*Prog01001*

Fundamentación tecnológica

Todo comenzó con Visual Basic (VB), el cual es un [lenguaje de programación](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n) [dirigido por eventos](https://es.wikipedia.org/w/index.php?title=Programaci%C3%B3n_dirigida_por_varios_eventos&action=edit&redlink=1). Este lenguaje de programación es un dialecto de [BASIC](https://es.wikipedia.org/wiki/BASIC), con importantes agregados. Su primera versión fue presentada en [1991](https://es.wikipedia.org/wiki/1991), con la intención de simplificar la programación utilizando un [ambiente de desarrollo.](https://es.wikipedia.org/wiki/Entorno_de_desarrollo_integrado)

En 2001 Microsoft propuso abandonar el desarrollo basado en la [API](https://es.wikipedia.org/wiki/API) [Win32](https://es.wikipedia.org/wiki/Win32) y pasar a un [framework](https://es.wikipedia.org/wiki/Framework) o marco común de librerías, independiente de la versión del sistema operativo [.NET Framework](https://es.wikipedia.org/wiki/.NET_Framework), a través de [Visual Basic .NET](https://es.wikipedia.org/wiki/Visual_Basic_.NET) fue el sucesor de Visual Basic 6.

Ahora centrándonos en 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 considera una evolución de [Visual Basic](https://es.wikipedia.org/wiki/Visual_Basic) implementada sobre el [framework .NET](https://es.wikipedia.org/wiki/Microsoft_.NET), la cual brinda nuevas características como el control estructurado de excepciones, la programación orientada a objetos, o los Genéricos, facilitando así el desarrollo de aplicaciones más avanzadas con herramientas modernas. 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, aunque existen otras alternativas, como [SharpDevelop](https://es.wikipedia.org/wiki/SharpDevelop) (que además es libre, lo que es una ventaja en cuanto a presupuesto).

Las ventajas de este lenguaje con respecto a su antecesor son:

* Completamente orientado a objetos.
* Controles de usuario y personalizados.
* Facilita el mantenimiento de grandes aplicaciones.
* Incremento de velocidad de respuesta del servidor.
* Mayor velocidad.
* Mayor seguridad.
* Posee una [curva de aprendizaje](https://es.wikipedia.org/wiki/Curva_de_aprendizaje) muy rápida.
* Integra el diseño e implementación de formularios de Windows.
* Permite usar con facilidad la plataforma de los sistemas Windows, dado que tiene acceso prácticamente total a la [API](https://es.wikipedia.org/wiki/API) de Windows, incluidas librerías actuales.
* Es uno de los lenguajes de uso más extendido, por lo que resulta fácil encontrar información, documentación y fuentes para los proyectos.
* Fácilmente extensible mediante librerías DLL y componentes ActiveX de otros lenguajes.
* Posibilita añadir soporte para ejecución de scripts, VBScript o JScript, en las aplicaciones mediante *Microsoft Script Control*.[7](https://es.wikipedia.org/wiki/Visual_Basic#cite_note-7)​
* Tiene acceso a la [API](https://es.wikipedia.org/wiki/API) multimedia de DirectX (versiones 7 y 8). También está disponible, de forma no oficial, un componente para trabajar con OpenGL 1.1.[8](https://es.wikipedia.org/wiki/Visual_Basic#cite_note-8)​
* Existe una versión, [VBA](https://es.wikipedia.org/wiki/VBA), integrada en las aplicaciones de [Microsoft Office](https://es.wikipedia.org/wiki/Microsoft_Office), tanto Windows como Mac, que permite programar macros para extender y automatizar funcionalidades en documentos, hojas de cálculo y bases de datos ([Access](https://es.wikipedia.org/wiki/Access)).
* Si bien permite desarrollar grandes y complejas aplicaciones, también provee un entorno adecuado para realizar pequeños prototipos rápidos.