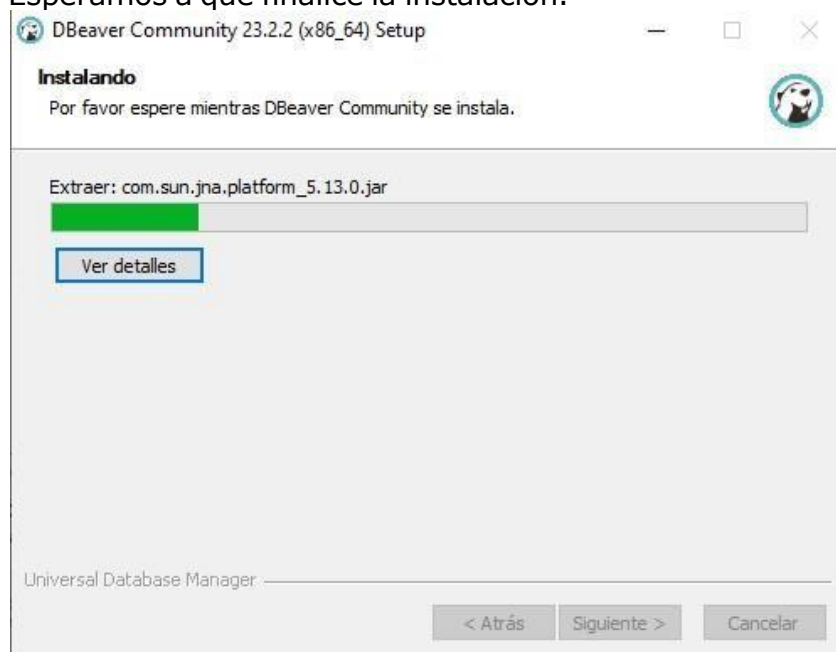
Si luego del cuadro anterior les aparece un mensaje diciendo algo como "Quieres permitir que esta aplicación haga cambios en el equipo", le dan click a la opción "Si" y continúan la instalación. Luego debe aparecer lo siguiente,







Esperamos a que finalice la instalación.

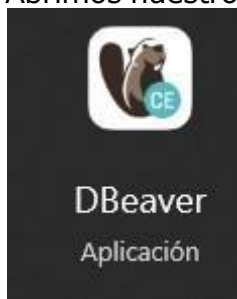


Por último, terminamos la instalación.

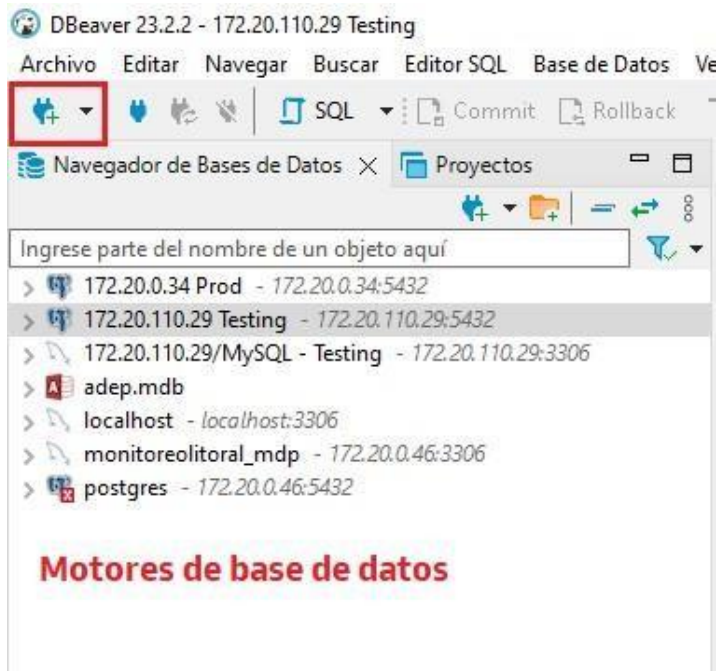


Ahora que ya instalamos el manejador de base de datos, vamos a configurar la conexión a nuestra base de datos PostgreSQL. Antes de continuar usted debe de tener el motor de base de datos instalado, en este caso PostgreSQL (ver instructivo de instalación).

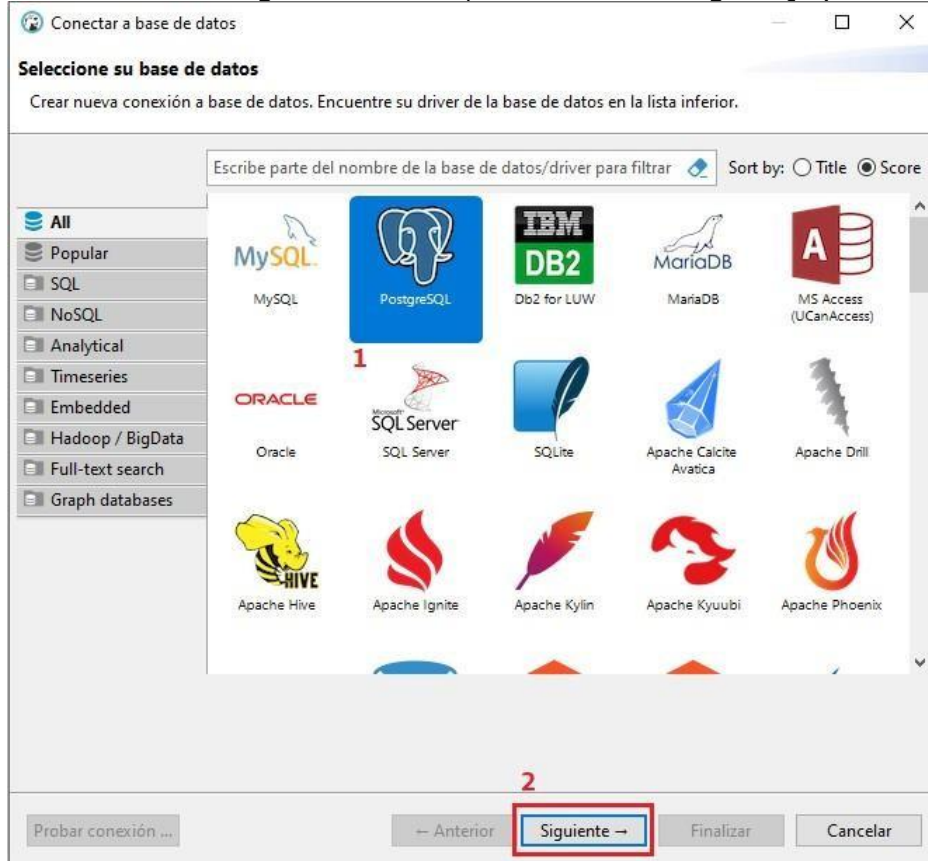
Abrimos nuestro manejador llamado DBeaver



Una vez abierto, veremos varias opciones. El primer paso es configurar la conexión con la base de datos postgresQL. Para ello dan click en el botón que se muestra en la imagen con el nro 1.



Nos mostrará el siguiente cuadro, seleccionar PostgreSQL y dar "Siguiete".

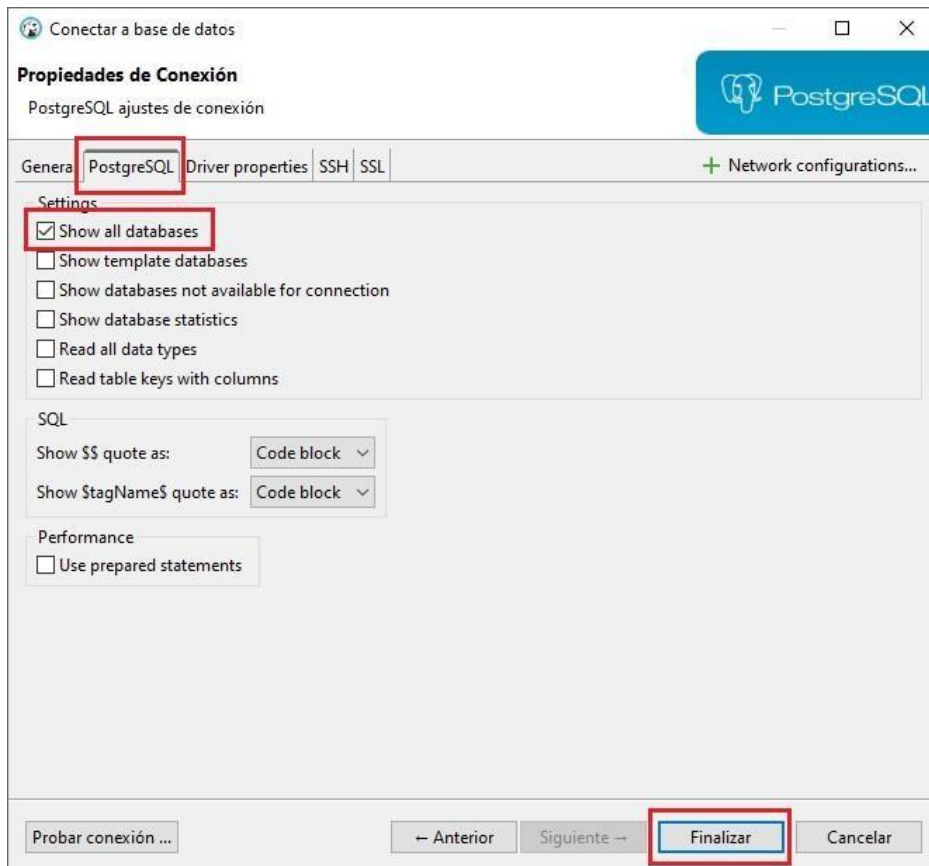


Luego deben configurar las credenciales para conectarse al motor de base de datos. En la pestaña "General" deben configurar el paso 1 y 2 tal cual como está en la imagen. Después en el punto 3 y 4 deben ingresar el usuario y contraseña que haya configurado cuando instalo el motor de base de datos. Por último, pruebe la conexión para saber si todo funciona correctamente. En caso que le pida descargar algunos drivers, hágalo.

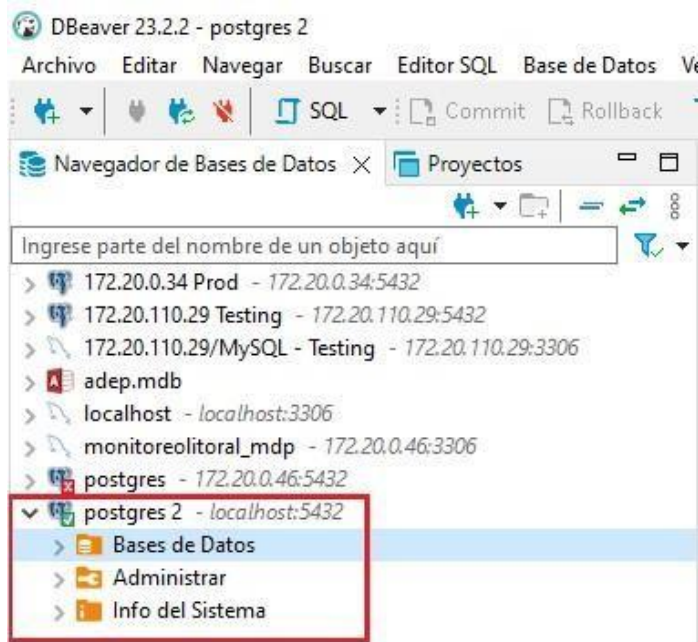
The screenshot shows the 'Conectar a base de datos' (Connect to database) dialog box with the 'General' tab selected. The 'PostgreSQL ajustes de conexión' (PostgreSQL connection settings) are displayed. The 'Server' section has 'Connect by:' set to 'Host'. The 'URL' field contains 'jdbc:postgresql://localhost:5432/postgres'. The 'Host' field is 'localhost' (marked with a red '1') and the 'Port' is '5432'. The 'Database' field is 'postgres' (marked with a red '2'). The 'Authentication' section has 'Authentication:' set to 'Database Native'. The 'Nombre de usuario:' (Username) is 'postgres' (marked with a red '3') and the 'Contraseña:' (Password) is masked with dots (marked with a red '4'). The 'Save password' checkbox is checked. The 'Advanced' section has 'Session role:' empty and 'Local Client:' set to 'PostgreSQL 15'. At the bottom, the 'Probar conexión ...' (Test connection ...) button is highlighted with a red box and a red '5'. Other buttons include 'Anterior' (Previous), 'Siguiente ->' (Next), 'Finalizar' (Finish), and 'Cancelar' (Cancel).

A modo de ejemplo se muestra una prueba de conexión exitosa. Ahora diríjase a la pestaña PostgreSQL y marque la opción "Show all databases". Por último de click en el botón Finalizar.

The screenshot shows the 'Connection test' dialog box with a successful connection message: 'Conectado (56 ms)' (Connected (56 ms)). The 'Server:' information is 'PostgreSQL 15.3' and 'PostgreSQL 15.3, compiled by Visual C++ build 1914, 64-bit'. The 'Driver:' information is 'PostgreSQL JDBC Driver 42.5.2'. At the bottom, there are two buttons: 'Aceptar' (Accept) and 'Detalles ↓' (Details).



Si todo fue configurado correctamente deben ver en el árbol donde se muestran las bases de datos la conexión recién hecha (en este ejemplo se llama "postgres 2"). Le damos doble click para abrirla y nos mostrará lo siguiente,



Un icono de check verde simboliza que está conectado al motor de base de datos. Luego dentro de "Bases de Datos" podremos ver y operar con las distintas bases de datos que aloja el motor al que nos conectamos.

Desarrollo

El proyecto consistió en crear una plataforma web para llevar un registro detallado y en tiempo real de las cantidades de hidrocarburos y petrolíferos que se producen, almacenan, transportan y comercializan en México. Para ello, la plataforma se desarrolló utilizando el framework ABP de una sola capa, debido a su estructura que permite crear sistemas modulares lo que garantiza una arquitectura robusta y escalable.

Para lograr el objetivo principal, se siguió un proceso de desarrollo iterativo que se divide en varias etapas. En la primera etapa, se llevó a cabo un análisis exhaustivo de los requerimientos del proyecto, lo que incluía la identificación de los actores involucrados en el proceso y la definición de las funcionalidades que se debían implementar en la plataforma.

Para ello, el proyecto debía contar con tres principales módulos; cada módulo sería dedicado al almacenamiento y creación de los siguientes elementos; Reportes-Dictámenes, Contribuyentes – Proveedores y Productos, cada módulo debía ser capaz de tener las funcionalidades de crear, modificar y eliminar registros, además que mostraría en una tabla con todos los registros guardados en la Base de Datos.

Para representar de una manera gráfica los requisitos del sistema se estructuro un Diagrama de Clases UML en el cual se representaron las clases, interfaces y sus relaciones en el sistema. Esto me ayudo a visualizar la estructura del sistema al proporcionarme una descripción completa y detallada de los objetos y sus relaciones.

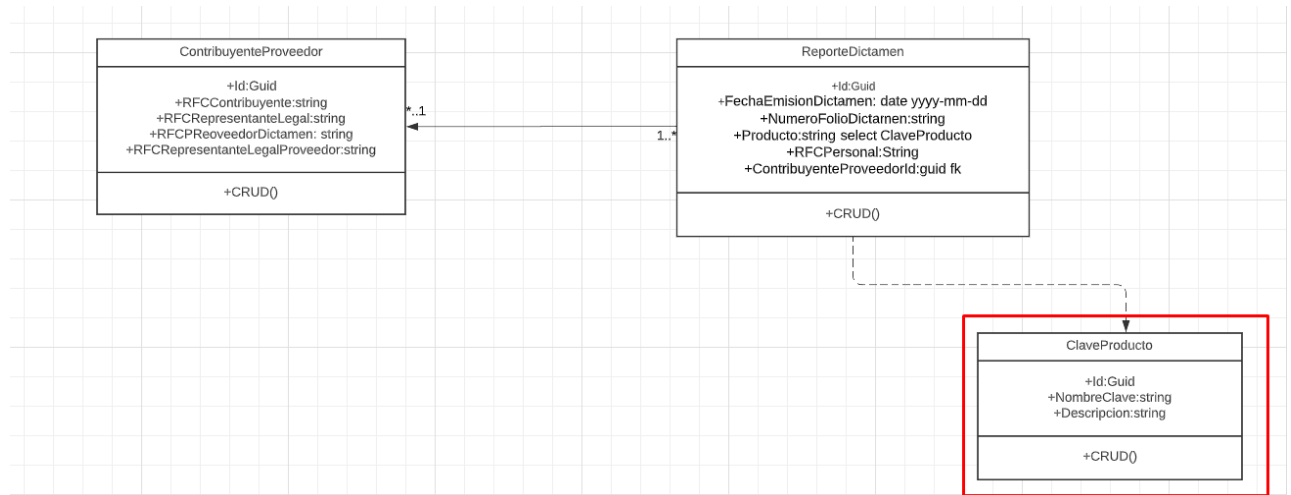


Imagen 1.- Diagrama de Clases UML "Sistema Web para la Generación de Dictámenes a partir de Análisis Volumétricos de Datos".

En la segunda etapa, se procedió a la configuración y adecuación de las entidades y módulos necesarios para la implementación de la plataforma. En esta etapa se definieron las entidades de la base de datos y se crearán los módulos necesarios para cada caso de uso.

Fue así que se estructuraron las entidades siguientes: ClaveProducto, Contribuyente, DictamenPrueba y una entidad extraordinaria llamada Dictamen XML que corresponde a una extensión de uso para Dictamen.

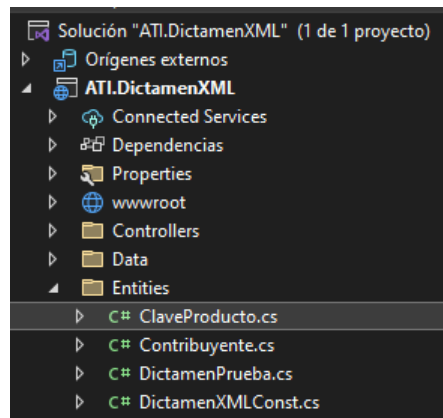


Imagen 2.- Arquitectura proyecto: Módulo de Entidades.

Los atributos de la entidad Producto son; "NombreClave" y "Descripción" en donde, se describirá el nombre con el cual identificaremos el producto y describirá al mismo para una mayor claridad.

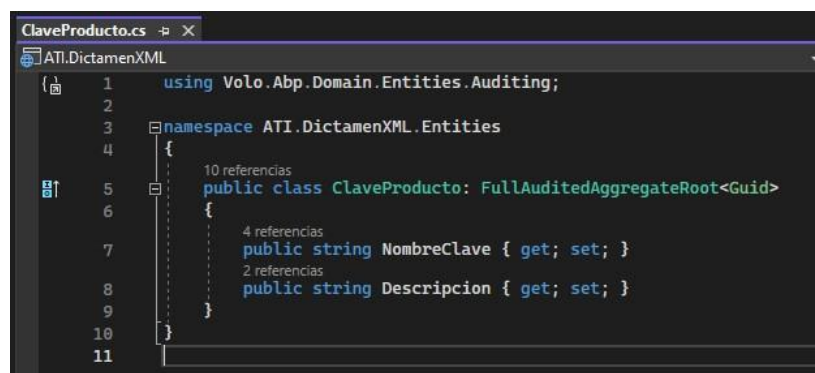


Imagen 3.- Entidad: Clave-Producto.

La entidad Contribuyentes tiene como atributos: "RFCContribuyente", "RFCRepresentante", "RFCProveedorDictamen" y "RFCRepresentanteLegalProveedor", cada atributo corresponde al RFC que identifica al Representante, Proveedor-Dictamen y Representante-Legal-Proveedor requisitos que son necesarios para la creación de un nuevo Contribuyente.

```

Contribuyente.cs
ClaveProducto.cs
ATI.DictamenXML
1 using Volo.Abp.Domain.Entities.Auditing;
2 namespace ATI.DictamenXML.Entities
3 {
4     {
5         13 referencias
6         public class Contribuyente : AuditedAggregateRoot<Guid>
7         {
8             4 referencias
9             public string RfcContribuyente { get; set; }
10            3 referencias
11            public string RfcRepresentanteLegal { get; set; }
12            3 referencias
13            public string RfcProveedorDictamen { get; set; }
14            3 referencias
15            public string RfcRepresentanteLegalProveedor { get; set; }
16        }
17    }
18 }

```

Imagen 4.- Entidad: Contribuyente.

En la entidad Dictamen cuenta con los atributos: "FechaEmisionDictamen", "NumeroFolioDictamen", "Producto", "RFCPersonal", "ContribuyenteProveedorId", "Contenido", "Estatus", "NumeroEmision". Este apartado describe con mayor detalle todas las características necesarias que implican la creación de un Dictamen nuevo, donde algunos atributos corresponden a llaves foraneas de las otras entidades.

```

DictamenPrueba.cs
Contribuyente.cs
ClaveProducto.cs
ATI.DictamenXML
1 using Volo.Abp.Domain.Entities.Auditing;
2 namespace ATI.DictamenXML.Entities
3 {
4     {
5         8 referencias
6         public class DictamenPrueba : FullAuditedAggregateRoot<Guid>
7         {
8             2 referencias
9             public DateTime FechaEmisionDictamen { get; set; }
10            2 referencias
11            public string NumeroFolioDictamen { get; set; }
12            2 referencias
13            public string Producto { get; set; }
14            2 referencias
15            public string RFCPersonal { get; set; }
16            4 referencias
17            public Guid ContribuyenteProveedorId { get; set; }
18            2 referencias
19            public string Contenido { get; set; }
20            2 referencias
21            public string Estatus { get; set; }
22            2 referencias
23            public string NumeroEmision { get; set; }
24        }
25    }
26 }

```

Imagen 5.- Endidad: Dictamen.

En la tercera etapa, se procedió a la implementación de la base de datos donde se almacenarán los registros generados por el usuario. Esta etapa debe garantizar la eficiencia y la seguridad en la gestión de los datos.

dbo.Dictamen...ucto [Diseño]			
Actualizar Archivo de script: dbo.DictamenClaveProducto.sql			
Nombre	Tipo de datos	Permitir valores NULL	
Id	uniqueidentifier	☐	
NombreClave	nvarchar(MAX)	☒	
Descripcion	nvarchar(MAX)	☒	
ExtraProperties	nvarchar(MAX)	☒	
ConcurrencyStamp	nvarchar(40)	☒	
CreationTime	datetime2(7)	☐	
CreatorId	uniqueidentifier	☒	
LastModificationTime	datetime2(7)	☒	
LastModifierId	uniqueidentifier	☒	
IsDeleted	bit	☐	
DeleterId	uniqueidentifier	☒	
DeletionTime	datetime2(7)	☒	
		☐	

Imagen 6.- DB: Clave -Producto.

dbo.Dictamen...edor [Diseño]			
Actualizar Archivo de script: dbo.DictamenContribuyenteProve			
Nombre	Tipo de datos	Permitir valores NULL	
Id	uniqueidentifier	☐	
RfcContribuyente	nvarchar(MAX)	☒	
RfcRepresentanteLegal	nvarchar(MAX)	☒	
RfcProveedorDictamen	nvarchar(MAX)	☒	
RfcRepresentanteLegalProv	nvarchar(MAX)	☒	
ExtraProperties	nvarchar(MAX)	☒	
ConcurrencyStamp	nvarchar(40)	☒	
CreationTime	datetime2(7)	☐	
CreatorId	uniqueidentifier	☒	
LastModificationTime	datetime2(7)	☒	
LastModifierId	uniqueidentifier	☒	
		☐	

Imagen 7.- DB: Contribuyente – Proveedor.

dbo.Dictamen...ueba [Diseño]			
Actualizar Archivo de script: dbo.DictamenDictamenPrueba.sql			
Nombre	Tipo de datos	Permitir valores NULL	
Id	uniqueidentifier	☐	
FechaEmisionDictamen	datetime2(7)	☐	
NumeroFolioDictamen	nvarchar(MAX)	☒	
Producto	nvarchar(MAX)	☒	
RFCPersonal	nvarchar(MAX)	☒	
ContribuyenteProveedorId	uniqueidentifier	☐	
ExtraProperties	nvarchar(MAX)	☒	
ConcurrencyStamp	nvarchar(40)	☒	
CreationTime	datetime2(7)	☐	
CreatorId	uniqueidentifier	☒	
LastModificationTime	datetime2(7)	☒	
LastModifierId	uniqueidentifier	☒	
IsDeleted	bit	☐	
DeleterId	uniqueidentifier	☒	
DeletionTime	datetime2(7)	☒	
Contenido	nvarchar(MAX)	☒	
Estatus	nvarchar(MAX)	☒	
NumeroEmision	nvarchar(MAX)	☒	
		☐	

Imagen 8.- DB: Dictamen.

Se estableció una ID única para identificar cada uno de los registros, por ello, se implementaron atributos Guid los cuales generán una cadena de 25 caracteres aleatorios y no duplicables, también se busco la relación de las entidades para implementar las llaves foráneas en las mismas.

En la cuarta etapa, se desarrollo la interfaz de usuario, tanto en el back-end como en el front-end, para que los usuarios puedan interactuar con la plataforma de manera intuitiva y fácil de usar. El framework de ABP proporciona un Servicio de Aplicaciones (Application Layer) CRUD que permite Crear, Actualizar, Eliminar y Obtener los datos, lo que hace posible heredar la arquitectura a través de CrudAppService a cada entidad.

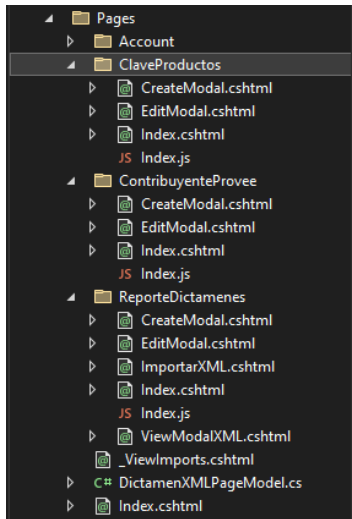


Imagen 9.- Vistas páginas.

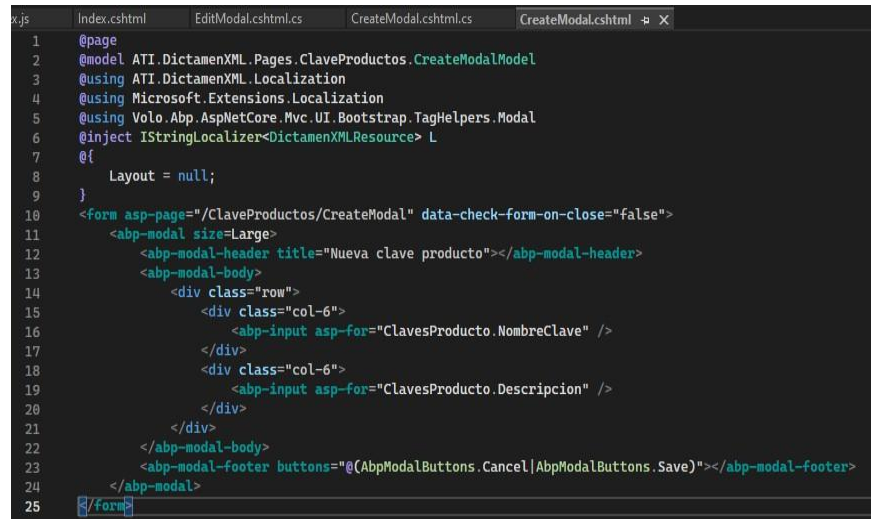


Imagen 10.- Página razor: "Clave Productos" – CreateModal.

La interfaz del proyecto se encuentra en la solución web, estando ordenada en una carpeta llamada "Pages", en donde se subdividen cada una de las entidades que se generaron para los requisitos del sistema. Todas las páginas cuentan con los servicios de aplicación que se utilizan para implementar los casos de uso. Estos exponen la lógica de dominio a la capa de presentación.

```
CreateModal.cshtml.cs  CreateModal.cshtml
ATI.DictamenXML
1 using ATI.DictamenXML.Services.Dtos.ClaveProductos;
2 using Microsoft.AspNetCore.Mvc;
3 using Microsoft.AspNetCore.Mvc.RazorPages;
4
5 namespace ATI.DictamenXML.Pages.ClaveProductos
6 {
7     [BindProperty]
8     public class CreateModalModel : DictamenXMLPageModel
9     {
10         public CreateUpdateClaveProductoDto ClavesProducto { get; set; }
11
12         //Inyección de dependencia
13         private readonly IClaveProductoAppService _claveProductoAppService;
14
15         public CreateModalModel(IClaveProductoAppService claveProductoAppService)
16         {
17             _claveProductoAppService = claveProductoAppService;
18         }
19
20         public void OnGet()
21         {
22             ClavesProducto = new CreateUpdateClaveProductoDto();
23         }
24
25         public async Task<IActionResult> OnPostAsync()
26         {
27             await _claveProductoAppService.CreateAsync(ClavesProducto);
28             return NoContent();
29         }
30     }
31 }
```

Imagen 11.- Página razor: "Clave Productos" – CreateModal.

En la última etapa, se realizaron pruebas de funcionalidad y usabilidad para garantizar que la plataforma cumple con los requerimientos definidos al principio del proyecto. Al ejecutar el Sistema Web creado, se obtienen las siguientes vistas y funcionalidades:

Log-in de usuario, vista de inicio de sesión en donde una vez dado de alta el usuario, podrá ingresar e interactuar con el Sistema.

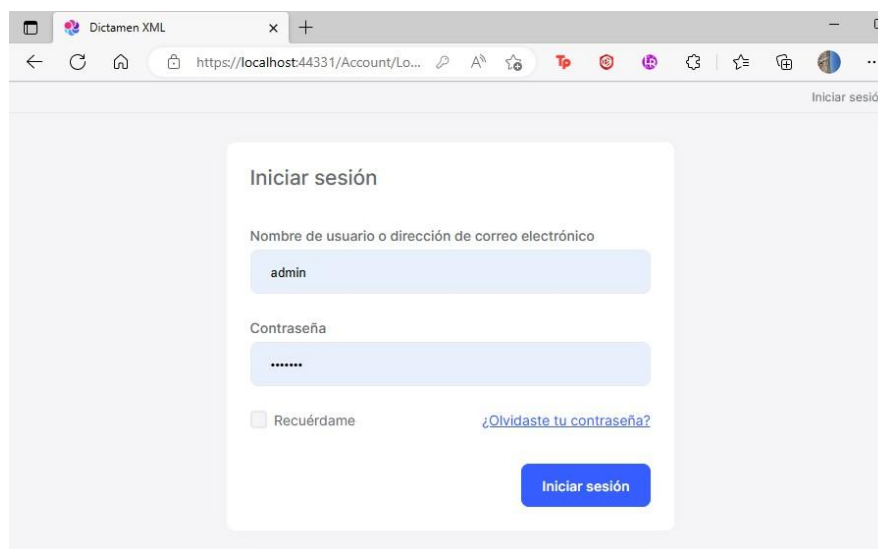


Imagen 12.- Interfaz de usuario gráfica, Log-in Aplicación.

Entre las vistas generadas en la interfaz de usuario, se encuentra nuestra primera sección correspondiente a Producto, en donde podemos visualizar en una tabla todos nuestros registros correspondientes.

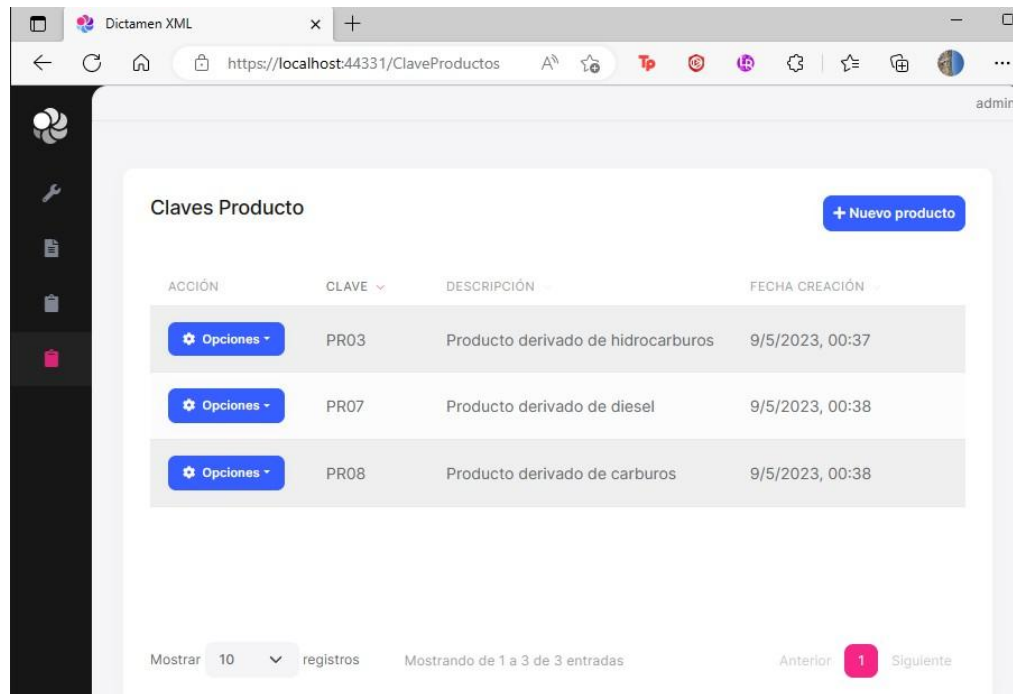


Imagen 13.- Interfaz de usuario gráfica, Productos.

En el módulo Dictamen podemos acceder a la creación de un "producto nuevo", al dar clic sobre el botón "Nuevo producto". Al accionarlo nos arrojará una vista emergente donde podremos ingresar los datos correspondientes a cada producto.

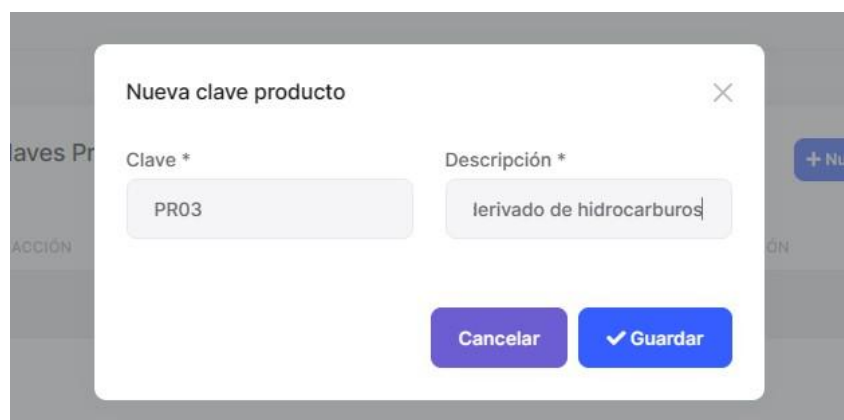


Imagen 14.- Interfaz de usuario gráfica, Nuevo Producto.

Para eliminar o editar el registro ya generado de un producto existente, se debe acceder al botón "Opciones" el cual se encuentra dentro de nuestra tabla de productos, en la columna "Acción", al dar clic nos muestra una Lista de acciones, en donde al seleccionar alguna, nos permitirá editar o borrar el registro.

ACCIÓN	CLAVE	DESCRIPCIÓN	FECHA CREACIÓN
Opciones Editar Eliminar	PR03	Producto derivado de hidrocarburos	9/5/2023, 00:37
	07	Producto derivado de diesel	9/5/2023, 00:38

Imagen 15.- Interfaz de usuario gráfica, Nuevo Producto.

La opción "Editar" nos arrojará la vista correspondiente a Editar, donde mostrará los datos del producto que seleccionamos y nos permitirá modificar y cambiar los campos correspondientes.

Editar

Clave *

PR03

Descripción *

Derivado de hidrocarburo:

Cancelar

✓ Guardar

Imagen 16.- Interfaz de usuario gráfica, Editar Producto.

Al dar clic en el botón "Guardar" los cambios se actualizarán y guardarán en el mismo registro del Producto Seleccionado. La acción "Borrar" nos arrojará una vista de alerta, donde nos alertará acerca de la eliminación del producto que seleccionamos.



Imagen 17.- Interfaz de usuario gráfica, Eliminar Producto.

En la vista correspondiente a **Contribuyente Proveedor**, podemos visualizar nuestra tabla con todos nuestros registros de contribuyentes.

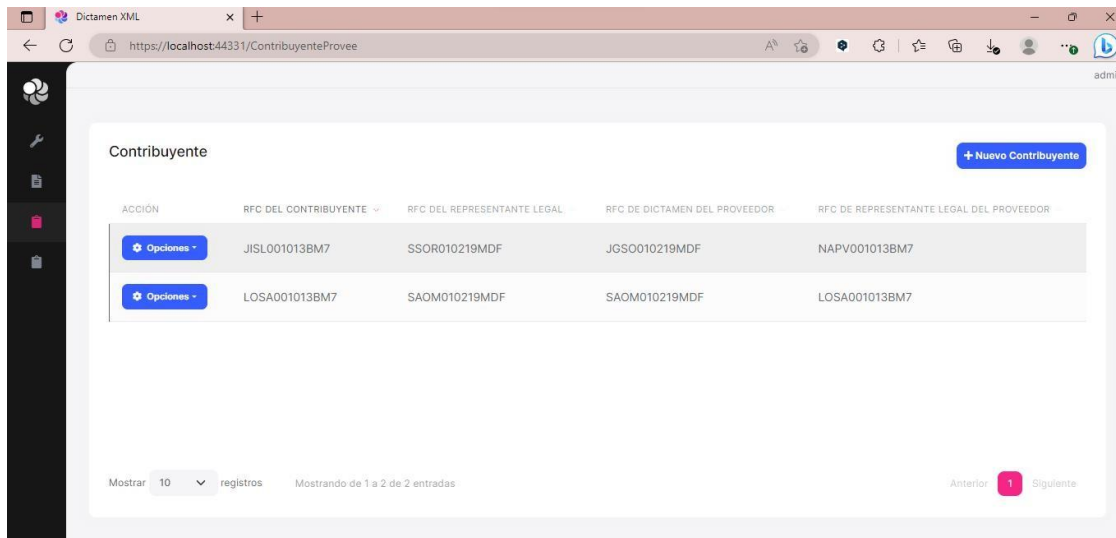


Imagen 18.- Interfaz de usuario gráfica, Productos.

En esta vista podemos acceder a la creación de un “contribuyente nuevo”, al dar clic sobre el botón “Nuevo contribuyente”. Al accionarlo nos arrojará una vista emergente donde podremos ingresar los datos correspondientes a cada producto.

Editar Contribuyente

×

RFC Contribuyente *

RFC Representante Legal

RFC del Proveedor del Dictamen *

RFC del Representante Legal del Proveedor

Cancelar

✓ Guardar

Imagen 19.- Interfaz de usuario gráfica, Nuevo Contribuyente.

Para eliminar o editar el registro ya generado de un contribuyente existente, se debe acceder al botón "Opciones" el cual se encuentra dentro de nuestra tabla de productos, en la columna "Acción", al dar clic nos muestra una Lista de acciones, en donde al seleccionar alguna, nos permitirá editar o borrar el registro.

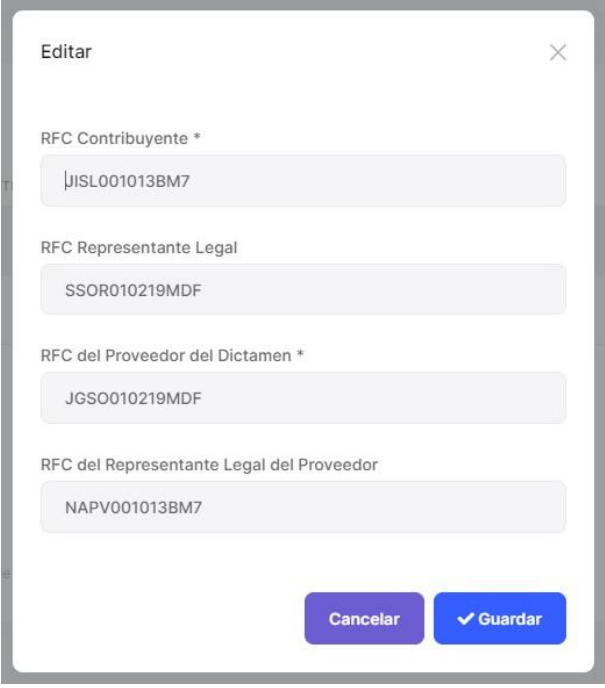
Contribuyente

+ Nuevo Contribuyente

Acción	RFC DEL CONTRIBUYENTE	RFC DEL REPRESENTANTE LEGAL	RFC DE DICTAMEN DEL PROVEEDOR	RFC DE REPRESENTANTE LEGAL DEL PROVEEDOR
Opciones Editar Eliminar	JISL001013BM7	SSOR010219MDF	JGSO010219MDF	NAPV001013BM7
	JSA001013BM7	SAOM010219MDF	SAOM010219MDF	LOSA001013BM7

Imagen 20.- Interfaz de usuario gráfica, Nuevo Contribuyente.

La opción “Editar” nos arrojará una nueva vista en donde mostrará los datos del contribuyente que seleccionamos y nos permitirá modificar y cambiar los campos correspondientes.

Una ventana modal con el título "Editar" y un botón de cerrar (X) en la esquina superior derecha. Contiene cuatro campos de texto con sus respectivos RFC: "RFC Contribuyente *" con el valor "JISL001013BM7", "RFC Representante Legal" con "SSOR010219MDF", "RFC del Proveedor del Dictamen *" con "JGSO010219MDF", y "RFC del Representante Legal del Proveedor" con "NAPV001013BM7". En la parte inferior hay dos botones: "Cancelar" y "✓ Guardar".

Editar

RFC Contribuyente *

JISL001013BM7

RFC Representante Legal

SSOR010219MDF

RFC del Proveedor del Dictamen *

JGSO010219MDF

RFC del Representante Legal del Proveedor

NAPV001013BM7

Cancelar ✓ Guardar

Imagen 21.- Interfaz de usuario gráfica, Editar Contribuyente.

Al dar clic en el botón “Guardar” los cambios se actualizarán y guardarán en el mismo registro del Producto Seleccionado. La acción “Borrar” nos arrojará una vista de alerta, donde nos alertará acerca de la eliminación del contribuyente seleccionado.

Una ventana modal de confirmación con un icono de advertencia (exclamación dentro de un círculo) en la parte superior. El texto principal dice "¿Estas seguro?". Debajo, en un tamaño de fuente menor, dice "Confirmar eliminacion de Contribuyente". En la parte inferior hay dos botones: "Cancelar" y "Sí".

!

¿Estas seguro?

Confirmar eliminacion de Contribuyente

Cancelar Sí

Imagen 22.- Interfaz de usuario gráfica, Eliminar Contribuyente.

Por último, el módulo de **Reporte Dictámenes**, podemos visualizar nuestra tabla con todos nuestros registros de Reportes.

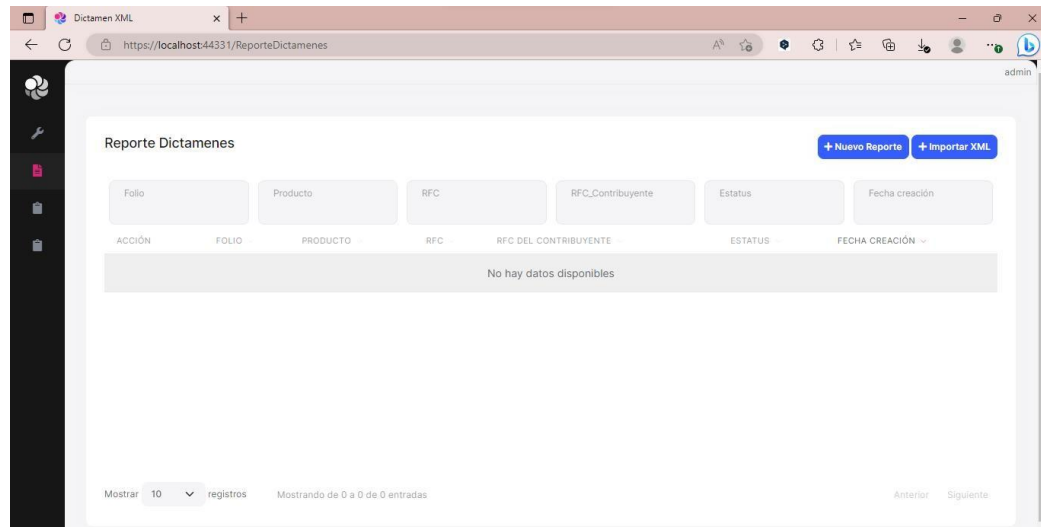


Imagen 23.- Interfaz de usuario gráfica, dictámenes.

En esta vista podemos acceder a la creación de un "Dictamen nuevo", al dar clic sobre el botón "Nuevo reporte". Al accionarlo nos arrojará una vista emergente donde podremos ingresar los datos correspondientes a cada reporte de dictamen.

A screenshot of a modal form titled 'Crear Dictamen'. The form contains several input fields: 'Fecha de Emision' with the value '26/01/0001 03:01 a. m.', 'Numero de Folio del Dictamen' with the value 'LOSA001013BM7123451425', 'Producto' with a dropdown menu showing 'Producto Interno Bruto - PR04', 'RFC Personal' with the value 'RONE151013VG7', and 'Seleccione Proveedor' with a dropdown menu showing 'LOSA001013BM7'. At the bottom of the form are two buttons: 'Cancelar' and '✓ Guardar'.

Imagen 24.- Interfaz de usuario gráfica, Nuevo Dictamen.

Para eliminar o editar el registro ya generado de un dictamen existente, se debe acceder al botón "Opciones" el cual se encuentra dentro de nuestra tabla de productos, en la columna "Acción", al dar clic nos muestra una Lista de acciones, en donde al seleccionar alguna, nos permitirá editar o borrar el registro.

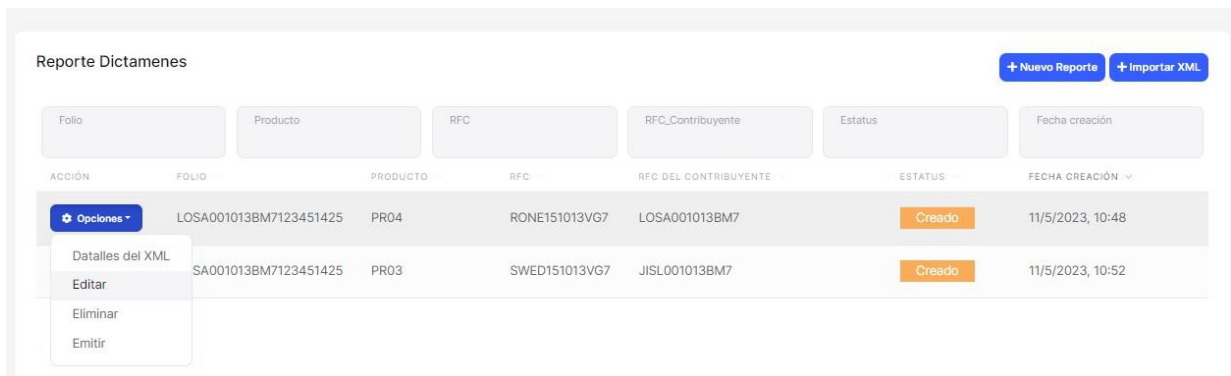


Imagen 25.- Interfaz de usuario gráfica, Nuevo Dictamen.

La opción "Editar" nos arrojará una nueva vista en donde mostrará los datos del dictamen que seleccionamos y nos permitirá modificar y cambiar los campos correspondientes.

The screenshot shows a modal window titled "Editar" with a close button (X) in the top right corner. The form contains the following fields: "Fecha de Emision *" with a date and time value "26/01/0001 03:01 a. m." and a calendar icon; "Numero de Folio del Dictamen" with the value "LOSA001013BM7123451425"; "Producto" with a dropdown menu showing "Producto Interno Bruto - PR04"; "RFC Personal" with the value "RONE151013VG7"; and "Seleccione Proveedor" with a dropdown menu showing "LOSA001013BM7". At the bottom of the form are two buttons: "Cancelar" and "✓ Guardar".

Imagen 26.- Interfaz de usuario gráfica, Editar Dictamen.

Al dar clic en el botón "Guardar" los cambios se actualizarán y guardarán en el mismo registro del Dictamen Seleccionado. La acción "Borrar" nos arrojará una vista de alerta, donde nos alertará acerca de la eliminación del contribuyente seleccionado.



Imagen 27.- Interfaz de usuario gráfica, Eliminar Dictamen.

Como podemos observar, cada modulo generado cubre con los aspectos básicos del sistema los cuales son el registro, modificación y eliminación de: Producto, su Contribuyente o Proveedor y su Dictamen generado según sea el caso. Podemos notar que los tres módulos tienen relación unos con los otros, pues comparten campos de registro y se extraen datos de una tabla a otra, que hacen la complementación del registro esperado.

Resultados y conclusiones

Los resultados obtenidos al finalizar el proyecto son satisfactorios, ya que se lograron cumplir con los objetivos establecidos y se obtuvo la funcionalidad esperada. Se espera que la plataforma permita llevar un registro detallado y en tiempo real de las cantidades de hidrocarburos y petrolíferos que se producen, almacenan, transportan y comercializan en México. Esto permitirá garantizar la transparencia y la legalidad en las transacciones, lo que a su vez contribuirá al fortalecimiento de la economía y al combate de la corrupción. Además, se espera que la plataforma sea escalable y pueda adaptarse a las necesidades futuras de los usuarios.

Beneficios de Aplicación del Sistema	CUMPLE O NO CUMPLE
Cumplimiento de las normativas y regulaciones gubernamentales.	SI
Acceso a información en tiempo real.	SI
Mayor eficiencia en el proceso de registro y control de las cantidades de hidrocarburos y petrolíferos.	SI
Mejora en la transparencia y legalidad de las transacciones de hidrocarburos y petrolíferos.	SI
Las empresas pueden asegurarse de cumplir con la nueva norma gubernamental de Controles Volumétricos.	SI
Ayudar a las empresas a ahorrar tiempo y recursos al automatizar ciertos procesos y al proporcionar herramientas de informes y análisis.	SI
Reducción de errores en los registros de volumen.	SI
Mayor transparencia y seguridad: Al llevar un registro detallado y preciso de las cantidades de hidrocarburos y petrolíferos que se producen, almacenan, transportan y comercializan, las empresas pueden mejorar su transparencia y seguridad en sus transacciones comerciales.	SI

Referencias

- Servicio de Administración Tributaria. (2024). Preguntas frecuentes sobre controles volumétricos. Recuperado de <https://omawww.sat.gob.mx>
- Diario Oficial de la Federación. (2024). Resolución Miscelánea Fiscal para 2024 (Anexos 30, 31 y 32). Recuperado de <https://www.dof.gob.mx>
- SGS México. (2024). Controles volumétricos de hidrocarburos: obligados para las empresas del sector. Recuperado de <https://www.sgs.com>
- Volumetrics. (2024). ¿Qué son los controles volumétricos del SAT? Recuperado de <https://volumetrics.com.mx>
- JD Group. (2024). Anexo 30 de la Resolución Miscelánea Fiscal: aspectos clave y actualizaciones. Recuperado de <https://www.jdgroup.net>
- Kernotek. (2024). Guía esencial para el control volumétrico en la industria de hidrocarburos y petrolíferos. Recuperado de <https://www.kernotek.mx>
- IP Oil & Gas Solutions. (2024). Medición y control volumétrico en la industria de hidrocarburos. Recuperado de <https://ip-oilgassolutions.com>
- Gobierno de México. (2024). Lo que debes saber de las nuevas obligaciones en materia de controles volumétricos. Recuperado de <https://www.gob.mx>
- Abp.io.(2024) Instalación de ABP.Recuperado de <https://abp.io/docs/latest/framework/fundamentals/application-startup>
- DBeaver.(2024).Instalación de Dbeaver.Recuperado de <https://dbeaver.com/docs/dbeaver/Installation/>
- Microsoft.(2024).Instalacion de Visual Studio 2022.Recuperado de <https://visualstudio.microsoft.com/es/vs/>