1. **INTRODUCCION A LAS HERRAMIENTAS.**

**SQL SERVER 2008, 20012 y 2014**.

Microsoft SQL Server es un sistema de manejo de bases de datos del [modelo relacional](https://es.wikipedia.org/wiki/Base_de_datos_relacional), desarrollado por la empresa [Microsoft](https://es.wikipedia.org/wiki/Microsoft).

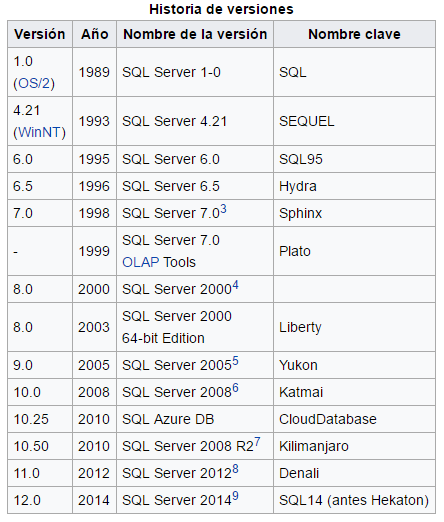
El lenguaje de desarrollo utilizado (por línea de comandos o mediante la interfaz gráfica de Management Studio) es [Transact-SQL](https://es.wikipedia.org/wiki/Transact-SQL" \o "Transact-SQL) (TSQL), una implementación del estándar ANSI del lenguaje SQL, utilizado para manipular y recuperar datos ([DML](https://es.wikipedia.org/wiki/Lenguaje_de_manipulaci%C3%B3n_de_datos)), crear tablas y definir relaciones entre ellas ([DDL](https://es.wikipedia.org/wiki/Lenguaje_de_definici%C3%B3n_de_datos)).

Dentro de los competidores más destacados de SQL Server están: [Oracle](https://es.wikipedia.org/wiki/Oracle), [MariaDB](https://es.wikipedia.org/wiki/MariaDB" \o "MariaDB), [MySQL](https://es.wikipedia.org/wiki/MySQL" \o "MySQL), [PostgreSQL](https://es.wikipedia.org/wiki/PostgreSQL" \o "PostgreSQL). SQL Server solo está disponible para [sistemas operativos](https://es.wikipedia.org/wiki/Sistema_operativo) [Windows](https://es.wikipedia.org/wiki/Windows) de Microsoft.

Puede ser configurado para utilizar varias instancias en el mismo servidor físico, la primera instalación lleva generalmente el nombre del servidor, y las siguientes - nombres específicos (con un guion invertido entre el nombre del servidor y el nombre de la instalación).

Versiones.

El [código fuente](https://es.wikipedia.org/wiki/C%C3%B3digo_fuente) original de SQL Server que fue utilizado en las versiones previas a la versión 7.0 habría sido comprado de [Sybase](https://es.wikipedia.org/wiki/Sybase" \o "Sybase), pero fue actualizado en las versiones 7.0 y 2000, y reescrito en la versión 2005. Generalmente, cada 2-3 años, una nueva versión es lanzada y, entre estos lanzamientos, se proponen [service packes](https://es.wikipedia.org/wiki/Service_Pack" \o "Service Pack) con mejoras y correcciones de [bugs](https://es.wikipedia.org/wiki/Bug), y [hotfixes](https://es.wikipedia.org/wiki/Hotfix" \o "Hotfix) por problemas urgentes en el sistema de seguridad o bugs críticos.



Características.

* Soporte de [transacciones](https://es.wikipedia.org/wiki/Transacci%C3%B3n_(base_de_datos)).
* Soporta [procedimientos almacenados](https://es.wikipedia.org/wiki/Procedimientos_almacenados).
* Incluye también un [entorno gráfico](https://es.wikipedia.org/wiki/Entorno_gr%C3%A1fico) de administración, que permite el uso de [comandos](https://es.wikipedia.org/wiki/Comando_(inform%C3%A1tica)) [DDL](https://es.wikipedia.org/wiki/Lenguaje_de_definici%C3%B3n_de_datos) y [DML](https://es.wikipedia.org/wiki/DML) gráficamente.
* Permite trabajar en modo [cliente-servidor](https://es.wikipedia.org/wiki/Cliente-servidor), donde la información y datos se alojan en el [servidor](https://es.wikipedia.org/wiki/Servidor) y los [terminales](https://es.wikipedia.org/wiki/Terminal_(inform%C3%A1tica)) o [clientes](https://es.wikipedia.org/wiki/Cliente_(inform%C3%A1tica)) de la [red](https://es.wikipedia.org/wiki/Red_de_computadoras) sólo acceden a la información.
* Además, permite administrar información de otros [servidores](https://es.wikipedia.org/wiki/Servidor) de datos.

Este sistema incluye una versión reducida, llamada [MSDE](https://es.wikipedia.org/wiki/Microsoft_SQL_Server_Data_Engine) con el mismo motor de [base de datos](https://es.wikipedia.org/wiki/Base_de_datos) pero orientado a proyectos más pequeños, que en sus versiones 2005 y 2008 pasa a ser el [SQL Express Edition](https://es.wikipedia.org/wiki/SQL_Express_Edition), que se distribuye en forma [gratuita](https://es.wikipedia.org/wiki/Freeware).

Es común desarrollar proyectos completos empleando Microsoft SQL Server y [Microsoft Access](https://es.wikipedia.org/wiki/Microsoft_Access) a través de los llamados ADP ([Access Data Project](https://es.wikipedia.org/w/index.php?title=Access_Data_Project&action=edit&redlink=1)). De esta forma se completa la [base de datos](https://es.wikipedia.org/wiki/Base_de_datos) (Microsoft SQL Server), con el entorno de desarrollo ([VBA](https://es.wikipedia.org/wiki/VBA) Access), a través de la implementación de aplicaciones de dos capas mediante el uso de formularios [Windows](https://es.wikipedia.org/wiki/Windows).

En el manejo de SQL mediante líneas de comando se utiliza el [SQLCMD](https://es.wikipedia.org/wiki/SQLCMD), o SQL, o PowerShell.

Para el desarrollo de aplicaciones más complejas (tres o más capas), Microsoft SQL Server incluye interfaces de acceso para varias plataformas de desarrollo, entre ellas [.NET](https://es.wikipedia.org/wiki/.NET_de_Microsoft), pero el servidor sólo está disponible para [Sistemas Operativos](https://es.wikipedia.org/wiki/Sistemas_operativos).

El tipo NUMERIC fue mejorado para ser usado como identificador de columna a partir de la versión 2008 R2.

Programación.

T-SQL.

Artículo principal: [T-SQL](https://es.wikipedia.org/wiki/T-SQL)

[T-SQL](https://es.wikipedia.org/wiki/T-SQL) (Transact-SQL) es el principal medio de interacción con el Servidor, el cual permite realizar las operaciones claves en SQL Server, incluyendo la creación y modificación de esquemas de base de datos, inserción y modificación de datos en la base de datos, así como la administración del servidor como tal. Esto se realiza mediante el envío de sentencias en T-SQL y declaraciones que son procesadas por el servidor y los resultados (o errores) regresan a la aplicación cliente.

Cliente Nativo de SQL.

Cliente Nativo de SQL, es la biblioteca de acceso a datos para los clientes de Microsoft SQL Server versión 2005 en adelante. Implementa de forma nativa soporte para las características de SQL Server, incluyendo la ejecución de la secuencia de datos tabular, soporte para bases de datos en espejo de SQL Server, soporte completo para todos los tipos de datos compatibles con SQL Server, conjuntos de operaciones asíncronas, las notificaciones de consulta, soporte para cifrado, así como recibir varios conjuntos de resultados en una sola sesión de base de datos. Cliente Nativo de SQL se utiliza como extensión de SQL Server plug-ins para otras tecnologías de acceso de datos, incluyendo ADO u OLE DB. Cliente Nativo de SQL puede también usarse directamente, pasando por alto las capas de acceso de datos.

Ediciones y servicios.

Cada versión de SQL Server posee distintas versiones con distintos precios (para cada versión) que dependen también en la configuración física del servidor. [10](https://es.wikipedia.org/wiki/Microsoft_SQL_Server#cite_note-10) A continuación se presentan las versiones principales:

Enterprise.

Contempla todas las características (deshabilitadas en otras ediciones). Es el tipo de versión con más privilegios existente en el mercado.

Developer.

Una edición con las mismas características que la Enterprise, con el fin de ser instalada solamente en ambiente de desarrollo y no en producción. Si se desarrolla para una edición Standard hay que tener en cuenta las características deshabilitadas para esta versión.

Standard.

Una versión limitada según la configuración del servidor y sus características, diseñada para servidores inferiores.

Por ejemplo: en la versión 2012, la edición Enterprise soporta un número ilimitado de [procesadores](https://es.wikipedia.org/wiki/Unidad_central_de_procesamiento), y la agregación de [memoria](https://es.wikipedia.org/wiki/Memoria_de_acceso_aleatorio) y [CPUs](https://es.wikipedia.org/wiki/Unidad_central_de_procesamiento" \o "Unidad central de procesamiento) en caliente sin la interrupción del [servicio](https://es.wikipedia.org/wiki/Servicio_de_Windows) o del servidor; mientras la edición Standard está limitada a 16 procesadores y no soporta la "agregación en caliente".

Express.

Una versión gratuita que posibilita la creación de bases de datos limitadas con características básicas, con el fin de apoyar aplicaciones que necesiten una solución simple para almacenamiento de una cantidad limitada de datos, o usuarios que sus recursos y necesidades son limitados.

En la versión 2012, esta edición puede utilizar un máximo de 1 GB de memoria, y almacenar no más de 10GB, funciona en servidores con un número máximo de cuatro procesadores. Estas limitaciones se mantienen en la versión 2014 (4 cores, 1GB ram, y 10Gb por base de datos).

SQL Azure.

Es una versión de SQL Server en la nube, que permite pagar mensualmente por el servicio sin la necesidad de mantener un servidor físico (On Premise). La empresa paga solo por el servicio, y el servicio es manejado a través de [torres de servidores](https://es.wikipedia.org/wiki/Torre_de_servidores) en distintos lugares en el mundo.

Con SQL Azure no es necesario instalar, mantener o actualizar un servidor físico; a pesar que este servicio depende de aspectos relacionados a problemas de seguridad con respecto a su presencia fuera de la empresa y a la disponibilidad de conexión a Internet.

Durante un tiempo, el servicio fue ampliado con la opción de crear un [servidor virtual](https://es.wikipedia.org/wiki/M%C3%A1quina_virtual) por la red, e instalar SQL Server tanto como uno de los servicios competidores, y manejar el servidor virtual como si fuera un servidor físico local (aunque físicamente no está accesible); y se puede diferenciar entre la opción original que esta denominada PAAS (Platform as a Service: El Plataforma como un Servicio) y la nueva opción de los servidores virtuales denominada IAAS (Infrastructure as a Service, Infraestructura como un Servicio).

Este servicio esta otorgado por Microsoft desde [2009](https://es.wikipedia.org/wiki/2009) y se une a servicios similares de [empresas de third-party](https://es.wikipedia.org/wiki/Third-party).

Interfaz de usuario.

SQL Server proporciona unos interfaz que han cambiado durante los años, de los cuales los más conocidos son los interfaz gráficos que están utilizados como herramienta de desarrollo estándar a los [desarrolladores](https://es.wikipedia.org/wiki/Desarrollador_de_software) y [administradores](https://es.wikipedia.org/wiki/Administrador_de_base_de_datos).

La interfaz gráfica hasta 2005 incluyó el Enterprise Manager con una vista de árbol de los distintos objetos y con la capacidad de manejarlos; y el Query analyzer como interfaz textual para ejecutar comandos de TSQL.

En la versión 2005 las dos herramientas se unificaron a una –el SQL Server Management Studio (SSMS), y a partir de 2008 fue incluida la opción de trabajar con el [Visual Studio](https://es.wikipedia.org/wiki/Visual_Studio)– la interfaz estándar de desarrollo de Microsoft (a los distintos [lenguajes](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n), [BI](https://es.wikipedia.org/wiki/Inteligencia_empresarial), etc.). Otro interfaz opcional es la utilización de [Línea de comandos](https://es.wikipedia.org/wiki/L%C3%ADnea_de_comandos), con herramientas como SQLCmd, ISQL, OSQL que posibilita la ejecución de [scripts](https://es.wikipedia.org/wiki/Script) y [procesamiento por lotes](https://es.wikipedia.org/wiki/Procesamiento_por_lotes). Desde 2008 se puede desarrollar con SQLCmd (SQL Command) a través del SSMS sin interconectarse al interfaz textual de Windows. Otra opción en el ámbito de scripts es la utilización del lenguaje de scripts [Powershell](https://es.wikipedia.org/wiki/Windows_PowerShell" \o "Windows PowerShell) de Microsoft.

Aparte de los interfaces estándares de SQL Server, se puede ejecutar comandos de TSQL con herramientas de conexión como ODBC y OLE-DB.[11](https://es.wikipedia.org/wiki/Microsoft_SQL_Server#cite_note-11)

Servicios.

A contrario de sistemas de bases de datos como [Microsoft Access](https://es.wikipedia.org/wiki/Microsoft_Access) que son "pasivas" y contienen un archivo a cual hay que conectar y la ejecución de los comandos se lleva a cabo en el [cliente](https://es.wikipedia.org/wiki/Cliente_(inform%C3%A1tica)) (la computadora de usuario), en SQL Server hay número de [servicios](https://es.wikipedia.org/wiki/Servicio_de_Windows), software que están ejecutadas en la memoria del servidor por parte del sistema, y por lo tanto aprovechan las capacidades del servidor que es más potente que los clientes, previenen congestión en la [red](https://es.wikipedia.org/wiki/Red_de_computadoras), y pueden programar tareas que corran aunque el cliente no está conectado.

Los servicios principales:

* SQL Server - El "motor" del sistema
* SQL Agent - Ejecución de tareas (Jobs, scripts programados) y envió de advertencias en caso de carga pesada e irregulares en el sistema
* Full-Text Filter Daemon Launcher - La utilización en los indexes especiales del "Full text search" por búsqueda textual avanzada
* SQL Browser - El "oyente" dedicado a comandos enviados y redirigirlos a su destino
* SSIS Server - La operación del [SSIS](https://es.wikipedia.org/wiki/SQL_Server_Integration_Services) (la herramienta de [ETL](https://es.wikipedia.org/wiki/Extract,_transform_and_load))
* SSAS Server - La operación del [SSAS](https://es.wikipedia.org/w/index.php?title=SQL_Server_Analysis_Services&action=edit&redlink=1) (la herramienta de [OLAP](https://es.wikipedia.org/wiki/OLAP))
* SSRS Server - La operación del [SSRS](https://es.wikipedia.org/w/index.php?title=SQL_Server_Reporting_Services&action=edit&redlink=1) (la herramienta de informes)

Capacidades y herramientas básicas.

Bases de datos.

En cada instalación de SQL Server hay 4 bases de datos de sistema, y la capacidad de crear nuevas bases de datos por el usuario, en los cuales los datos están almacenados en tablas.

Estas bases de datos, creadas por parte de los usuarios, incluyen básicamente un archivo de datos (con el sufijo mdf) con las tablas y los distintos objetos a nivel de la base de datos; y un archivo de registro (con el sufijo ldf) con las transacciones abiertas, y transacciones cerradas, Sujeto al modelo de recuperación seleccionado (se puede acumular en el archivo de registro todos los cambios en la base de datos desde el último respaldo). Se puede crear un conjunto de archivos de datos además del principal (con el sufijo ndf) por consideraciones de eficiencia, partición de carga de trabajo entre los discos rígidos, etc.

Las bases de datos del sistema:

* master - Todos los procedimientos, funciones y tablas del sistema que están utilizadas por parte de todas las bases de datos y que están instaladas automáticamente, tanto como las que han sido creado por parte de los administradores del sistema. Además, todas las definiciones en respecto a la seguridad a nivel del servidor, están almacenadas en esta base de datos.
* msdb - Almacenamiento de las tareas del agente, los códigos de [CLR](https://es.wikipedia.org/wiki/CLR) combinados en el sistema, los paquetes de SSIS, y otros más.
* model - El molde de las bases de datos. Cada nueva base de datos se crea como una copia de esta base de datos, menos que algo más estaba definido explícitamente.
* tempdb - Base de datos temporal que se crea de nuevo cada vez que el servicio reinicia. Se utiliza para almacenar tablas temporales creadas por parte de los usuarios o el sistema (por ejemplo, en ordenaciones complejos).

Tablas fijas y temporales.

Desde la perspectiva lógica, los datos almacenados en las bases de datos en tablas, que mediante ellas se implementa la teoría de las bases de datos relacionales. La tabla se divide en filas y columnas (A veces se les conoce como registros y campos). Las tablas pueden ser fijas o temporales, mientras que en el segundo caso existen físicamente en la base de datos tempdb, y se borran automáticamente en caso de desconexión de la sesión o de la conexión al servidor, depende en el tipo de la tabla temporal.

Desde la perspectiva física, el sistema divide los archivos de la base datos en Extents de 64 KB, y cada cual a ocho páginas de 8 KB. Generalmente, cada Extent se asigna a una tabla o un [índice](https://es.wikipedia.org/wiki/%C3%8Dndice_(base_de_datos)), menos las tablas pequeñas; y cada página se asigna siempre a una tabla específica. El sistema es responsable del aumento de los archivos, de acuerdo con los ajustes del usuario, y de asignar Extents y páginas a las tablas.

A las tablas se puede crear índices. Los índices se almacenan junto a la tabla (Non Clustered Index) o son la tabla en sí (Clustered Index). Los índices asisten en la búsqueda de datos en las tablas (como los ficheros en las librerías), en ordenarlas, y la definición de [claves primarias](https://es.wikipedia.org/wiki/Clave_primaria).

Entre las tablas se puede crear una [relación de uno a muchos](https://es.wikipedia.org/wiki/Clave_for%C3%A1nea).

Aparte de las tablas de los usuarios, hay tablas que almacenan meta data: datos sobre el sistema mismo, los diferentes objetos, los derechos, estadísticas sobre el rendimiento del sistema (DMV), etc.

Tipos de datos.

Para cada columna en una tabla y a cada variable o parámetro, se define un tipo de datos que sean almacenados en él, entre ellos:

1. Números: Números enteros y no enteros en distintos tamaños, y en diferentes niveles de precisión; y auto incremento opcional.
2. Textos: Cadenas de distintas longitudes, y distintas capacidades de apoyar distintas lenguas.
3. Fechas: Fechas en distintos niveles de precisión, desde días completos hasta fracciones menores de un segundo, que apoyan fechas a partir del principio del siglo 20 o del calendario gregoriano, y la capacidad de diferenciar entre distintos usos de horarios.
4. [XML](https://es.wikipedia.org/wiki/XML): Datos textuales (cadenas) que representan conjuntos estándares de datos (estándar SGML).
5. Datos binarios: Datos almacenados como datos binarios (bits y bytes), que posibilitan el almacenamiento de archivos gráficos, etc.
6. Geography: Representación estándar de información geográfica, tales como estados, zonas geográficas, localidades; y los cálculos como distancias.
7. Geometry: Representación estándar de puntas, líneas, superficies en el plano; y las relaciones entre ellas.
8. Hierarchid: Representación estándar de [información jerárquica](https://es.wikipedia.org/wiki/Jerarqu%C3%ADa) como lista de materiales, relaciones de subordinación entre empleados, etc.

Vistas.

Las vistas representan generalmente comandos de extracción de datos, que se almacenan sin los datos (que están almacenados en las tablas). Esta opción nos posibilita crear extracciones complejas o estándares, almacenarlas como vistas, y utilizar las vistas sin la necesidad de escribir de nuevo los comandos o mantener los códigos donde ellas aparecen. Adicionalmente, es un medio muy importante para otorgar derechos selectivos de lectura (en caso que queremos posibilitar a un usuario contemplar parcialmente las columnas o las filas de una tabla).

Una vista se puede considerar una tabla virtual o una consulta almacenada. Los datos accesibles a través de una vista no están almacenados en un objeto distinto de la base de datos. Lo que está almacenado en la base de datos es una instrucción SELECT. El resultado de la instrucción SELECT forma la tabla virtual que la vista devuelve. El usuario puede utilizar dicha tabla virtual haciendo referencia al nombre de la vista en instrucciones Transact-SQL, de la misma forma en que se hace referencia a las tablas. Las vistas se utilizan para alguna de estas funciones, o para todas:  
• Restringir el acceso del usuario a filas concretas de una tabla. Por ejemplo, permitir que un empleado sólo vea las filas que guardan su trabajo en una tabla de seguimiento de actividad laboral.  
• Restringir el acceso del usuario a columnas específicas. Por ejemplo, permitir que los empleados que no trabajen en el departamento de nóminas vean las columnas de nombre, oficina, teléfono y departamento de la tabla de empleados, pero no permitir que vean las columnas con los datos de salario u otra información personal.  
• Combinar columnas de varias tablas de forma que parezcan una sola tabla.  
• Agregar información en lugar de presentar los detalles. Por ejemplo, presentar la suma de una columna o el valor máximo o mínimo de una columna.

Las vistas se crean definiendo la instrucción SELECT que recupera los datos presentados por la vista. Las tablas de datos a las que hace referencia la instrucción SELECT se conocen como las tablas base para la vista. Las vistas en todas las versiones de SQL Server son actualizables (pueden ser objetivo de instrucciones UPDATE, DELETE o INSERT) mientras la modificación afecte sólo a una de las tablas base de la vista.

Procedimientos almacenados.

Los procedimientos son scripts de comandos de TSQL, que pueden ser ejecutados con distintos parámetros. Por ejemplo, procedimiento que obtiene número de año como parámetro, y actualiza una tabla de resumen de ventas, con las ventas de los agentes en el dicho año, basada en la tabla de registro de ventas.

Los procedimientos almacenados pueden facilitar en gran medida la administración de la base de datos y la visualización de información sobre dicha base de datos y sus usuarios. Los procedimientos almacenados son una colección precompilada de instrucciones SQL e instrucciones de control de flujo opcionales almacenadas bajo un solo nombre y procesadas como una unidad. Los procedimientos almacenados se guardan en una base de datos; se pueden ejecutar desde una aplicación y permiten variables declaradas por el usuario, ejecución condicional y otras funciones eficaces de programación. Los procedimientos almacenados pueden contener flujo de programas, lógica y consultas a la base de datos. Pueden aceptar parámetros, proporcionar resultados de parámetros, devolver conjuntos de resultados individuales o múltiples y devolver valores.

Las ventajas de utilizar procedimientos almacenados en SQL Server en vez de programas Transact-SQL almacenados localmente en equipos clientes consisten en que:  
• Permiten una programación modular.

Puede crear el procedimiento una vez, almacenarlo en la base de datos, y llamarlo desde el programa el número de veces que desee. Un especialista en programación de bases de datos puede crear procedimientos almacenados, que luego será posible modificar independientemente del código fuente del programa. Facilitan el mantenimiento.

• Permiten una ejecución más rápida.

En situaciones en las que se necesita una gran cantidad de código Transact-SQL, o si las operaciones se realizan varias veces, los procedimientos almacenados pueden ser más rápidos que los lotes de código Transact-SQL. Los procedimientos son analizados y optimizados en el momento de su creación, y es posible utilizar una versión del procedimiento que se encuentra en la memoria después de que se ejecute por primera vez. Las instrucciones de Transact-SQL que se envían varias veces desde el cliente cada vez que deben ejecutarse tienen que ser compiladas y optimizadas siempre que SQL Server las ejecuta.

• Pueden reducir el tráfico de red.

Una operación que necesite centenares de líneas de código Transact-SQL puede realizarse mediante una sola instrucción que ejecute el código en un procedimiento, en vez de enviar cientos de líneas de código por la red.

• Pueden utilizarse como mecanismo de seguridad.

Es posible conceder permisos a los usuarios para ejecutar un procedimiento almacenado, incluso si no cuentan con permiso para ejecutar directamente las instrucciones del procedimiento.

Funciones definidas por el usuario.

Las funciones son un objeto que combina algunas capacidades de las vistas, con otras de los procedimientos. Como las vistas, pueden extraer datos y ejecutar cálculos, y devuelven un resultado al usuario o al programa que les ejecuto. Tanto como los procedimientos, incluyen códigos de TSQL, y pueden ser ejecutados con parámetros.

Las funciones devuelven un valor o un conjunto de valores.

Las funciones definidas por el usuario se crean con la instrucción CREATE FUNCTION, se modifican con la instrucción ALTER FUNCTION y se quitan con la instrucción DROP FUNCTION. Todos los nombres de funciones completos (database\_name.owner\_name.function\_name) definidos por el usuario deben ser únicos. Para crear, modificar o quitar funciones definidas por el usuario, debe tener permisos de CREATE FUNCTION. Los usuarios distintos del propietario deben tener permiso EXECUTE para una función, y solo así podrán utilizarla en una instrucción de Transact-SQL. Para crear o modificar tablas con referencias a funciones definidas por el usuario en la restricción CHECK, la cláusula DEFAULT o la definición de una columna calculada, también debe tener permiso REFERENCES para las funciones. Los errores de Transact-SQL que producen la cancelación de una instrucción y continúan con la siguiente instrucción del módulo, como desencadenadores o procedimientos almacenados, se tratan de forma distinta dentro de una función. En las funciones, estos errores hacen que se detenga la ejecución de la función. Esto hace que se cancele la función que invocó la instrucción. Una función definida por el usuario no tiene ninguno o tiene varios parámetros de entrada y devuelve un valor escalar o una tabla. Una función puede tener un máximo de 1024 parámetros de entrada. Cuando un parámetro de la función toma un valor predeterminado, debe especificarse la palabra clave DEFAULT al llamar a la función para poder obtener el valor predeterminado. Este comportamiento es diferente del de los parámetros con valores predeterminados de los procedimientos almacenados, para los cuales omitir el parámetro implica especificar el valor predeterminado. Las funciones definidas por el usuario no admiten parámetros de salida.

Consultas Distribuidas.

Las consultas distribuidas tienen acceso a datos de varios orígenes, que pueden estar almacenados en un equipo o en equipos distintos. Microsoft SQL Server 2000 admite las consultas distribuidas a través de OLE DB Las consultas distribuidas proporcionan a los usuarios de SQL Server acceso a:

• Datos distribuidos almacenados en múltiples instancias SQL Server.

• Datos heterogéneos almacenados en varios orígenes de datos relacionales y no relacionales a los que se tiene acceso mediante un proveedor OLE DB.

Los proveedores OLE DB exponen datos en objetos tabulares llamados conjuntos de filas. En las instrucciones Transact-SQL, SQL Server 2000 permite que se haga referencia a los conjuntos de filas de los proveedores OLE DB como si fueran una tabla de SQL Server. En las instrucciones SELECT, INSERT, UPDATE y DELETE de Transact-SQL, se puede hacer referencia directa a las tablas y vistas de orígenes de datos externos. Puesto que las consultas distribuidas usan OLE DB como interfaz subyacente, éstas tienen acceso a los sistemas DBMS relacionales tradicionales con procesadores de consultas SQL, así como a los datos administrados por orígenes de datos de capacidad y sofisticación diversas. Siempre que el software propietario de los datos los expone en un conjunto de filas tabular a través del proveedor OLE DB, los datos se podrán usar en las consultas distribuidas. Nota: El uso de las consultas distribuidas en SQL Server es similar a la funcionalidad de las tablas vinculadas mediante ODBC, que anteriormente admitía Microsoft Access. Esta funcionalidad se encuentra ahora integrada en SQL Server con OLE DB como interfaz para los datos externos.

[Transacciones](https://es.wikipedia.org/wiki/Transacci%C3%B3n_(inform%C3%A1tica)).

Una transacción es un conjunto de comandos, que se está ejecutado completamente o no ejecutado en absoluto: todo o nada. Por ejemplo, si una suma de dinero fue trasladada de una cuenta bancaria a otra, y hay que actualizar ambas cuentas sobre el depósito y la retirada; es obligatorio que ambas cuentas se actualizan juntas, o ninguna (en caso que una de las actualizaciones falla); para evitar consecuencias inconsistentes de un depósito sin ninguna retirada, o vice versa. Por lo tanto, una transacción es una secuencia de operaciones realizadas como una sola unidad lógica de trabajo. Una unidad lógica de trabajo debe exhibir cuatro propiedades, conocidas como propiedades ACID (atomicidad, coherencia, aislamiento y durabilidad), para ser calificada como transacción:

* Atomicidad.

Una transacción debe ser una unidad atómica de trabajo, tanto si se realizan todas sus modificaciones en los datos, como si no se realiza ninguna de ellas.

* Coherencia.

Cuando finaliza, una transacción debe dejar todos los datos en un estado coherente. En una base de datos relacional, se deben aplicar todas las reglas a las modificaciones de la transacción para mantener la integridad de todos los datos. Todas las estructuras internas de datos, como índices de árbol B o listas doblemente vinculadas, deben estar correctas al final de la transacción.

* Aislamiento.

Las modificaciones realizadas por transacciones simultáneas se deben aislar de las modificaciones llevadas a cabo por otras transacciones simultáneas. Una transacción ve los datos en el estado en que estaban antes de que otra transacción simultánea los modificara o después de que la segunda transacción se haya concluido, pero no ve un estado intermedio. Esto se conoce como seriabilidad debido a que su resultado es la capacidad de volver a cargar los datos iniciales y reproducir una serie de transacciones para finalizar con los datos en el mismo estado en que estaban después de realizar las transacciones originales.

* Durabilidad.

Una vez concluida una transacción, sus efectos son permanentes en el sistema. Las modificaciones persisten aún en el caso de producirse un error del sistema.

El optimizador.

El optimizador es una parte del software que "toma la decisión" de cómo cada comando se ejecutará, tanto que la ejecución será lo más eficiente, o por lo menos bastante eficiente (es decir, bastante eficiente para evitar seguir buscando otra solución, que aún que sea más eficiente, el precio de la búsqueda adicional "costará" más que el ahorro de recursos).

[SQL](https://es.wikipedia.org/wiki/SQL) es un [lenguaje declarativo](https://es.wikipedia.org/wiki/Programaci%C3%B3n_declarativa), en el cual el desarrollador declara que quiere extraer o actualizar sin la necesidad de indicar cómo (a contrario de los [lenguajes imperativos](https://es.wikipedia.org/wiki/Programaci%C3%B3n_imperativa), y por lo tanto el optimizador juega un papel protagónico, que de acuerdo con las estadísticas que el sistema almacena sobre las distribuciones de los datos en las tablas, los indexes, y reglas internas; toma la decisión adecuada.

Privilegios y seguridad de datos.

Para conectarse al SQL Server, se necesita un Login (usuario a nivel del servidor). Cuando la política de seguridad se define como Windows Authentication y el servidor se combina con las definiciones del [Domain](https://es.wikipedia.org/wiki/Dominio_(redes_inform%C3%A1ticas)" \o "Dominio (redes informáticas)), los Logins se definen en el [Active Directory](https://es.wikipedia.org/wiki/Active_Directory). Cuando la definición es SQL Server Authentication los logins (usuario y contraseña) se definen en el SQL Server mismo. Consecuentemente, en el primer caso hay que identificarse con nombre y contraseña solamente al conectarse a la red, y luego se conecta automáticamente a todos los servidores que son Windows Authentication (con el Login global); y en el segundo caso hay que identificarse al conectarse a cada servidor de SQL Server Authentication (cada vez con un Login local).

A nivel de la base de datos, el usuario se identifica como un User que está relacionado generalmente al Login (que es a nivel del servidor), y los privilegios al User existen solamente en el ámbito de la base de datos (además a los privilegios al Login). Para otorgar derechos generales puede asistirse con listas de Server Roles (roles a nivel del servidor) o Database Roles (roles a nivel de la base de datos específica), cada cual con privilegios específicos a un rol específico; y cada usuario asociado con uno de estos Roles obtiene los privilegios asociados con él. Además, el administrador puede otorgar derechos específicos, y crear otros Database Roles (no se puede crear Server Roles).

Los privilegios a nivel del servidor incluyen la capacidad de crear bases de datos, utilizar las tareas (Jobs), crear respaldos de bases de datos y restaurarlos, modificar las definiciones del servidor, etc. Los privilegios a nivel de la base de datos posibilitan extraer y actualizar datos, crear objetos como procedimientos y tablas, utilizar dichos objetos, etc. Como regla general se puede otorgar derechos (Grant), revocar privilegios existentes (Revoke), y denegar privilegios aún no existen (Deny).

Otras herramientas de servicio.

Adicionalmente a sus capacidades elementarías como herramienta de gestión de bases de datos relacionales (crear tablas, definir las relaciones entre ellas, gestión de transacciones, crear índices etc, SQL Server apoya una lista que aumenta de otras herramientas de servicio; aparte de herramientas de terceros que cambian o que completan lo que existe.

Respaldos y recuperaciones.

Aparte de soluciones de alternativas a nivel del sistema operativo (respaldo de los archivos de la base de datos), hay una herramienta integrada en el SQL Server que posibilita un respaldo completo o diferencial, de acuerdo con el modelo de recuperación (Recovery Model) predefinido a la base de datos; y una recuperación completa o a un punto de tiempo. Aparte de un respaldo de la base de datos se puede respaldarlos a través de un guion (con o sin los datos). A partir de 2008, se puede comprimir los archivos de respaldo.

Compresión.

A partir de 2008 se añadió la opción de compresión que posibilita comprimir el tamaño físico de las tablas y los índices, y utilizar más eficientemente el volumen de los discos rígidos y reducir operaciones de IO (lo que aumenta la carga del CPU). Como ya ha sido mencionado, se puede comprimir también los archivos de respaldo.

Replicación, alta disponibilidad, y recuperación de desastres.

Algunas herramientas posibilitan crear replicas parciales o completos de las bases de datos, mejorar la disponibilidad, y recuperar de desastres; aparte de la opción de respaldar y recuperar las bases de datos, una opción que se supone que es muy lenta.

Las copias creadas por estas herramientas, pueden ser utilizadas como un respaldo disponible inmediatamente en caso de que el sistema falla, tanto como una réplica en un subsistema, independiente del sistema de origen, para equilibrio de carga; y que las actualizaciones se ejecuten en la base de datos, y las recuperaciones para los reportajes se ejecuten en la réplica.

Entre dichas herramientas se puede mencionar la [replicación](https://es.wikipedia.org/wiki/Replicaci%C3%B3n_(inform%C3%A1tica)) que posibilita crear una réplica sincronizada de la base de datos, el [Mirroring](https://es.wikipedia.org/wiki/Espejo_(Internet)" \o "Espejo (Internet)) que ejecuta en el servidor de espejo cada actualización que se ejecuta en el servidor de origen, Log Shipping que posibilita almacenar una copia sincronizada a través de archivos de registro (Log) con todas las actualizaciones en el servidor de origen, y las herramientas de alta disponibilidad y recuperación de desastres (HADR, desde 2012) que solucionan problemas de Mirroring en relación con el tiempo de reacción de los fallos técnicos y la disponibilidad de las copias en tiempos de paz.

El agente y la programación de tareas.

El agente es el servicio encargado de la programación de tareas, y se encarga de ejecutarlas independientemente. Generalmente el ejecuta tareas de mantenimiento, tareas complejas de [ETL](https://es.wikipedia.org/wiki/Extract,_transform_and_load), [respaldos](https://es.wikipedia.org/wiki/Copia_de_seguridad), etc.

Mantenimiento.

Con el fin de mejorar el rendimiento del sistema hay que mantener las estadísticas, utilizadas por el optimizador, organizar los archivos físicos, etc; y se utilizan herramientas dedicadas para estos propósitos, que se ejecutan periódicamente por tareas programadas, y de una manera coordinada con las tareas de ETL y de respaldo.

Service Broker.

Una tecnología que fue agregada en 2005 que implementa [arquitectura orientada a servicios](https://es.wikipedia.org/wiki/Arquitectura_orientada_a_servicios), y que posibilita ejecuciones asíncronos: primero que nada para enviar mensajes entre distintas aplicaciones que se ejecutan simultáneamente, pero también para ejecutar procedimientos asíncronamente, en la manera de [dispara y olvida](https://es.wikipedia.org/wiki/Dispara_y_olvida" \o "Dispara y olvida) - un procedimiento que se ejecuta en una sesión diferente de la sesión que lo inicio, y ambos procedimientos siguen ejecutado independientemente uno del otro.

Enviar Correos Electrónicos.

SQL Server tiene la capacidad de enviar [correos electrónicos](https://es.wikipedia.org/wiki/Correo_electr%C3%B3nico) a través del código. Esta capacidad se utiliza generalmente para enviar alertas sobre problemas en el sistema (por ejemplo, si el CPU aumento a un umbral predefinido o si el proceso [ETL](https://es.wikipedia.org/wiki/ETL) falló), pero también cuando el proceso termino exitosamente.

Full Text Search (Búsqueda de Textos completos)[[editar](https://es.wikipedia.org/w/index.php?title=Microsoft_SQL_Server&action=edit&section=33" \o "Editar sección: Full Text Search (Búsqueda de Textos completos))]

Una herramienta que posibilita indexar columnas textuales como textos y no solo como cadenas; y ejecutar búsquedas complejas dependientes en el sentido del texto y en el idioma. Por ejemplo, buscamos un verbo, y queremos obtener todas las ocurrencias de sus conjugaciones.

Rastrear.

Estas herramientas incluyen el Trace que posibilita rastrear actividades con el fin de mantener cargas y fallos, y seguridad de datos (recuperación no permitida de datos), el Profiler que posibilita rastrear los commandos que se ejecutan y los eventos que se ocurren en el servidor, y el Extended Events que fue agregado en 2008 y cambia el profiler gracias a su baja signatura (la consumición de recursos y la influencia sobre el rendimiento del servidor).

Aparte de estos, se puede utilizar 2 tipos de Trigers (disparadores) para rastrear los cambios y las actividades: DML Trigers pre definidos sobre las tablas y las vistas y que se inician por instrucciones de actualización de datos (Select / Update / Delete), y DDL Triggers que se inician por cambios en los objetos mismos (y no en los datos), en el nivel de la base de datos o del servidor.

Combinación de CLR.

A partir de 2005 se puede combinar fácilmente en SQL Server procedimientos, funciones, y funciones de agregado desarrolladas en [CLR](https://es.wikipedia.org/wiki/CLR). Hay que desarrollar el código en una de las herramientas de desarrollo de [.NET](https://es.wikipedia.org/wiki/.NET), crear un archivo [DLL](https://es.wikipedia.org/wiki/DLL), y combinarlo en el sistema. La ventaja de esta tecnología es sus capacidades en problemas que no son exclusivamente de bases de datos (manipulación de datos), e incluyen cálculos complejos o manipulaciones textuales de cadenas.

Herramientas de [Inteligencia empresarial](https://es.wikipedia.org/wiki/Inteligencia_empresarial).

Una instalación típica incluye también las herramientas de BI (Inteligencia empresarial):

[SSIS](https://es.wikipedia.org/wiki/SQL_Server_Integration_Services) (SQL Server Integration Services)[[editar](https://es.wikipedia.org/w/index.php?title=Microsoft_SQL_Server&action=edit&section=37" \o "Editar sección: SSIS (SQL Server Integration Services))]

Una herramienta de [ETL](https://es.wikipedia.org/wiki/Extract,_transform_and_load) que posibilita la extracción de datos de distintos orígenes (no solo SQL Server), la transformación de dichos datos, y la carga (generalmente pero no obligatoriamente a [almacén de datos](https://es.wikipedia.org/wiki/Almac%C3%A9n_de_datos)).

[SSAS](https://es.wikipedia.org/w/index.php?title=SQL_Server_Analysis_Services&action=edit&redlink=1) (SQL Server Analysis Services).

Una herramienta para crear [Bases de Datos Multidimensionales](https://es.wikipedia.org/w/index.php?title=Bases_de_Datos_Multidimensionales&action=edit&redlink=1) (no relacionales), que se puede explorar mediante extracciones de datos en distintos niveles de agrupación, profundización (Drill Down) de una suma a sus detalles, y utilización de [MDX](https://es.wikipedia.org/wiki/Expresiones_multidimensionales) (un lenguaje parecido a SQL, adaptado a bases de datos multidimensionales).

[SSRS](https://es.wikipedia.org/w/index.php?title=SQL_Server_Reporting_Services&action=edit&redlink=1) (SQL Server Reporting Services).

Una herramienta para crear y dar formato a informes, otorgar derechos de contemplación en ellos, y su distribución. Se puede contemplarlos con un [Navegador web](https://es.wikipedia.org/wiki/Navegador_web), y se puede exportarlos a archivos de [Excel](https://es.wikipedia.org/wiki/Microsoft_Excel), [PDF](https://es.wikipedia.org/wiki/PDF), etc. los datos se extraen generalmente del almacén de datos o del OLAP.

Desventajas.

En versiones de 32 bits, SQL Server usa Address Windowing Extension (AWE) para hacer el direccionamiento por encima de 4 GB. Esto le impide usar la administración dinámica de memoria, y sólo le permite alojar un máximo de 64 GB de memoria compartida. Esta limitación es exclusiva de sistemas operativos 32 bits; en sistemas operativos 64 bits, la memoria máxima que se puede direccionar en Edición Estándar es 64 Gb y en Edición Enterprise 4Tb

**Software de Programación.**

**Visual Studio .NET 2003**

Visual Studio .NET es un conjunto completo de herramientas de desarrollo para la construcción de aplicaciones Web ASP, servicios Web XML, aplicaciones para escritorio y aplicaciones móviles. Visual Basic .NET, Visual C++ .NET, Visual C# .NET y Visual J# .NET utilizan el mismo entorno de desarrollo integrado (IDE), que les permite compartir herramientas y facilita la creación de soluciones en varios lenguajes. Asimismo, dichos lenguajes aprovechan las funciones de .NET Framework, que ofrece acceso a tecnologías clave para simplificar el desarrollo de aplicaciones Web ASP y servicios Web XML.

Visual J#.

Visual J# es una herramienta de desarrollo que pueden utilizar los programadores que estén familiarizados con la sintaxis de Java para generar aplicaciones y servicios en .NET Framework. Integra la sintaxis de Java en el entorno de desarrollo integrado (IDE) de Visual Studio .NET. Visual J# admite también la mayor parte de la funcionalidad de Visual J++ 6.0, incluidas las extensiones de Microsoft. Visual J# no es una herramienta para el desarrollo de aplicaciones que se ejecuten en Java Virtual Machine. Las aplicaciones y los servicios generados con Visual J# se ejecutarán solamente en .NET Framework. Microsoft ha desarrollado Visual J# de manera independiente. No está refrendado ni aprobado por Sun Microsystems, Inc. Para obtener más información, vea [Introducción a Visual J#](https://msdn.microsoft.com/es-mx/library/cc434262(v=vs.71).aspx).

Aplicaciones para dispositivos inteligentes.

El entorno de desarrollo integrado de Visual Studio .NET incluye ahora herramientas para desarrollar aplicaciones para dispositivos inteligentes, como Pocket PC. Mediante las herramientas y .NET Compact Framework, un subconjunto de .NET Framework, puede crear, generar, depurar e implementar aplicaciones que utilizan .NET Compact Framework para ejecutarse en asistentes digitales personales (PDA), teléfonos móviles y otros dispositivos de recursos restringidos. Para obtener más información, vea [Proyectos de dispositivos inteligentes](https://msdn.microsoft.com/es-mx/library/cc433708(v=vs.71).aspx).

Diseñador de ASP.NET Mobile.

El Diseñador de ASP.NET Mobile complementa a ASP.NET y a .NET Framework, permitiendo generar aplicaciones Web para teléfonos móviles, PDA y localizadores. Este diseñador está integrado en el IDE de Visual Studio. Puede crear aplicaciones Web móviles, utilizar el diseñador móvil para modificar un formulario Web móvil y, a continuación, generar y ejecutar la aplicación, todo desde Visual Studio. Para obtener más información, vea [Introducción al uso del Diseñador de ASP.NET Mobile](https://msdn.microsoft.com/es-mx/library/cc365194.aspx).

Formularios Web Forms.

Los formularios Web Forms son una tecnología ASP.NET que se utiliza para crear páginas Web programables. Los formularios Web Forms se representan como código HTML y secuencias de comandos compatibles con exploradores, lo que permite ver las páginas en cualquier explorador y plataforma. Mediante el uso de formularios Web Forms se pueden crear páginas Web arrastrando y colocando controles en el diseñador y agregando código posteriormente, de forma parecida a la creación de formularios en Visual Basic. Para obtener más información, vea [Introducción a las páginas de formularios Web Forms](https://msdn.microsoft.com/es-mx/library/cc437550(v=vs.71).aspx).

Formularios Windows Forms.

Los formularios Windows Forms son la nueva plataforma de desarrollo de aplicaciones para Microsoft Windows, basados en .NET Framework. Este marco de trabajo proporciona un conjunto de clases claro, orientado a objetos y ampliable, que permite desarrollar complejas aplicaciones para Windows. Además, los formularios Windows Forms pueden actuar como interfaz de usuario local en una solución distribuida de varios niveles. Para obtener más información, vea [Introducción a los formularios Windows Forms](https://msdn.microsoft.com/es-mx/library/aa983655(v=vs.71).aspx).

Servicios Web XML.

Los servicios Web XML son aplicaciones que pueden recibir solicitudes y datos mediante el uso de XML sobre HTTP. Los servicios Web XML no están asociados a una tecnología de componentes específica o a una convención de llamada a objetos concreta, por lo que cualquier lenguaje, modelo de componente o sistema operativo puede tener acceso a ellos. En Visual Studio .NET se pueden crear e incluir con rapidez servicios Web XML mediante Visual Basic, Visual C#, JScript, Extensiones administradas para C++ o servidor ATL. Para más información, vea Programar el Web con servicios Web XML.

Compatibilidad con XML.

El Lenguaje de marcado extensible (XML) proporciona un método para describir datos estructurados. XML es un subconjunto de SGML optimizado para la entrega a través de Web. El Consorcio World Wide Web (W3C) define los estándares de XML para que los datos estructurados sean uniformes e independientes de las aplicaciones. Visual Studio .NET admite totalmente XML e incluye el Diseñador XML para facilitar la edición de XML y la creación de esquemas XML. Para más información, vea [Datos y esquemas XML](https://msdn.microsoft.com/es-mx/library/cc438225(v=vs.71).aspx) y [Diseñador XML](https://msdn.microsoft.com/es-mx/library/aa291459(v=vs.71).aspx).

El entorno .NET Framework.

.NET Framework es un entorno multilenguaje que permite generar, implantar y ejecutar Servicios Web y aplicaciones XML. Consta de tres partes principales:

* Common Language Runtime   A pesar de su nombre, el motor de tiempo de ejecución desempeña una función tanto durante la ejecución como durante el desarrollo de los componentes. Cuando el componente se está ejecutando, el motor de tiempo de ejecución es responsable de administrar la asignación de memoria, iniciar y detener subprocesos y procesos, y hacer cumplir la directiva de seguridad, así como satisfacer las posibles dependencias del componente sobre otros componentes. Durante el desarrollo, el papel del motor de tiempo de ejecución cambia ligeramente; a causa de la gran automatización que permite (por ejemplo, en la administración de memoria), el motor simplifica el trabajo del programador, especialmente al compararlo con la situación actual de la tecnología COM. En concreto, funciones tales como la reflexión reducen de forma espectacular la cantidad de código que debe escribir el programador para convertir la lógica de empresa en componentes reutilizables.
* Clases de programación unificadas   El entorno de trabajo ofrece a los programadores un conjunto unificado, orientado a objetos, jerárquico y extensible de bibliotecas de clases (API). Actualmente, los programadores de C++ utilizan las Microsoft Foundation Classes y los programadores de Java utilizan las Windows Foundation Classes. El entorno de trabajo unifica estos modelos dispares, ofreciendo a los programadores de Visual Basic y JScript la posibilidad de tener también acceso a las bibliotecas. Con la creación de un conjunto de API comunes para todos los lenguajes de programación, Common Language Runtime permite la herencia, el control de errores y la depuración entre lenguajes. Todos los lenguajes de programación, desde JScript a C++, pueden tener acceso al entorno de trabajo de forma parecida y los programadores pueden elegir libremente el lenguaje que desean utilizar.
* ASP.NET   ASP.NET está construida sobre las clases de programación del entorno .NET Framework, y proporciona un modelo para aplicaciones Web con un conjunto de controles y una infraestructura que simplifican la creación de aplicaciones Web ASP. ASP.NET incluye un conjunto de controles que encapsulan los elementos comunes de la interfaz de usuario de HTML, como cuadros de texto y menús desplegables. Sin embargo, dichos controles se ejecutan en el servidor Web, y envían al explorador Web su interfaz de usuario en forma de HTML. En el servidor, los controles exponen un modelo de programación orientado a objetos que ofrece al programador de Web la riqueza de la programación orientada a objetos. ASP.NET proporciona también servicios de infraestructura, como administración de estado de sesión y reciclado de procesos, que reducen aún más la cantidad de código que debe escribir el programador, e incrementan la fiabilidad de las aplicaciones. Asimismo, ASP.NET utiliza estos mismos conceptos para permitir a los programadores la entrega de software en forma de servicio. Mediante las funciones de servicios Web XML, los programadores de ASP.NET pueden escribir su lógica de empresa y utilizar la infraestructura de ASP.NET para entregar el servicio a través de SOAP. Para obtener más información, vea Soap Community Links.

**Visual Basic .NET**.

Visual Basic .NET (VB.NET) es un [lenguaje de programación](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n) [orientado a objetos](https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos) que se puede considerar una evolución de [Visual Basic](https://es.wikipedia.org/wiki/Visual_Basic) implementada sobre el [framework .NET](https://es.wikipedia.org/wiki/Microsoft_.NET" \o "Microsoft .NET). Su introducción resultó muy controvertida, ya que debido a cambios significativos en el lenguaje VB.NET no es retro compatible con Visual Basic, pero el manejo de las instrucciones es similar a versiones anteriores de Visual Basic, facilitando así el desarrollo de aplicaciones más avanzadas con herramientas modernas. Para mantener eficacia en el desarrollo de las aplicaciones. La gran mayoría de programadores de VB.NET utilizan el [entorno de desarrollo integrado](https://es.wikipedia.org/wiki/Entorno_de_desarrollo_integrado) [Microsoft Visual Studio](https://es.wikipedia.org/wiki/Microsoft_Visual_Studio) en alguna de sus versiones (desde el primer Visual Studio .NET hasta Visual Studio .NET 2015, que es la última versión de Visual Studio para la plataforma .NET).

Al igual que con todos los lenguajes de programación basados en .NET, los programas escritos en VB .NET requieren el Framework .NET o Mono para ejecutarse.

Versiones.

Visual Basic 2008 (VB 9.0).

Para esta versión se añadieron varias novedades, incluyendo:

* Soporte para [LINQ](https://es.wikipedia.org/wiki/Language_Integrated_Query)
* [Expresiones lambda](https://es.wikipedia.org/wiki/C%C3%A1lculo_lambda)
* Literales [XML](https://es.wikipedia.org/wiki/Extensible_Markup_Language)

Visual Basic 2010 ofrece soporte para entorno de ejecución dinámica.

VB 2010 forma parte de [Microsoft Silverlight](https://es.wikipedia.org/wiki/Microsoft_Silverlight). Visual Basic es un lenguaje de programación que permite realizar diferentes programas de registro anidados.

Relación con Visual Basic.

Si Visual Basic .NET debe considerarse una mera versión de [Visual Basic](https://es.wikipedia.org/wiki/Visual_Basic), o si debe considerarse como un nuevo lenguaje de programación es un tema que ha traído mucha discusión, y que aún la trae.

La sintaxis básica es prácticamente la misma entre VB y VB.NET, con la excepción de los añadidos para soportar nuevas características como el control estructurado de excepciones, la programación orientada a objetos, o los Genéricos.

Las diferencias entre VB y VB.NET son profundas, sobre todo en cuanto a metodología de programación y bibliotecas, pero ambos lenguajes siguen manteniendo un gran parecido, cosa que facilita notablemente el paso de VB a VB.NET.

Versiones del entorno de programación para Visual Basic.

Al mismo tiempo que evolucionaba el lenguaje, las herramientas que Microsoft proporciona para programar también evolucionaban para adaptarse a las novedades.

Visual Studio .NET (2002).

Visual Studio .NET se publicó en 2002 y fue la primera versión de Visual Studio en introducir el framework .NET. Esta versión de Visual Studio introdujo, junto con el Framework .NET tres nuevos lenguajes de programación, Visual C#, VB.NET y Visual J#.

En esta primera versión de Visual Studio .NET se podían programar aplicaciones Windows.Forms (aplicaciones de escritorio) y aplicaciones [ASP.NET](https://es.wikipedia.org/wiki/ASP.NET) (Aplicaciones Web).

Visual Studio .NET 2003.

Visual Studio .NET 2003 se publicó en 2003 fue una actualización menor de Visual Studio .NET, básicamente propiciada por la introducción de la versión 1.1 del Framework .NET.

En esta versión se añadió por primera vez la posibilidad de programar para dispositivos móviles usando .NET, ya fuera usando el Compact Framework, o ASP.NET.

Añade soporte de 64-bit (x86-64: AMD64 e Intel 64, e IA-64: Itanium)

Ediciones: Express, Standard, ready bost, Professional, Tools for Office, y 5 ediciones Visual Studio Team System (Architects, Software Developers, Testers, y Database Professionals)

La versión interna de Visual Studio 2005 es la 8.0, mientras que el formato del archivo es la 9.0.

Visual Basic Express Edition.

A partir de la introducción en el mercado de la versión 2005 de Visual Studio Microsoft publicó lo que se conoce como ediciones Exp de distintos programas. Las versiones Express son versiones limitadas pero gratuitas, pensadas para usos no profesionales (principiantes, aficionados y pequeños negocios), existiendo una edición independiente para cada lenguaje.

Visual Basic Express Edition es una versión de Visual Studio limitada. Esta versión permite sólo programar en VB.NET, y además limita el tipo de proyectos que se pueden desarrollar. Visual Web Developer Express Edition permite programar páginas ASP.NET en VB.

Se lanzó el Service Pack 1 para Visual Studio 2005 el 14 de diciembre de 2006.

Visual Studio 2008.

El IDE de Visual Studio 2008 permite trabajar contra 3 .NET frameworks diferentes:

* .NET Framework 2.0
* .NET Framework 3.0
* .NET Framework 3.5

También es muy fácil de usar gracias al desarrollo de hardware.

Además, integra el framework ASP.NET AJAX para el desarrollo de AJAX..

Visual Studio 2010.

El 12 de Abril del 2010, Microsoft publica Visual Studio 2010, nombre clave Dev10, y .NET Framework versión 4. Compatible con Visual Basic.net, con una interfaz rediseñada, más sencilla y con soporte para diseño de aplicaciones en Windows 7.

Entornos de desarrollo (IDE) alternativos a Visual Studio.

Para desarrollar en VB.NET existen algunas alternativas a Visual Studio, quizás la más notable sea SharpDevelop.

SharpDevelop.

[SharpDevelop](https://es.wikipedia.org/wiki/SharpDevelop) es un entorno de programación integrado que permite programar en C# y en VB.NET.

Este es un entorno publicado bajo licencia [LGPL](https://es.wikipedia.org/wiki/Licencia_p%C3%BAblica_general_limitada_de_GNU), lo que implica que es libre y que disponemos del código fuente.

MonoDevelop.

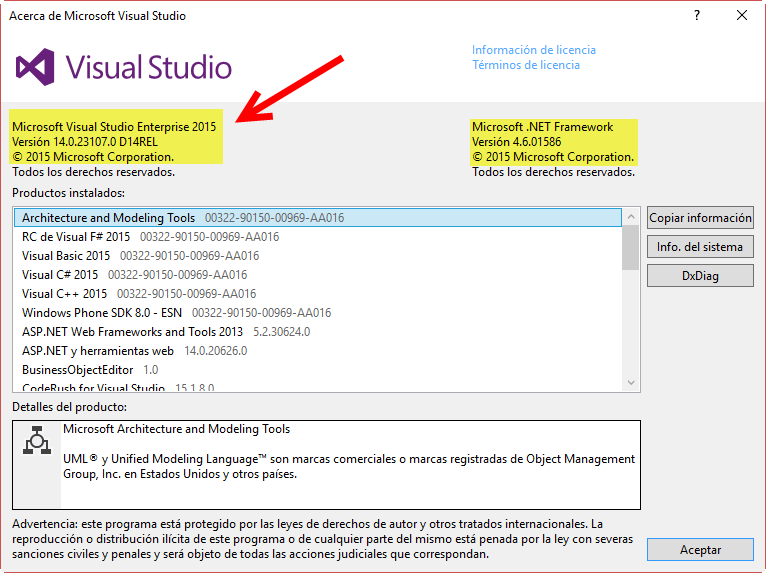
[MonoDevelop](https://es.wikipedia.org/wiki/MonoDevelop) es una implementación de SharpDevelop para programar usando [Mono](https://es.wikipedia.org/wiki/Proyecto_Mono), una implementación libre de .NET que funciona en distintos sistemas operativos.

Desarrollo multiplataforma.

Gracias al esfuerzo que ha realizado la gente del proyecto [Mono](https://es.wikipedia.org/wiki/Proyecto_Mono) para implementar una versión compatible 100% con .NET que incluye la inmensa mayoría de las bibliotecas y una implementación multiplataforma de Windows.Forms, resulta posible programar para distintos sistemas operativos usando VB.NET.

Con relación a SofTec.

La versión de Visual Basic .NET que se está utilizando para desarrollar en el Instituto Tecnológico de Tepic es Microsoft Visual Studio Enterprise 2015 Versión 14.0.23107.0 D14REL.



**ASP.NET.**

ASP.NET es un modelo de desarrollo Web unificado que incluye los servicios necesarios para crear aplicaciones Web empresariales con el código mínimo. ASP.NET forma parte de .NET Framework y al codificar las aplicaciones ASP.NET tiene acceso a las clases en .NET Framework. El código de las aplicaciones puede escribirse en cualquier lenguaje compatible con el Common Language Runtime (CLR), entre ellos Microsoft Visual Basic, C#, JScript .NET y J#. Estos lenguajes permiten desarrollar aplicaciones ASP.NET que se benefician del Common Language Runtime, seguridad de tipos, herencia, etc. ASP.NET es un [framework para aplicaciones web](https://es.wikipedia.org/wiki/Framework_para_aplicaciones_web" \o "Framework para aplicaciones web) desarrollado y comercializado por [Microsoft](https://es.wikipedia.org/wiki/Microsoft). Es usado por programadores y diseñadores para construir [sitios web](https://es.wikipedia.org/wiki/Sitio_web) dinámicos, [aplicaciones web](https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web) y [servicios web](https://es.wikipedia.org/wiki/Servicio_Web) [XML](https://es.wikipedia.org/wiki/XML). Apareció en enero de 2002 con la versión 1.0 del [.NET Framework](https://es.wikipedia.org/wiki/.NET_Framework), y es la tecnología sucesora de la tecnología [Active Server Pages](https://es.wikipedia.org/wiki/Active_Server_Pages) (ASP). ASP.NET está construido sobre el [Common Language Runtime](https://es.wikipedia.org/wiki/Common_Language_Runtime" \o "Common Language Runtime), permitiendo a los programadores escribir código ASP.NET usando cualquier lenguaje admitido por el [.NET Framework](https://es.wikipedia.org/wiki/.NET_Framework).

ASP.NET incluye:

* Marco de trabajo de página y controles
* Compilador de ASP.NET
* Infraestructura de seguridad
* Funciones de administración de estado
* Configuración de la aplicación
* Supervisión de estado y características de rendimiento
* Capacidad de depuración
* Marco de trabajo de servicios Web XML
* Entorno de host extensible y administración del ciclo de vida de las aplicaciones
* Entorno de diseñador extensible

Marco de trabajo de páginas y controles.

El marco de trabajo de páginas y controles ASP.NET es un marco de trabajo de programación que se ejecuta en un servidor Web para generar y representar de forma dinámica páginas Web ASP.NET. Las páginas Web ASP.NET se pueden solicitar a cualquier explorador o dispositivo del cliente y ASP.NET representa el marcado (como HTML) al explorador que realizó la solicitud. Como norma, puede utilizar la misma página para varios exploradores, porque ASP.NET representa el marcado adecuado para el explorador que realiza la solicitud. Sin embargo, puede diseñar una página Web ASP.NET para ejecutarse en un explorador determinado, como Microsoft Internet Explorer 6, y aprovechar así todas las características de ese explorador. ASP.NET es compatible con los controles móviles de los dispositivos preparados para trabajar en Web como teléfonos celulares, PC portátiles y asistentes digitales personales (PDA).

Las páginas Web ASP.NET están completamente orientadas a objetos. En las páginas Web ASP.NET se puede trabajar con elementos HTML que usen propiedades, métodos y eventos. El marco de trabajo de páginas ASP.NET quita los detalles de implementación relacionados con la separación de cliente y servidor inherente a las aplicaciones Web presentando un modelo unificado que responde a los eventos de los clientes en el código que se ejecuta en el servidor. El marco de trabajo también mantiene automáticamente el estado de la página y de los controles que contenga durante el ciclo vital de procesamiento de la página. Para obtener más información, vea [Información general sobre páginas Web ASP.NET](https://msdn.microsoft.com/es-es/library/428509ah(v=vs.100).aspx).

El marco de trabajo de páginas y controles ASP.NET también permite encapsular la funcionalidad común de la interfaz de usuario en controles fáciles de usar y reutilizables. Los controles se escriben una vez, se pueden utilizar en varias páginas y se integran en la página Web ASP.NET en la que se colocan durante la representación.

El marco de trabajo de páginas y controles ASP.NET también proporciona funciones para controlar la apariencia y el funcionamiento general de los sitios Web a través de temas y máscaras. Se pueden definir temas y máscaras y, a continuación, aplicarlos en las páginas o controles. Para obtener más información, vea [Información general sobre temas y máscaras de ASP.NET](https://msdn.microsoft.com/es-es/library/ykzx33wh(v=vs.100).aspx).

Además de los temas, es posible definir páginas principales que se crean para conseguir un diseño coherente en las páginas de la aplicación. Una página principal única define el diseño y el comportamiento estándar deseados para todas las páginas (o un grupo de páginas) de la aplicación. A continuación, se pueden crear páginas de contenido individuales con el contenido específico de la página que se desee mostrar. Cuando los usuarios solicitan las páginas de contenido, las combinan con la página principal con el fin de generar un resultado que combine el diseño de la página principal con el de la página de contenido. Para obtener más información, vea [Información general sobre las páginas principales ASP.NET](https://msdn.microsoft.com/es-es/library/wtxbf3hh(v=vs.100).aspx).

Compilador de ASP.NET.

Compila todo el código de ASP.NET, lo que permite el establecimiento inflexible de tipos, las optimizaciones de rendimiento y el enlace en tiempo de compilación, entre otras ventajas. Una vez que se ha compilado el código, el Common Language Runtime compila una vez más código de ASP.NET en código nativo, lo que permite un mayor rendimiento.

ASP.NET incluye un compilador que compilará todos los componentes de la aplicación, incluidas las páginas y los controles, en un ensamblado que el entorno de host de ASP.NET puede utilizar a continuación para atender las solicitudes del usuario. Para obtener más información, vea [Información general sobre la compilación de ASP.NET](https://msdn.microsoft.com/es-es/library/ms178466(v=vs.100).aspx).

Infraestructura de seguridad.

Además de las características de seguridad de .NET, ASP.NET proporciona una infraestructura de seguridad avanzada para autenticar y autorizar el acceso de los usuarios y realizar otras tareas relacionadas con la seguridad. Puede autenticar usuarios con la autenticación de Windows suministrada por IIS o puede administrar la autenticación con su propia base de datos de usuario utilizando la autenticación mediante formularios ASP.NET y la suscripción ASP.NET. Además, puede administrar la autorización a las capacidades e información de su aplicación Web mediante los grupos de Windows o su propia base de datos de funciones personalizada utilizando las funciones de ASP.NET. Resulta fácil quitar, agregar o reemplazar estos esquemas dependiendo de las necesidades de la aplicación. Para obtener más información, vea los temas siguientes:

* [Proteger sitios web ASP.NET](https://msdn.microsoft.com/es-es/library/91f66yxt(v=vs.100).aspx)
* [Administrar usuarios mediante suscripciones](https://msdn.microsoft.com/es-es/library/tw292whz(v=vs.100).aspx)
* [Administrar autorizaciones con funciones](https://msdn.microsoft.com/es-es/library/9ab2fxh0(v=vs.100).aspx)
* [Proveedor de autenticación mediante formularios](https://msdn.microsoft.com/es-es/library/9wff0kyh(v=vs.100).aspx)

ASP.NET siempre se ejecuta con una identidad particular de Windows de modo que puede asegurar su aplicación utilizando las capacidades de Windows como, por ejemplo, las listas de control de acceso (ACL) de NTFS, permisos de la base de datos, etc. Para obtener más información sobre la identidad de ASP.NET, vea [Configurar la identidad de procesos en ASP.NET](https://msdn.microsoft.com/es-es/library/dwc1xthy(v=vs.100).aspx) y [Suplantación de ASP.NET](https://msdn.microsoft.com/es-es/library/xh507fc5(v=vs.100).aspx).

Funciones de administración de estado.

ASP.NET proporciona funcionalidad de administración de estado intrínseca que permite almacenar información entre las solicitudes de página, como la información de clientes o el contenido del carro de la compra. Puede guardar y administrar información específica de la aplicación, específica de la sesión, específica de la página, específica del usuario y definida por el desarrollador. Esta información puede ser independiente de cualquier control de la página,

ASP.NET ofrece funciones de estado distribuidas, lo que le permite administrar información de estado en múltiples instancias de la misma aplicación en un equipo o en varios. Para obtener más información, vea [Información general sobre la administración de estados de ASP.NET](https://msdn.microsoft.com/es-es/library/75x4ha6s(v=vs.100).aspx).

Configuración de ASP.NET.

Las aplicaciones ASP.NET utilizan un sistema de configuración que le permite definir valores de configuración para su servidor Web, para un sitio Web o para aplicaciones individuales. Puede crear valores de configuración cuando se implementan las aplicaciones ASP.NET y puede agregar o revisar los valores de configuración en cualquier momento con un impacto mínimo en aplicaciones y servidores Web de operaciones. Los valores de configuración de ASP.NET se almacenan en archivos basados en la tecnología XML. Dado que estos archivos XML son archivos de texto ASCII, es fácil realizar cambios de configuración a sus aplicaciones Web. Puede extender el esquema de configuración para satisfacer sus requisitos. Para obtener más información, vea [Información general sobre la configuración de ASP.NET](https://msdn.microsoft.com/es-es/library/ms178683(v=vs.100).aspx).

Supervisión de estado y características de rendimiento.

ASP.NET incluye características que le permiten supervisar el estado y el rendimiento de su aplicación ASP.NET. La supervisión del estado de ASP.NET permite proporcionar información sobre eventos clave que proporcionan información sobre el estado de una aplicación y sobre las condiciones de error. Estos eventos muestran una combinación de diagnósticos y características de supervisión, a la vez que proporcionan un elevado grado de flexibilidad en lo que respecta a lo que se registra y cómo. Para obtener más información, vea [Información general sobre la supervisión de estado en ASP.NET](https://msdn.microsoft.com/es-es/library/bb398933(v=vs.100).aspx).

ASP.NET admite dos grupos de contadores de rendimiento a los que pueden obtener acceso las aplicaciones:

* El grupo de contadores de rendimiento del sistema ASP.NET
* El grupo de contadores de rendimiento de la aplicación ASP.NET

Para obtener más información, vea [Supervisar el rendimiento de una aplicación ASP.NET](https://msdn.microsoft.com/es-es/library/3xxk09t8(v=vs.100).aspx).

Capacidad de depuración.

ASP.NET aprovecha la infraestructura de depuración en tiempo de ejecución para permitir la depuración entre lenguajes y equipos. Se pueden depurar tanto objetos administrados como no administrados, así como todos los lenguajes compatibles con el Common Language Runtime y los lenguajes de script. Para obtener información detallada, vea [Depuración en ASP.NET](https://msdn.microsoft.com/es-es/library/ms228091(v=vs.100).aspx).

Además, el marco de trabajo de páginas ASP.NET proporciona un modo de seguimiento que permite insertar mensajes de instrumentalización en las páginas Web ASP.NET. Para obtener más información, vea [Lo nuevo en seguimiento de ASP.NET](https://msdn.microsoft.com/es-es/library/h1ztfx8b(v=vs.100).aspx).

Marco de trabajo de servicios Web XML.

ASP.NET es compatible con los servicios Web XML. Un servicio Web XML es un componente que incluye funcionalidad de empresa que permite a las aplicaciones intercambiar información entre firewalls utilizando estándares como los servicios de mensajería HTTP y XML. Los servicios Web XML no están relacionados con ninguna tecnología de componentes ni con ninguna convención de llamada a objetos en concreto. Como resultado, pueden obtener acceso a los servicios Web XML los programas escritos en cualquier lenguaje, que usen cualquier modelo de componentes y se ejecuten en cualquier sistema operativo. Para obtener más información, vea [Servicios web XML con ASP.NET](https://msdn.microsoft.com/es-es/library/ba0z6a33(v=vs.100).aspx).

Entorno de host extensible y administración del ciclo de vida de las aplicaciones.

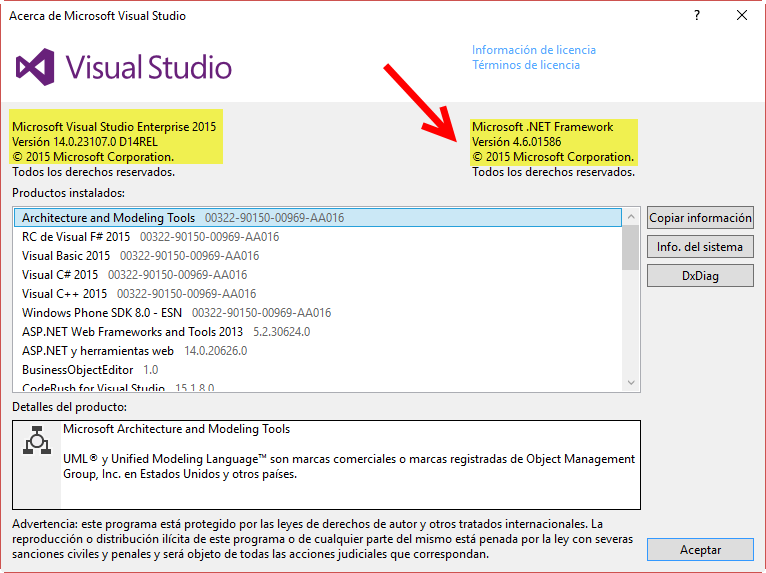
ASP.NET incluye un entorno de host extensible que controla el ciclo de vida de una aplicación desde el momento en que un usuario cualquiera tiene acceso a un recurso (como una página) en la aplicación hasta el momento en que se cierra la aplicación. Aunque ASP.NET se basa en un servidor Web (IIS) como un host de la aplicación, ASP.NET proporciona gran parte de la propia funcionalidad de host. La arquitectura de ASP.NET permite responder a los eventos de aplicación y crear controladores y módulos HTTP personalizados. Para obtener más información, vea [Información general sobre el ciclo de vida de una aplicación ASP.NET para IIS 5.0 y 6.0](https://msdn.microsoft.com/es-es/library/ms178473(v=vs.100).aspx).

Entorno de diseñador extensible.

ASP.NET incluye la compatibilidad mejorada para crear diseñadores de controles de servidor Web para utilizarlos con una herramienta de diseño visual como Visual Studio. Los diseñadores permiten crear una interfaz de usuario en tiempo de diseño para un control; de este modo, los desarrolladores pueden configurar las propiedades y el contenido del control en una herramienta de diseño visual. Para obtener más información, vea [Información general sobre los diseñadores de controles ASP.NET](https://msdn.microsoft.com/es-es/library/wxh45wzs(v=vs.100).aspx).

Con relación a SofTec.

La versión de ASP.NET que se está utilizando para desarrollar en el Instituto Tecnológico de Tepic es Microsoft .NET Framework 2015 Versión 4.6.01586.



**JavaScript.**

JavaScript (JS) es un [lenguaje de programación](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n) [interpretado](https://es.wikipedia.org/wiki/Int%C3%A9rprete_(inform%C3%A1tica)), dialecto del estándar [ECMAScript](https://es.wikipedia.org/wiki/ECMAScript" \o "ECMAScript). Se define como [orientado a objetos](https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos), [basado en prototipos](https://es.wikipedia.org/wiki/Programaci%C3%B3n_basada_en_prototipos), [imperativo](https://es.wikipedia.org/wiki/Programaci%C3%B3n_imperativa), débilmente tipado y dinámico. Se utiliza principalmente en su forma del [lado del cliente (client-side)](https://es.wikipedia.org/wiki/Lado_del_cliente), implementado como parte de un [navegador web](https://es.wikipedia.org/wiki/Navegador_web) permitiendo mejoras en la [interfaz de usuario](https://es.wikipedia.org/wiki/Interfaz_de_usuario) y [páginas web](https://es.wikipedia.org/wiki/P%C3%A1gina_web) dinámicas aunque existe una forma de JavaScript del [lado del servidor](https://es.wikipedia.org/wiki/Script_del_lado_del_servidor)(Server-side JavaScript o SSJS). Su uso en [aplicaciones](https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_inform%C3%A1tica) externas a la [web](https://es.wikipedia.org/wiki/World_Wide_Web), por ejemplo en documentos [PDF](https://es.wikipedia.org/wiki/PDF), aplicaciones de escritorio (mayoritariamente [widgets](https://es.wikipedia.org/wiki/Widget)) es también significativo.

Desde el 2012, todos los navegadores modernos soportan completamente ECMAScript 5.1, una versión de javascript. Los navegadores más antiguos soportan por lo menos ECMAScript 3. La sexta edición se liberó en julio del 2015.

JavaScript se diseñó con una sintaxis similar a [C](https://es.wikipedia.org/wiki/C_(lenguaje_de_programaci%C3%B3n)), aunque adopta nombres y convenciones del lenguaje de programación [Java](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n)). Sin embargo, Java y JavaScript tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del [Document Object Model](https://es.wikipedia.org/wiki/Document_Object_Model" \o "Document Object Model) (DOM).

Tradicionalmente se venía utilizando en páginas web [HTML](https://es.wikipedia.org/wiki/HTML) para realizar operaciones y únicamente en el marco de la [aplicación cliente](https://es.wikipedia.org/wiki/Cliente_(inform%C3%A1tica)), sin acceso a funciones del [servidor](https://es.wikipedia.org/wiki/Servidor). Actualmente es ampliamente utilizado para enviar y recibir información del servidor junto con ayuda de otras tecnologías como [AJAX](https://es.wikipedia.org/wiki/AJAX). JavaScript se interpreta en el [agente de usuario](https://es.wikipedia.org/wiki/Agente_de_usuario) al mismo tiempo que las sentencias van descargándose junto con el código [HTML](https://es.wikipedia.org/wiki/HTML).

Desde el lanzamiento en junio de 1997 del estándar [ECMAScript](https://es.wikipedia.org/wiki/ECMAScript" \o "ECMAScript) 1, han existido las versiones 2, 3 y 5, que es la más usada actualmente (la 4 se abandonó ). En junio de 2015 se cerró y publicó la versión ECMAScript 6.

Nacimiento de JavaScript.

JavaScript fue desarrollado originalmente por [Brendan Eich](https://es.wikipedia.org/wiki/Brendan_Eich" \o "Brendan Eich) de [Netscape](https://es.wikipedia.org/wiki/Netscape_Communications_Corporation) con el nombre de Mocha, el cual fue renombrado posteriormente a LiveScript, para finalmente quedar como JavaScript. El cambio de nombre coincidió aproximadamente con el momento en que Netscape agregó compatibilidad con la tecnología Java en su [navegador web Netscape Navigator](https://es.wikipedia.org/wiki/Netscape_Navigator) en la versión 2.002 en diciembre de 1995. La denominación produjo confusión, dando la impresión de que el lenguaje es una prolongación de Java, y se ha caracterizado por muchos como una estrategia de mercadotecnia de Netscape para obtener prestigio e innovar en el ámbito de los nuevos lenguajes de programación web.

«JAVASCRIPT» es una [marca registrada](https://es.wikipedia.org/wiki/Marca_registrada) de [Oracle Corporation](https://es.wikipedia.org/wiki/Oracle_Corporation). Es usada con licencia por los productos creados por [Netscape Communications](https://es.wikipedia.org/wiki/Netscape_Communications) y entidades actuales como la [Fundación Mozilla](https://es.wikipedia.org/wiki/Fundaci%C3%B3n_Mozilla).

[Microsoft](https://es.wikipedia.org/wiki/Microsoft) dio como nombre a su dialecto de JavaScript «[JScript](https://es.wikipedia.org/wiki/JScript" \o "JScript)», para evitar problemas relacionadas con la marca. JScript fue adoptado en la versión 3.0 de Internet Explorer, liberado en agosto de 1996, e incluyó compatibilidad con el [Efecto 2000](https://es.wikipedia.org/wiki/Efecto_2000) con las funciones de fecha, una diferencia de los que se basaban en ese momento. Los dialectos pueden parecer tan similares que los términos «JavaScript» y «JScript» a menudo se utilizan indistintamente, pero la especificación de JScript es incompatible con la de ECMA en muchos aspectos.

Para evitar estas incompatibilidades, el [World Wide Web Consortium](https://es.wikipedia.org/wiki/World_Wide_Web_Consortium" \o "World Wide Web Consortium) diseñó el estándar [Document Object Model](https://es.wikipedia.org/wiki/Document_Object_Model" \o "Document Object Model) (DOM, o Modelo de Objetos del Documento en español), que incorporan [Konqueror](https://es.wikipedia.org/wiki/Konqueror" \o "Konqueror), las versiones 6 de [Internet Explorer](https://es.wikipedia.org/wiki/Internet_Explorer) y [Netscape Navigator](https://es.wikipedia.org/wiki/Netscape_Navigator), [Opera](https://es.wikipedia.org/wiki/Opera_(navegador)) la versión 7, [Mozilla Application Suite](https://es.wikipedia.org/wiki/Mozilla_Application_Suite) y [Mozilla Firefox](https://es.wikipedia.org/wiki/Mozilla_Firefox) desde su primera versión.

En 1997 los autores propusieron JavaScript para que fuera adoptado como estándar de la European Computer Manufacturers 'Association [ECMA](https://es.wikipedia.org/wiki/ECMA), que a pesar de su nombre no es europeo sino internacional, con sede en Ginebra. En junio de 1997 fue adoptado como un estándar ECMA, con el nombre de [ECMAScript](https://es.wikipedia.org/wiki/ECMAScript" \o "ECMAScript). Poco después también como un estándar [ISO](https://es.wikipedia.org/wiki/Organizaci%C3%B3n_Internacional_para_la_Estandarizaci%C3%B3n).

JavaScript en el lado servidor.

Netscape introdujo una implementación de [script del lado del servidor](https://es.wikipedia.org/wiki/Script_del_lado_del_servidor) con [Netscape Enterprise Server](https://es.wikipedia.org/w/index.php?title=Netscape_Enterprise_Server&action=edit&redlink=1), lanzada en diciembre de 1994 (poco después del lanzamiento de JavaScript para navegadores web). A partir de mediados de la década de los 2000, ha habido una proliferación de implementaciones de JavaScript para el lado servidor. [Node.js](https://es.wikipedia.org/wiki/Node.js) es uno de los notables ejemplos de JavaScript en el lado del servidor, siendo usado en proyectos importantes.

Desarrollos posteriores.

JavaScript se ha convertido en uno de los lenguajes de programación más populares en internet. Al principio, sin embargo, muchos desarrolladores renegaban del lenguaje porque el público al que va dirigido lo formaban publicadores de artículos y demás aficionados, entre otras razones. La llegada de [Ajax](https://es.wikipedia.org/wiki/AJAX) devolvió JavaScript a la fama y atrajo la atención de muchos otros programadores. Como resultado de esto hubo una proliferación de un conjunto de [frameworks](https://es.wikipedia.org/wiki/Framework_para_aplicaciones_web" \o "Framework para aplicaciones web) y [librerías](https://es.wikipedia.org/wiki/Biblioteca_(inform%C3%A1tica)) de ámbito general, mejorando las prácticas de programación con JavaScript, y aumentado el uso de JavaScript fuera de los [navegadores web](https://es.wikipedia.org/wiki/Navegador_web), como se ha visto con la proliferación de entornos [JavaScript del lado del servidor](https://es.wikipedia.org/w/index.php?title=Server-side_JavaScript&action=edit&redlink=1). En enero de 2009, el proyecto [CommonJS](https://es.wikipedia.org/w/index.php?title=CommonJS&action=edit&redlink=1" \o "CommonJS (aún no redactado)) fue inaugurado con el objetivo de especificar una librería para uso de tareas comunes principalmente para el desarrollo fuera del navegador web.

En junio de 2015 se cerró y publicó el estándar [ECMAScript](https://es.wikipedia.org/wiki/ECMAScript" \o "ECMAScript) 6 con un soporte irregular entre navegadores y que dota a JavaScript de características avanzadas que se echaban de menos y que son de uso habitual en otros lenguajes como, por ejemplo, módulos para organización del código, verdaderas [clases](https://es.wikipedia.org/wiki/Clase_(inform%C3%A1tica)) para [programación orientada a objetos](https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos), expresiones de flecha, iteradores, generadores o promesas para programación asíncrona.

La versión 7 de ECMAScript se conoce como ECMAScript 2016, y es la última versión disponible, publicada en junio de 2016. Se trata de la primera versión para la que se usa un nuevo procedimiento de publicación anual y un proceso de desarrollo abierto.

Características.

Las siguientes características son comunes a todas las implementaciones que se ajustan al estándar ECMAScript, a menos que especifique explícitamente en caso contrario.

Imperativo y estructurado.

JavaScript es compatible con gran parte de la [estructura de programación](https://es.wikipedia.org/wiki/Programaci%C3%B3n_estructurada) de [C](https://es.wikipedia.org/wiki/C_(lenguaje_de_programaci%C3%B3n)) (por ejemplo, sentencias if, bucles for, sentencias switch, etc.). Con una salvedad, en parte: en C, el [ámbito](https://es.wikipedia.org/wiki/%C3%81mbito_(programaci%C3%B3n)) de las [variables](https://es.wikipedia.org/wiki/Variable_(programaci%C3%B3n)) alcanza al bloque en el cual fueron definidas; sin embargo JavaScript no es compatible con esto, puesto que el ámbito de las variables es el de la función en la cual fueron declaradas. Esto cambia con la versión de ECMAScript 2015, ya que añade compatibilidad con block scoping por medio de la palabra clave let. Como en C, JavaScript hace distinción entre [expresiones](https://es.wikipedia.org/wiki/Expresi%C3%B3n_(inform%C3%A1tica)) y sentencias. Una diferencia sintáctica con respecto a C es la inserción automática de punto y coma, es decir, en JavaScript los puntos y coma que finalizan una sentencia pueden ser omitidos.

Dinámico.

Tipado dinámico.

Como en la mayoría de [lenguajes de scripting](https://es.wikipedia.org/wiki/Script), el [tipo](https://es.wikipedia.org/wiki/Sistema_de_tipos) está asociado al valor, no a la variable. Por ejemplo, una variable x en un momento dado puede estar ligada a un número y más adelante, religada a una [cadena](https://es.wikipedia.org/wiki/String). JavaScript es compatible con varias formas de comprobar el tipo de un objeto, incluyendo [duck typing](https://es.wikipedia.org/wiki/Duck_typing" \o "Duck typing). Una forma de saberlo es por medio de la palabra clave typeof.

Objetual.

JavaScript está formado casi en su totalidad por [objetos](https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos). Los objetos en JavaScript son [arrays asociativos](https://es.wikipedia.org/wiki/Vector_asociativo" \o "Vector asociativo), mejorados con la inclusión de prototipos (ver más adelante). Los nombres de las propiedades de los objetos son claves de tipo cadena: obj.x = 10 y obj['x'] = 10 son equivalentes, siendo la notación con punto [azúcar sintáctico](https://es.wikipedia.org/wiki/Az%C3%BAcar_sint%C3%A1ctico). Las propiedades y sus valores pueden ser creados, cambiados o eliminados en tiempo de ejecución. La mayoría de propiedades de un objeto (y aquellas que son incluidas por la cadena de la herencia prototípica) pueden ser enumeradas a por medio de la instrucción de bucle for... in. JavaScript tiene un pequeño número de objetos predefinidos como son Function y Date.

Evaluación en tiempo de ejecución.

JavaScript incluye la función eval que permite evaluar expresiones como expresadas como cadenas en tiempo de ejecución. Por ello se recomienda que eval sea utilizado con precaución y que se opte por utilizar la función JSON.parse() en la medida de lo posible, pues resulta mucho más segura.

Funcional.

Funciones de primera clase.

A las [funciones](https://es.wikipedia.org/wiki/Subrutinas) se les suele llamar ciudadanos de primera clase; son objetos en sí mismos. Como tal, poseen propiedades y métodos, como .call() y .bind(). Una función anidada es una función definida dentro de otra. Esta es creada cada vez que la función externa es invocada. Además, cada función creada forma una [clausura](https://es.wikipedia.org/wiki/Clausura_(inform%C3%A1tica)); es el resultado de evaluar un ámbito conteniendo en una o más variables dependientes de otro ámbito externo, incluyendo constantes, variables locales y argumentos de la función externa llamante. El resultado de la evaluación de dicha clausura forma parte del estado interno de cada objeto función, incluso después de que la función exterior concluya su evaluación.

Prototípico.

Prototipos.

JavaScript usa [prototipos](https://es.wikipedia.org/wiki/Programaci%C3%B3n_basada_en_prototipos) en vez de [clases](https://es.wikipedia.org/wiki/Clase_(inform%C3%A1tica)) para el uso de [herencia](https://es.wikipedia.org/wiki/Herencia_(inform%C3%A1tica)). Es posible llegar a emular muchas de las características que proporcionan las clases en lenguajes orientados a objetos tradicionales por medio de prototipos en JavaScript.

Funciones como constructores de objetos.

Las funciones también se comportan como constructores. Prefijar una llamada a la función con la palabra clave new crear una nueva instancia de un prototipo, que heredan propiedades y métodos del constructor (incluidas las propiedades del prototipo de Object). ECMAScript 5 ofrece el método Object.create, permitiendo la creación explícita de una instancia sin tener que heredar automáticamente del prototipo de Object (en entornos antiguos puede aparecer el prototipo del objeto creado como null). La propiedad prototype del constructor determina el objeto usado para el prototipo interno de los nuevos objetos creados. Se pueden añadir nuevos métodos modificando el prototipo del objeto usado como constructor. Constructores predefinidos en JavaScript, como Array u Object, también tienen prototipos que pueden ser modificados. Aunque esto sea posible se considera una mala práctica modificar el prototipo de Object ya que la mayoría de los objetos en Javascript heredan los métodos y propiedades del objeto prototype, objetos los cuales pueden esperar que estos no hayan sido modificados.

Otras características.

Entorno de ejecución.

JavaScript normalmente depende del entorno en el que se ejecute (por ejemplo, en un [navegador web](https://es.wikipedia.org/wiki/Navegador_web)) para ofrecer objetos y métodos por los que los scripts pueden interactuar con el "mundo exterior". De hecho, depende del entorno para ser capaz de proporcionar la capacidad de incluir o importar scripts (por ejemplo, en [HTML](https://es.wikipedia.org/wiki/HTML) por medio del tag <script>). (Esto no es una característica del lenguaje, pero es común en la mayoría de las implementaciones de JavaScript.)

Funciones variádicas.

Un número indefinido de parámetros pueden ser pasados a la función. La función puede acceder a ellos a través de los parámetros o también a través del objeto local arguments. Las [funciones variádicas](https://es.wikipedia.org/w/index.php?title=Funci%C3%B3n_vari%C3%A1dica&action=edit&redlink=1) también pueden ser creadas usando el método .apply().

Funciones como métodos.

A diferencia de muchos lenguajes orientados a objetos, no hay distinción entre la definición de función y la definición de [método](https://es.wikipedia.org/wiki/M%C3%A9todo_(inform%C3%A1tica)). Más bien, la distinción se produce durante la llamada a la función; una función puede ser llamada como un método. Cuando una función es llamada como un método de un objeto, la palabra clave this, que es una variable local a la función, representa al objeto que invocó dicha función.

Arrays y la definición literal de objetos.

Al igual que muchos lenguajes de script, arrays y objetos ([arrays asociativos](https://es.wikipedia.org/wiki/Vector_asociativo" \o "Vector asociativo) en otros idiomas) pueden ser creados con una sintaxis abreviada. De hecho, estos literales forman la base del formato de datos [JSON](https://es.wikipedia.org/wiki/JSON).

Expresiones regulares.

JavaScript también es compatible con [expresiones regulares](https://es.wikipedia.org/wiki/Expresiones_regulares) de una manera similar a [Perl](https://es.wikipedia.org/wiki/Perl), que proporcionan una sintaxis concisa y poderosa para la manipulación de texto que es más sofisticado que las funciones incorporadas a los objetos de tipo string.

Extensiones específicas del fabricante.

JavaScript se encuentra oficialmente bajo la organización de [Mozilla Foundation](https://es.wikipedia.org/wiki/Mozilla_Foundation), y periódicamente se añaden nuevas características del lenguaje. Sin embargo, sólo algunos [motores JavaScript](https://es.wikipedia.org/wiki/Int%C3%A9rprete_de_JavaScript) son compatibles con estas características:

* Las propiedades get y set (también compatibles con WebKit, Opera,[35](https://es.wikipedia.org/wiki/JavaScript#cite_note-35) ActionScript y Rhino).
* Cláusulas catch condicionales.
* Protocolo iterador adoptado de [Python](https://es.wikipedia.org/wiki/Python).
* Corrutinas también adoptadas de Python.
* Generación de listas y expresiones por comprensión también adoptado de Python.
* Establecer el ámbito a bloque a través de la palabra clave let.
* Desestructuración de arrays y objetos (forma limita de emparejamiento de patrones).
* Expresiones concretas en funciones (function(args) expr).
* [ECMAScript para XML](https://es.wikipedia.org/w/index.php?title=ECMAScript_para_XML&action=edit&redlink=1) (E4X), una extensión que añade compatibilidad nativa XML a ECMAScript.

Uso de Páginas Web.

El uso más común de JavaScript es escribir funciones embebidas o incluidas en páginas HTML y que interactúan con el Document Object Model (DOM o Modelo de Objetos del Documento) de la página. Algunos ejemplos sencillos de este uso son:

Cargar nuevo contenido para la página o enviar datos al servidor a través de AJAX sin necesidad de recargar la página (por ejemplo, una red social puede permitir al usuario enviar actualizaciones de estado sin salir de la página).

Animación de los elementos de página, hacerlos desaparecer, cambiar su tamaño, moverlos, etc.

Contenido interactivo, por ejemplo, juegos y reproducción de audio y vídeo.

Validación de los valores de entrada de un formulario web para asegurarse de que son aceptables antes de ser enviado al servidor.

Transmisión de información sobre los hábitos de lectura de los usuarios y las actividades de navegación a varios sitios web. Las páginas Web con frecuencia lo hacen para hacer análisis web, seguimiento de anuncios, la personalización o para otros fines.

Dado que el código JavaScript puede ejecutarse localmente en el navegador del usuario (en lugar de en un servidor remoto), el navegador puede responder a las acciones del usuario con rapidez, haciendo una aplicación más sensible. Por otra parte, el código JavaScript puede detectar acciones de los usuarios que HTML por sí sola no puede, como pulsaciones de teclado. Las aplicaciones como [Gmail](https://es.wikipedia.org/wiki/Gmail) se aprovechan de esto: la mayor parte de la lógica de la interfaz de usuario está escrita en JavaScript, enviando peticiones al servidor (por ejemplo, el contenido de un mensaje de correo electrónico). La tendencia cada vez mayor por el uso de la programación Ajax explota de manera similar esta técnica.

Un motor de JavaScript (también conocido como intérprete de JavaScript o implementación JavaScript) es un intérprete que interpreta el código fuente de JavaScript y ejecuta la secuencia de comandos en consecuencia. El primer motor de JavaScript fue creado por Brendan Eich en Netscape Communications Corporation, para el navegador web Netscape Navigator. El motor, denominado SpiderMonkey, está implementado en C. Desde entonces, ha sido actualizado (en JavaScript 1.5) para cumplir con el ECMA-262 edición 3. El motor Rhino, creado principalmente por Norris Boyd (antes de Netscape, ahora en Google) es una implementación de JavaScript en [Java](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n)). Rhino, como SpiderMonkey, es compatible con el ECMA-262 edición 3.

Un navegador web es, con mucho, el entorno de acogida más común para JavaScript. Los navegadores web suelen crear objetos no nativos, dependientes del entorno de ejecución, para representar el Document Object Model (DOM) en JavaScript. El servidor web es otro entorno común de servicios. Un servidor web JavaScript suele exponer sus propios objetos para representar objetos de petición y respuesta HTTP, que un programa JavaScript podría entonces interrogar y manipular para generar dinámicamente páginas web.

Debido a que JavaScript es el único lenguaje por el que los más populares navegadores comparten su apoyo, se ha convertido en un lenguaje al que muchos frameworks en otros lenguajes compilan, a pesar de que JavaScript no fue diseñado para tales propósitos. A pesar de las limitaciones de rendimiento inherentes a su naturaleza dinámica, el aumento de la velocidad de los motores de JavaScript ha hecho de este lenguaje un entorno para la compilación sorprendentemente factible.

Consideraciones acerca de la compatibilidad.

Artículo principal: [Interoperabilidad web](https://es.wikipedia.org/w/index.php?title=Interoperabilidad_web&action=edit&redlink=1).

Debido a que JavaScript se ejecuta en entornos muy variados, una parte importante de las pruebas y la depuración es probar y verificar que el código JavaScript funciona correctamente en múltiples navegadores. La interfaz DOM para acceder y manipular páginas web no es parte del estándar ECMAScript, o de la propia JavaScript. El DOM es definido por los esfuerzos de estandarización del [W3C](https://es.wikipedia.org/wiki/W3C), una organización independiente. En la práctica, las implementaciones que hacen de JavaScript los distintos navegadores difieren tanto entre ellos mismos como de las normas del estándar.

Para hacer frente a estas diferencias, los autores de JavaScript pudieron ser capaces de escribir código compatible con los estándares que también fuera capaz de ejecutarse correctamente en la mayoría de los navegadores, o en su defecto, que al menos se pudiera escribir código capaz de comprobar la presencia de ciertas funcionalidades del navegador y que se comportase de manera diferente si no se dispusiese de dicha funcionalidad. Existen casos en los que dos navegadores pueden llegar a implementar la misma característica, pero con un comportamiento diferente, hecho que a los programadores les puede resultar de ayuda para detectar qué navegador se está ejecutando en ese instante y así cambiar el comportamiento de su escritura para que coincida. Los programadores también suelen utilizar bibliotecas o herramientas que tengan en cuenta las diferencias entre navegadores.

Además, los scripts pueden no funcionar para algunos usuarios. Por ejemplo, un usuario puede:

* utilizar un navegador antiguo sin compatibilidad completa con la API DOM,
* utilizar un navegador PDA o teléfono móvil que no puede ejecutar JavaScript
* tener la ejecución de JavaScript deshabilitada, como precaución de seguridad,
* utilizar un navegador de voz debido a, por ejemplo, una discapacidad visual.

Para apoyar a estos usuarios, los programadores web suelen crear páginas que sean [tolerante a fallos](https://es.wikipedia.org/wiki/Tolerancia_a_fallos) según el agente de usuario (tipo de navegador) que no admita JavaScript. En particular, la página debe seguir siendo útil, aunque sin las características adicionales que JavaScript habría añadido. Un enfoque alternativo que muchos encuentran preferible es primero crear contenido utilizando las tecnologías que funcionan en todos los navegadores, y mejorar el contenido para los usuarios que han permitido JavaScript.

Accesibilidad.

Artículo principal: [Accesibilidad web](https://es.wikipedia.org/wiki/Accesibilidad_web).

Suponiendo que el usuario no haya desactivado la ejecución de código JavaScript, en el lado del cliente JavaScript debe ser escrito tanto con el propósito de mejorar las experiencias de los visitantes con discapacidad visual o física, como el de evitar ocultar información a estos visitantes.

Los [lectores de pantalla](https://es.wikipedia.org/wiki/Lector_de_pantalla), utilizados por los ciegos y deficientes visuales, pueden ser tenidos en cuenta por JavaScript y así poder acceder y leer los elementos DOM de la página. El código HTML escrito debe ser lo más conciso, navegable y semánticamente rico posible, tanto si JavaScript se ejecuta como si no.

JavaScript no debería de ser totalmente dependiente de los eventos de ratón del navegador y debería ser accesible para aquellos usuarios que no quieran hacer uso del [ratón (informática)](https://es.wikipedia.org/wiki/Rat%C3%B3n_(inform%C3%A1tica)) para navegar o que opten por utilizar solamente el teclado. Hay eventos independientes del dispositivo, tales como onfocus y onchange que son preferibles en la mayoría de los casos.

JavaScript no debe ser utilizado para crear confusión o desorientación al usuario web. Por ejemplo, modificar o desactivar la funcionalidad normal del navegador, como cambiar la forma en que el botón de navegar hacia atrás o el evento de actualización se comportan, son prácticas que generalmente son mejores evitar. Igualmente, desencadenar eventos que el usuario puede no tener en cuenta reduce la sensación de control del usuario y provoca cambios inesperados al contenido de la página.

A menudo, el proceso de dotar a una página web compleja el mayor grado [accesibilidad](https://es.wikipedia.org/wiki/Accesibilidad_web) posible, se convierte en un problema no trivial donde muchos temas se acaban llevando al debate y a la opinión, siendo necesario el compromiso de todos hasta el final. Sin embargo, los agentes de usuario y las tecnologías de apoyo a personas con discapacidad están en constante evolución y nuevas directrices e información al respecto siguen publicándose en la web.

Vulnerabilidades cross-site.

Artículos principales: [Cross-site scripting](https://es.wikipedia.org/wiki/Cross-site_scripting) y [Cross Site Request Forgery](https://es.wikipedia.org/wiki/Cross_Site_Request_Forgery).

Un problema común de seguridad en JavaScript es el [cross-site scripting](https://es.wikipedia.org/wiki/Cross-site_scripting" \o "Cross-site scripting) o XSS, una violación de la política de mismo origen. Las vulnerabilidades XSS permiten a un atacante inyectar código JavaScript en páginas web visitadas por el usuario. Una de esas webs podría ser la de un banco, pudiendo el atacante acceder a la aplicación de banca con los privilegios de la víctima, lo que podría revelar información secreta o transferir dinero sin la autorización de la víctima. Una solución para las vulnerabilidades XSS es utilizar HTML escaping cuando se muestre información de fuentes no confiables

Algunos navegadores incluyen una protección parcial contra los ataques XSS reflejados (el atacante está en la misma petición web). El atacante proporciona una URL incluyendo código malicioso. Sin embargo, incluso los usuarios de los navegadores son vulnerables a otros ataques XSS, tales como aquellos en los que el código malicioso se almacena en una base de datos. Sólo el correcto diseño de las aplicaciones Web en la parte servidora puede prevenir totalmente XSS. Las vulnerabilidades XSS también pueden ocurrir debido a errores de ejecución por los desarrolladores del navegador.

Otra vulnerabilidad es la falsificación de petición de sitio cruzado o CSRF. En CSRF, el código del sitio web atacante engaña al navegador de la víctima, permitiendo al atacante realizar peticiones en nombre de la víctima, haciendo imposible saber a la aplicación de destino (por ejemplo, la de un banco haciendo una transferencia de dinero) saber si la petición ha sido realizada voluntariamente por el usuario o por un ataque CSRF.

El ataque funciona porque, si el sitio de destino hace uso únicamente de las cookies para autenticar las solicitudes de la víctima, las peticiones iniciadas por el código del atacante tendrán las mismas credenciales de acceso legítimo que las solicitudes iniciadas por el propio usuario.

En general, la solución a CSRF consiste en introducir un campo de formulario oculto cuyo valor se utilice para realizar la autenticación, y no sólo por medio de las cookies, en solicitudes que puedan tener efectos duraderos. La comprobación de la cabecera HTTP referer también puede servir de ayuda.

"Hijacking JavaScript" es un tipo de ataque CSRF en el que una etiqueta <script> en el sitio web del atacante explota una vulnerabilidad en la página del sitio de la víctima que le hace devolver información privada, en forma de JSON o código JavaScript. Las posibles soluciones son:

que se requiera un token de autenticación en los parámetros de las peticiones POST y GET para aquellas peticiones que requieran devolver información privada del usuario.

usar POST y nunca GET para solicitudes que devuelven información privada

Herramientas de desarrollo.

En JavaScript, disponer de un depurador se convierte en necesario cuando se desarrollan grandes aplicaciones, no triviales. Dado que puede haber diferencias de implementación entre los diferentes navegadores (especialmente en cuanto al DOM), es útil tener acceso a un depurador para cada uno de los navegadores a los cuales nuestra aplicación web irá dirigido.

Los depuradores web están disponibles para Internet Explorer, Firefox, Safari, Google Chrome y Opera.

Existen tres depuradores disponibles para [Internet Explorer](https://es.wikipedia.org/wiki/Internet_Explorer): [Microsoft Visual Studio](https://es.wikipedia.org/wiki/Microsoft_Visual_Studio) es el más avanzado de los tres, seguido de cerca por Microsoft Script Editor (un componente de [Microsoft Office](https://es.wikipedia.org/wiki/Microsoft_Office)) y, finalmente, Microsoft Script Debugger, que es mucho más básico que el otro dos, aunque es gratuito. El [IDE](https://es.wikipedia.org/wiki/Entorno_de_desarrollo_integrado) gratuito Microsoft Visual Web Developer Express ofrece una versión limitada de la funcionalidad de depuración de JavaScript en el Microsoft Visual Studio. Internet Explorer ha incluido herramientas de desarrollo desde la versión 8 (se muestra pulsando la tecla F12). Las aplicaciones web dentro de [Firefox](https://es.wikipedia.org/wiki/Firefox) se pueden depurar usando el [Firebug](https://es.wikipedia.org/wiki/Firebug" \o "Firebug) add-on o el antiguo depurador Venkman. Firefox también tiene integrada una consola de errores básica, que registra y evalúa JavaScript. También registra errores de [CSS](https://es.wikipedia.org/wiki/Hojas_de_estilo_en_cascada) y advertencias. Opera incluye un conjunto de herramientas llamado Dragonfly. El Inspector Web de [WebKit](https://es.wikipedia.org/wiki/WebKit" \o "WebKit) incluye un depurador de JavaScript utilizado en [Safari](https://es.wikipedia.org/wiki/Safari), junto con una versión modificada de [Google Chrome](https://es.wikipedia.org/wiki/Google_Chrome).

Existen algunas herramientas de ayuda a la depuración, también escritas en JavaScript y construidas para ejecutarse en la Web. Un ejemplo es el programa [JSLint](https://es.wikipedia.org/w/index.php?title=JSLint&action=edit&redlink=1" \o "JSLint (aún no redactado)), desarrollado por [Douglas Crockford](https://es.wikipedia.org/w/index.php?title=Douglas_Crockford&action=edit&redlink=1), quien ha escrito extensamente sobre el lenguaje. JSLint analiza el código JavaScript para que este quede conforme con un conjunto de normas y directrices y que aseguran su correcto funcionamiento y mantenibilidad.

**Suite de DevExpress.**

DevExpress es el mejor aliado en el desarrollo. Es una de las más completas suites de componentes de UI para el desarrollo en todas las plataformas de .NE T como Windows Forms, ASP.NET, MVC, Silverlight y Windows 8 XAML. Hoy en día los usuarios están demandando cada vez más ingenieros de software. La presión para ofrecer más servicios con mayor calidad en menos tiempo nunca ha sido tan grande. A medida que la complejidad de los sistemas de software aumenta, crecen también los desafíos que enfrentamos. Estos desafíos conducen inevitablemente a un número creciente de tareas tediosas y consumen un tiempo precioso que conspiran para esparcir su creatividad y productividad.

Las herramientas de desarrollo de Developer Express están diseñadas específicamente para ayudarlo a satisfacer estas crecientes demandas de calidad y funcionalidad, por el aumento de productividad, eliminando el tedio.

Con los componentes de Developer Express el desarrollo es mucho más fácil, rápido e intuitivo.

Ventajas del uso de DevExpress.

Si bien en la actualidad existe un sin número de suites de componentes muy buenas, DevExpress se encuentra sobre ellas por varias razones, como son:

* Cuenta con controles para todas las plataformas de Microsoft Windows.
* Posee más de 90 controles mediante los cuales se pueden diseñar aplicaciones de alta complejidad.
* La creación de los componentes es sami – automática, DevExpress se encarga de realizar todo el código necesario para la visualización y llenado de los componentes según la plataforma utilizada.
* Permite llenar de manera sencilla cada uno de los componentes con información traída de una conexión de base de datos.
* Mejora el rendimiento de las aplicaciones al optimizar el código de llenado de las vistas.
* Funciona en cualquier explorador. (Aplicaciones Web)
* El tiempo de desarrollo se reduce considerablemente al no tener que teclear todo el código.
* Incluye plantillas predefinidas para varios tipos de aplicaciones.
* Incluye la herramienta Theme Builder, la cual permite editar el estilo de los controles y genera automáticamente la hoja de estilos de la aplicación siguiendo el patrón elegido.
* La documentación que incluye tanto en Visual Studio (descripciones de métodos y parámetros) como externa (documentación de clases y ejemplos) es realmente extensa y útil.
* Incluye video tutoriales y demostraciones sencillas de comprender.
* Es altamente personalizable.
* Cuenta con soporte rápido y eficaz para los problemas que se puedan presentar.

Desventajas de DevExpress.

* Solo funciona para plataformas de  Microsoft.
* Existen algunas limitantes al incluir los componentes, ya que estos ya tienen características y comportamientos definidos.
* Requiere licencia.

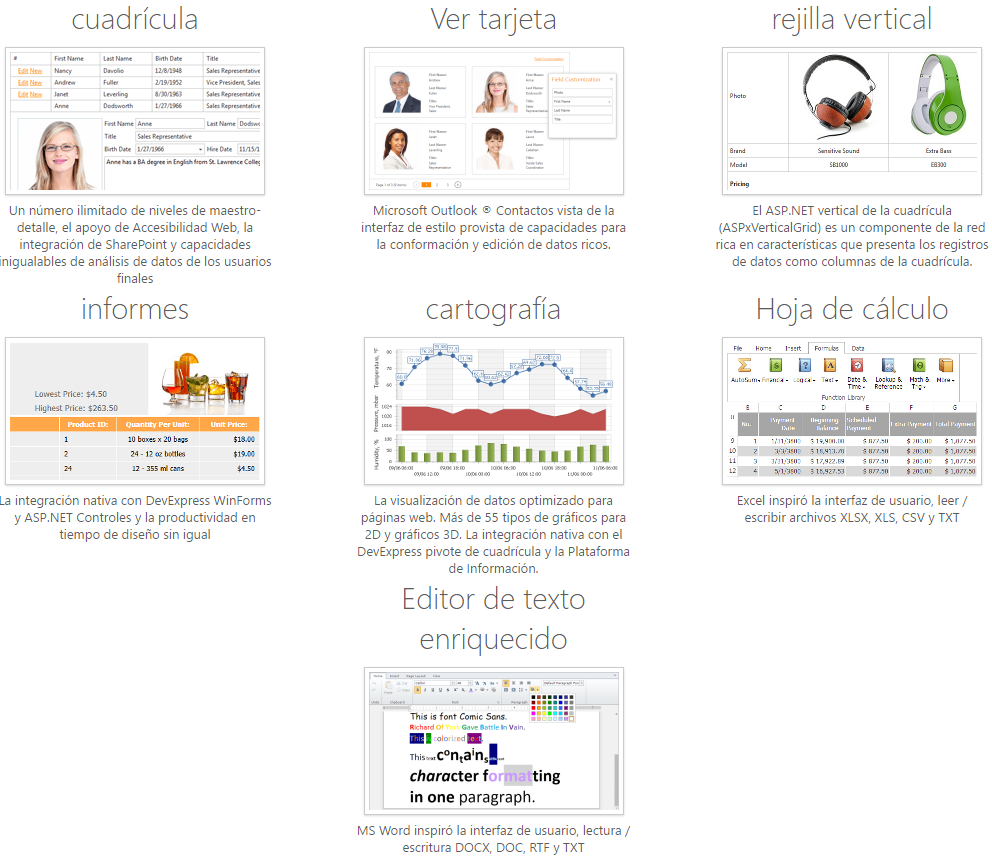
Con los años, DevExpress ha ganado numerosos premios de la industria para sus controles y herramientas de desarrollo, junto con nuestra influencia dentro de la comunidad de desarrolladores. Varios de los premios fueron elegidos por los desarrolladores de software de todo el mundo lo cual refleja la pasión de programadores y desarrolladores fieles para las tecnologías de DevExpress.

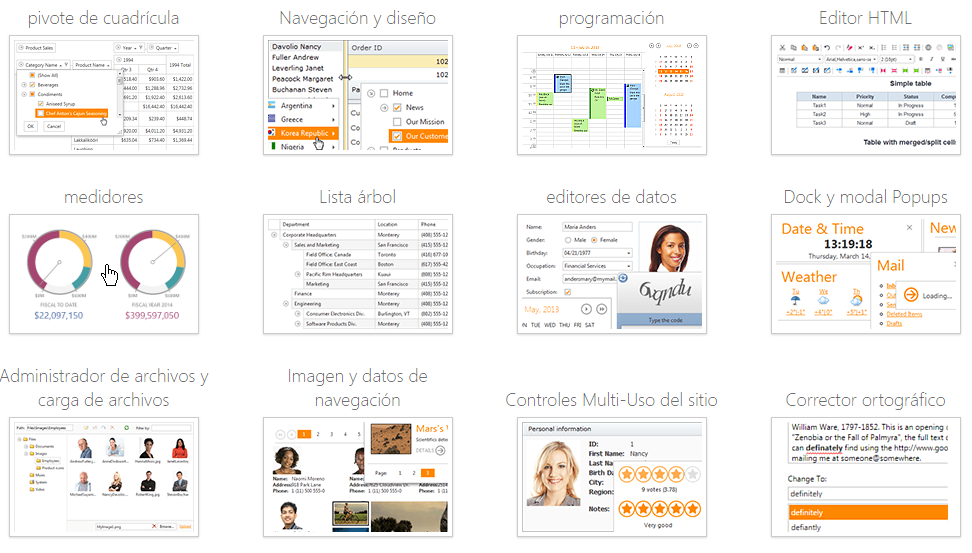
Mucho más importante que cualquier premio, sin embargo, es la realidad de que todos los controles de DevExpress abrazan la solución de las necesidades empresariales de los clientes sobre una base diaria. DevExpress está siempre innovando nuevos controles o añadiendo nuevas características a los controles que ya existen, para mejorar su alcance y funcionamiento. Están siempre al pendiente de las solicitudes de cambio o problemas de soporte técnico, es una empresa responsable que cumple y supera las expectativas cada día.





Algunos de los controles para Visual Studio son:

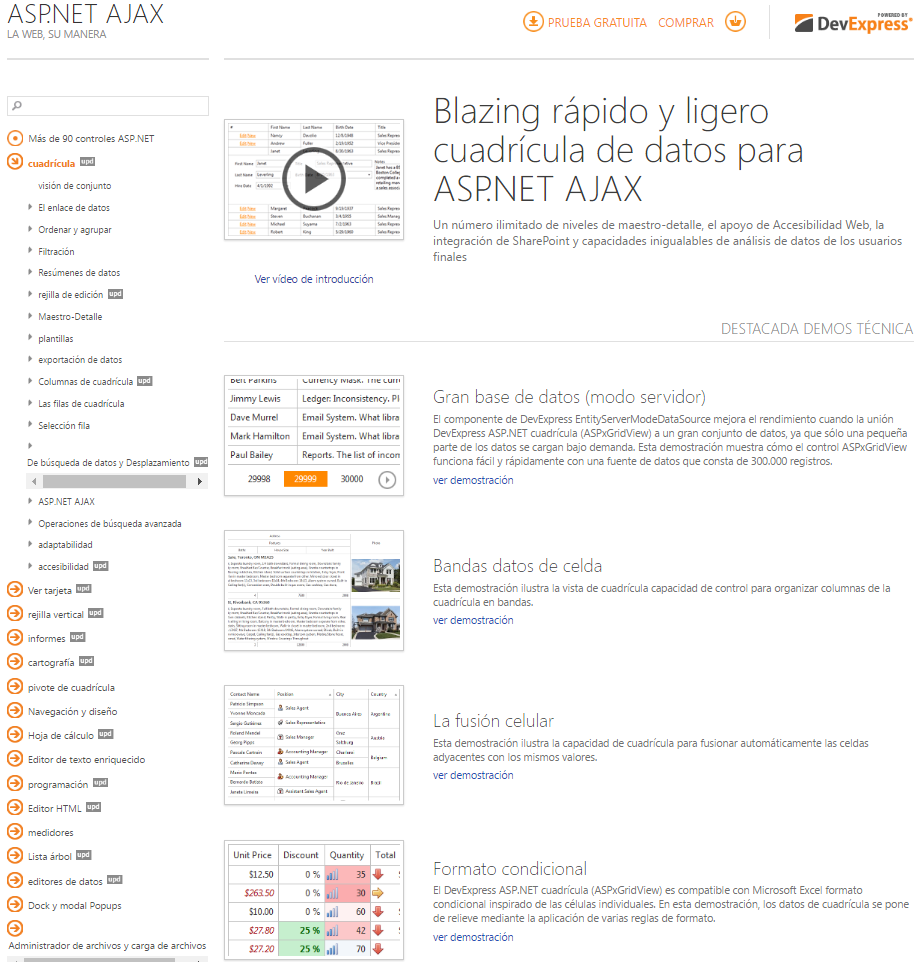




Algunos de los controles básicos para edición de datos:



Algunos de los controles



Con respecto SofTec.

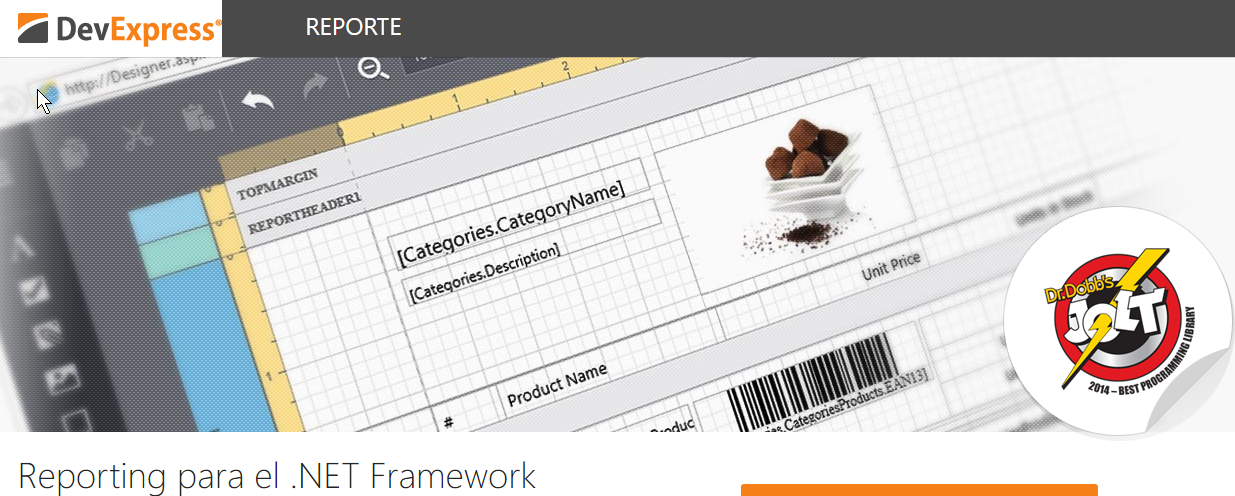
Para el proyecto del Reloj Checador y Justificación de Incidencias, se utilizó la versión 12.1, la cual posteriormente se migro a las 13.1, y por último se migro a la versión 15.1, que es con la que actualmente está funcionando SofTec.

**Suite de XtraReports.**

El éxito a largo plazo de la empresa es totalmente dependiente de la transmisión eficaz de información de misión crítica en toda la organización.

Demostrar el impacto de las decisiones individuales para la gestión como los empleados exige una solución de informes intuitiva, una con excepcionales opciones de rendimiento y personalización.

DevExpress XtraReports ofrece opciones de personalización fáciles de usar y un rico conjunto de controles de informe, incluyendo multi-dimensionales tablas dinámicas y gráficos para que pueda construir informes de claridad informativa sin igual.



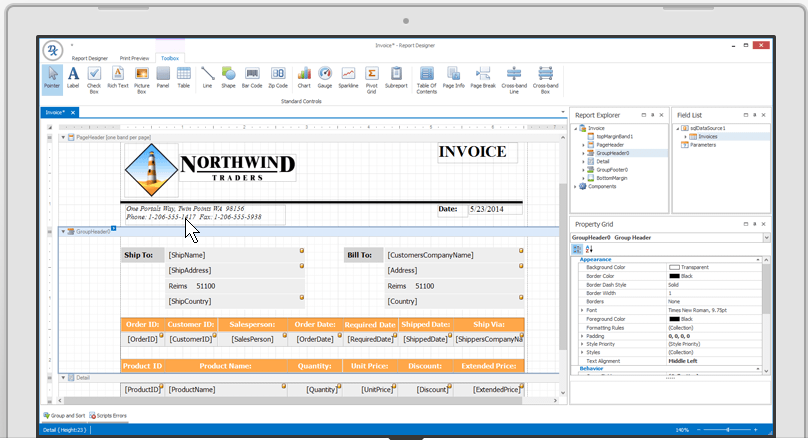
Características integradas que se incluyen como parte DevExpress de informes para .NET:



Entregar datos a su manera.

Información que es compatible con cualquier proveedor de datos.

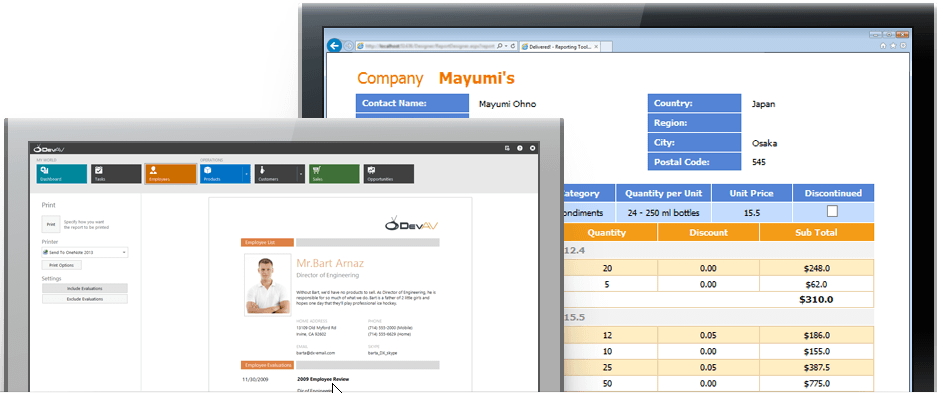
En el mundo actual, el apoyo a todos los principales proveedores de datos, servicios o las API no es un lujo, es un requisito. Herramientas que ponen restricciones sobre el consumo de datos impiden la productividad de la organización de informes. La única opción de futuro es una capa de presentación informe de datos agnóstica, que asegura que cuando se genera un informe, se puede ver en cualquier dispositivo o factor de forma con cualquier origen de base de datos o datos. DevExpress XtraReports garantiza su capacidad para consumir datos de cualquier fuente, a través de su apoyo a los objetos de datos de Visual Studio .NET y  su capacidad para unirse a un informe de datos XML o cualquier objeto de datos que implementa IList, IList <T> y IEnumerable <T > interfaces.



Entregar flexibilidad cada hora.

Informes de .NET que se construye para usted y sus usuarios finales.

El objetivo es simple: proporcionar a los usuarios de negocio con la información necesaria para supervisar los resultados de negocio y tomar decisiones inteligentes en tiempo real. Mientras que el objetivo puede ser simple, la realidad es que la mayoría de los informes son rígidos y no permiten a los usuarios explorar eficientemente la información en sus propios términos. Las necesidades cambiantes de la empresa dictan que una solución de informes ofrecen opciones de personalización del usuario final simples y directas para que los usuarios de informes pueden manipular libremente de salida para maximizar la claridad. La flexibilidad es clave y la facilidad de uso primordial. La dinámica de DevExpress XtraReports le ayuda a superar las limitaciones asociadas con las soluciones tradicionales de información con un totalmente integrado   
conjunto de herramientas de productividad, asistentes de informes, plantillas de informes pre-construidos y diseñadores de informes de los usuarios finales.



Entregar informes totalmente personalizables.

Diseñadores de usuario final para WinForms, ASP.NET, WPF y Silverlight.

Los usuarios avanzados exigen opciones y quieren tener el control sobre la información que distribuyen dentro de la empresa. La necesidad de cumplir estos requisitos   
puede ser desalentador, especialmente para los informes. Con sus potentes diseñadores de informes, DevExpress XtraReports hace que sea fácil de integrar la personalización de informes en tiempo de ejecución en su próximo proyecto. Ya sea que necesite un diseñador de Windows rica en características o prefiere utilizar una herramienta basada en la web 100%, la suite incluye todo lo que necesita para capacitar a sus usuarios y reducir el número de informes que necesita para crear dentro de Visual Studio.

