

# Universidad Abierta Para Adultos



Educación superior a distancia virtual

**Formación General**

**CARRERA**

**Ingeniería en Software**

**ASIGNATURA**

**Programación II**

**FACILITADOR**

**Diógenes Amaury Martínez Silverio**

**PARTICIPANTE**

**Vladimir Núñez Calderón**

**MATRICULA**

**100057484**

**TEMA**

**Trabajo Final**

**FECHA**

**15 de marzo del 2023**

**Santiago de los Caballeros, República Dominicana**

## INTRODUCCIÓN:

ADO.NET es una forma de trabajar con datos de diferentes lugares usando objetos y componentes que se parecen entre sí. ADO.NET usa proveedores de datos, que son partes del código que saben cómo hablar con cada tipo de lugar de datos (como SQL Server, Oracle o ODBC). Los proveedores de datos tienen objetos importantes como:

- El objeto Connection, que es como un cable que conecta con el lugar de datos y que se hace con una cadena de conexión que dice cómo conectarse.
- El objeto Command, que es como una orden que se manda al lugar de datos usando el cable Connection. El objeto Command puede traer resultados o cambiar los datos según la orden.
- El objeto DataReader, que es como un lector rápido que lee los resultados que trae el objeto Command uno por uno. El objeto DataReader sirve para leer los datos y no guardarlos en memoria.
- El objeto DataAdapter, que es como un puente entre el lugar de datos y el objeto DataSet. El objeto DataAdapter usa uno o varios objetos Command para traer los datos y llenar el DataSet o para mandar los cambios del DataSet al lugar de datos.
- El objeto DataSet, que es como una copia en memoria de los datos del lugar de datos. El objeto DataSet tiene uno o varios objetos DataTable, que son como tablas con filas y columnas. El objeto DataSet permite cambiar los datos sin usar el cable Connection y luego sincronizarlos con el lugar de datos usando el DataAdapter.

Entre las órdenes más usadas del objeto Command están ExecuteNonQuery y ExecuteScalar. ExecuteNonQuery se usa para mandar órdenes al lugar de datos que no traen ningún resultado, como cambiar o borrar los datos, o para hacer cosas en la base de datos, como ver cómo está hecha o crear objetos nuevos. ExecuteScalar se usa para mandar órdenes al lugar de datos que traen un valor único, como un número que representa una suma o un conteo de datos.

Temas	Autor	Opinión
Ado.Net	ADO.NET es una tecnología de acceso a datos del marco .NET de Microsoft que proporciona comunicación entre sistemas relacionales y no relacionales a través de un conjunto común de componentes. ADO.NET está diseñado para permitir que los desarrolladores escriban código administrado para obtener acceso sin conexión a las fuentes de datos, que pueden ser relacionales o no. ADO.NET también incluye el Entity Data Model, un modelo de datos conceptual que se puede usar para modelar los datos de un dominio determinado para que las aplicaciones puedan interactuar con los datos como objetos.	ADO.NET es una forma de conectar tu programa con una base de datos y trabajar con los datos que hay en ella. Puedes usar ADO.NET para leer, escribir, modificar y borrar datos de diferentes tipos de bases de datos. También puedes usar ADO.NET para crear una representación de los datos en tu programa que sea más fácil de entender y manipular.
Proveedores de datos	Los proveedores de datos son componentes de ADO.NET que permiten conectarse a una base de datos, ejecutar comandos y leer datos de la base de datos. Los proveedores de datos residen como una capa ligera entre la fuente de datos y el código, proporcionando servicios de acceso a datos con mayor rendimiento. Los proveedores de datos son específicos para cada tipo de base de datos, como SQL Server, Oracle, XML, etc. y se componen de clases como Connection, Command, DataReader y DataAdapter.	Los proveedores de datos son piezas de código que te ayudan a comunicarte con una base de datos desde tu programa. Cada base de datos tiene su propio proveedor que sabe cómo hablar con ella. Los proveedores de datos tienen clases que te permiten hacer cosas como abrir una conexión, enviar una consulta, leer los resultados y cerrar la conexión.
Objeto Connection	El objeto Connection en C# es un componente de ADO.NET que se utiliza para conectar con un origen de datos mediante una cadena de conexión. El objeto Connection utilizado depende del proveedor de datos que se elija para trabajar con un tipo de base de datos específico. Por ejemplo, para conectarse a SQL Server se usa el objeto SqlConnection y para conectarse a Oracle se usa el objeto OracleConnection. El objeto Connection tiene métodos para abrir y cerrar la conexión, así como eventos para recuperar	El objeto Connection es una pieza de código que te permite hablar con una base de datos desde tu programa en C#. Cada base de datos tiene su propio objeto Connection que sabe cómo conectarse con ella. El objeto Connection también te permite saber si la conexión está abierta o cerrada y si hay algún mensaje o error de la base de datos.

	mensajes informativos o detectar cambios en el estado de la conexión.	
Objeto Command	<p>El objeto Command en C# es un componente de ADO.NET que se utiliza para ejecutar una consulta o un comando en una base de datos y devolver los resultados en un objeto DataReader o un objeto DataSet. El objeto Command también se puede usar para ejecutar un procedimiento almacenado o una operación masiva. El objeto Command debe estar asociado a un objeto Connection que indique el origen de datos al que se envía el comando. El objeto Command tiene propiedades para especificar el texto del comando, el tipo de comando, los parámetros y la duración del tiempo de espera.</p>	<p>El objeto Command es una pieza de código que te permite enviar una orden o una pregunta a una base de datos desde tu programa en C#. El objeto Command necesita saber con qué base de datos se comunica usando el objeto Connection. El objeto Command también necesita saber qué orden o pregunta quieres enviar, cómo quieres enviarla y cuánto tiempo puedes esperar la respuesta. El objeto Command te devuelve la respuesta de la base de datos en otro objeto que puedes leer o guardar.</p>
Objeto DataTable	<p>El objeto DataTable en ADO.NET es un componente que representa una tabla de datos relacionales en la memoria. El objeto DataTable puede contener filas y columnas que almacenan los datos, así como restricciones y relaciones que definen las reglas de integridad y asociación entre los datos. El objeto DataTable se puede crear y usar de manera independiente o como parte de un objeto DataSet, que es una colección de tablas y otros objetos relacionados con los datos. El objeto DataTable se puede llenar desde un origen de datos externo como una base de datos mediante un objeto DataAdapter.</p> <p>El objeto DataTable C# es el mismo que el objeto DataTable ADO.NET, solo que se usa en el lenguaje de programación C#. El objeto DataTable C# se puede declarar e instanciar usando la sintaxis de C#</p>	<p>El objeto DataTable C# es una pieza de código que te permite crear y usar una tabla de datos en tu programa en C#. Una tabla de datos es como una hoja de cálculo que tiene filas y columnas para guardar información. El objeto DataTable C# te permite crear la tabla con el nombre y el tamaño que quieras, y llenarla con los datos que necesites. El objeto DataTable C# también te permite conectar la tabla con otras fuentes de datos como una base de datos o un archivo.</p>
Objeto DataReader	El objeto DataReader en C# es un componente de ADO.NET que se utiliza para leer los datos de un origen de datos	El objeto DataReader C# es una pieza de código que te permite leer los datos de

	<p>como una base de datos de forma secuencial y eficiente. El objeto <code>DataReader</code> se crea mediante la llamada al método <code>Command.ExecuteReader</code>, que envía una consulta o un comando al origen de datos y devuelve un objeto <code>DataReader</code> con los resultados. El objeto <code>DataReader</code> tiene métodos para avanzar por las filas del resultado y acceder a los valores de cada columna. El objeto <code>DataReader</code> es una buena opción cuando se trata de recuperar grandes cantidades de datos, ya que no los almacena en memoria sino que los lee a medida que están disponibles.</p>	<p>una base de datos desde tu programa en C#. El objeto <code>DataReader</code> C# necesita saber con qué base de datos se comunica usando el objeto <code>Command</code>. El objeto <code>DataReader</code> C# también necesita saber qué datos quieres leer, cómo quieres leerlos y cuánto tiempo puedes esperar la respuesta. El objeto <code>DataReader</code> C# te devuelve los datos de la base de datos en un formato que puedes usar o guardar.</p>
Objeto <code>DataAdapter</code>	<p>El objeto <code>DataAdapter</code> en C# es un componente de ADO.NET que se utiliza para recuperar datos de un origen de datos como una base de datos y llenar un objeto <code>DataSet</code> con ellos. El objeto <code>DataSet</code> es una representación en memoria de los datos que se pueden manipular y actualizar sin conexión al origen de datos. El objeto <code>DataAdapter</code> también se utiliza para actualizar el origen de datos con los cambios realizados en el objeto <code>DataSet</code> mediante su método <code>Update</code>. El objeto <code>DataAdapter</code> usa la asignación para hacer coincidir las columnas del origen de datos con las del objeto <code>DataSet</code> usando el objeto <code>DataTableMapping</code>.</p>	<p>El <code>DataAdapter</code> en C# es una herramienta que sirve para traer datos de una base de datos y guardarlos en un <code>DataSet</code>. El <code>DataSet</code> es como una copia de los datos que se puede usar y cambiar sin estar conectado a la base de datos. El <code>DataAdapter</code> también sirve para enviar los cambios del <code>DataSet</code> a la base de datos con el método <code>Update</code>. El <code>DataAdapter</code> usa el <code>DataTableMapping</code> para relacionar las columnas de la base de datos con las del <code>DataSet</code>.</p>
Objeto <code>DataSet</code>	<p>El objeto <code>DataSet</code> en C# es una clase de ADO.NET que representa un conjunto de datos en memoria que se puede manipular y actualizar sin conexión a la base de datos. Un objeto <code>DataSet</code> puede contener una o más tablas (objetos <code>DataTable</code>) y las relaciones entre ellas (objetos <code>DataRelation</code>). Para crear un objeto <code>DataSet</code> se puede usar la sintaxis:</p> <p><code>DataSet nombre = new DataSet ("nombre");</code></p>	<p>Un <code>DataSet</code> es como una copia de una parte de la base de datos que se puede usar en el programa sin necesidad de estar conectado a la base de datos todo el tiempo. Un <code>DataSet</code> puede tener varias tablas y las conexiones entre ellas. Para crear un <code>DataSet</code> se le pone un</p>

	<p>donde nombre es el identificador del objeto DataSet y “nombre” es el nombre del conjunto de datos.</p>	<p>nombre y se le dice qué tablas va a tener.</p>
ExecuteNonQuery	<p>El ExecuteNonQuery en C# es un método de la clase SqlCommand que se usa para ejecutar instrucciones SQL que no devuelven ningún resultado, como UPDATE, INSERT o DELETE. También se puede usar para realizar operaciones de catálogo, como consultar la estructura de una base de datos o crear objetos de base de datos como tablas. Para usar el ExecuteNonQuery se necesita crear un objeto SqlCommand con la instrucción SQL y la cadena de conexión al origen de datos.</p>	<p>El ExecuteNonQuery es una forma de enviar órdenes a la base de datos que no esperan una respuesta, como cambiar, agregar o borrar datos. También se puede usar para ver o crear cosas en la base de datos, como tablas. Para usar el ExecuteNonQuery se necesita decir qué orden se quiere enviar y cómo conectarse a la base de datos.</p>
ExecuteScalar	<p>El ExecuteScalar en C# es un método de la clase SqlCommand que se usa para ejecutar instrucciones SQL que devuelven un valor único, como un valor agregado (por ejemplo, una suma o un conteo) de una base de datos. Esto requiere menos código que usar el ExecuteReader método y luego procesar los datos devueltos por un SqlDataReader. Para usar el ExecuteScalar se necesita crear un objeto SqlCommand con la instrucción SQL y la cadena de conexión al origen de datos.</p>	<p>El ExecuteScalar es una forma de enviar órdenes a la base de datos que esperan una respuesta de un solo valor, como un número que representa una suma o un conteo de datos. Esto requiere menos código que enviar órdenes que devuelven varios datos y luego buscar el valor que se quiere. Para usar el ExecuteScalar se necesita decir qué orden se quiere enviar y cómo conectarse a la base de datos.</p>

## **CONCLUSIÓN:**

ADO.NET es una forma muy buena y fácil para trabajar con diferentes tipos de lugares de datos desde las aplicaciones .NET Framework. ADO.NET tiene varios objetos y componentes que ayudan a comunicarse, preguntar, leer, guardar y cambiar los datos según lo que quiera el desarrollador. Los proveedores de datos .NET Framework tienen una parte específica para cada tipo de lugar de datos e usan objetos comunes como Connection, Command, DataReader, DataAdapter y DataSet. Estos objetos permiten trabajar con los datos conectados o desconectados, y tienen órdenes como ExecuteNonQuery y ExecuteScalar para mandar diferentes tipos de órdenes al lugar de datos.