

## **Tarea 3**

Antonio García Sánchez



## Editores de código

### ¿Qué son?

Un editor de código es una herramienta de software que permite escribir, modificar y gestionar a los desarrolladores el código fuente de los programas. Recordemos que el código fuente es el conjunto de instrucciones escritas en un lenguaje de programación que un desarrollador crea para construir un software.

Algunas características de un editor de código:

- **Resaltado de sintaxis:** Colorea las palabras clave, funciones y variables de diferentes colores para que el código sea más fácil de leer y entender.
- **Autocompletado:** Sugiere automáticamente las siguientes palabras o funciones que podrías escribir, ahorrándote tiempo y reduciendo errores.
- **Indentación automática:** Organiza el código de forma correcta, facilitando su lectura y mantenimiento. \* La indentación se refiere a la técnica de agregar un espacio inicial (indentado) al principio de las líneas de código, que ayuda a delimitar visualmente los bloques y estructuras de control. Para hacer el indentado se agregan espacios o tabulaciones al principio de las líneas de código y resaltar la jerarquía de las estructuras de control, como bucles, condicionales y funciones.
- **Búsqueda y reemplazo:** Te permite buscar y reemplazar rápidamente texto dentro de tu código.
- **Numeración de líneas:** Cada línea del código tiene un número, lo que facilita encontrar errores o hacer referencias específicas.
- **Personalización:** Puedes personalizar el editor con diferentes temas, fuentes y atajos de teclado para adaptarlo a tus preferencias.



## Veamos 5 editores de código.

### ★ Visual Studio Code



# Visual Studio Code

Es un editor de código (también un ide) gratuito y de código abierto (cualquiera puede acceder al código del software para verlo, modificarlo y distribuirlo a su gusto) que ha ganado mucha popularidad entre los desarrolladores. Es multiplataforma, lo que significa que se puede utilizar en Windows, Linux y Mac. Ofrece una gran cantidad de características útiles como un sistema integrado de control de versiones, depuración incorporada, y la capacidad de agregar extensiones para personalizar el editor según sus necesidades. Además, su interfaz de usuario es limpia y fácil de usar, lo que facilita la edición de código para los desarrolladores de todos los niveles de experiencia. Entre las características notables de Visual Studio Code podemos encontrar *IntelliSense*, una herramienta de resaltado de sintaxis y finalización de palabras que ayuda a escribir código de forma más rápida y sencilla o *Command Palette* para encontrar diferentes comandos para todas tus extensiones.

“Visual Studio” y “Visual Studio Code” no son lo mismo. Visual Studio es un entorno de desarrollo integrado (IDE) y Visual Studio Code es un editor de texto enriquecido como Sublime Text y Atom.



★ Sublime Text



## Sublime Text

Es un editor de código multiplataforma, tiene una interfaz limpia y una funcionalidad increíblemente rica que se puede ampliar mediante plugins. Ofrece una función de «edición en múltiples líneas», que permite a los desarrolladores realizar cambios en varias líneas de código al mismo tiempo, lo que puede aumentar la productividad al trabajar con grandes bases de código. Hay que destacar que no es gratuito para empresas, pero para uso individual la licencia gratuita es de 3 años y luego puedes seguir utilizándolo sin pagar pero sin actualizaciones (yo lo he instalado y no me ha pedido que me suscriba). Es conocido por su rapidez y eficiencia, así como por su capacidad para manejar archivos y proyectos de gran tamaño sin ralentizarse.

★ Notepad ++

Es un editor de texto online gratuito y ligero que admite multitud de lenguajes de programación. Es gratuito y de código abierto. Está específicamente diseñado para Windows, y es conocido por su alta eficiencia y rendimiento, permitiendo la edición de grandes archivos de texto sin ralentizar tu sistema. Se trata de una mejora significativa del Bloc de notas tradicional de Windows y ofrece muchas características que son especialmente útiles para los programadores. Tiene una velocidad de ejecución asombrosa. Es liviano y utiliza menos recursos informáticos, por lo que es adecuado para usuarios con sistemas de gama baja. Una característica notable de este editor de texto es el *mapa del documento*, el cual es excelente para trabajar en un proyecto con largas líneas de códigos. Esta función muestra una descripción general completa de todo el documento, por lo que puedes alternar entre las secciones sin tener que desplazarte hacia arriba y abajo. Simplemente haz clic en una sección en el mapa del documento y el editor saltará automáticamente a esa parte del archivo. En este editor también puedes abrir varios archivos al mismo tiempo para pasar fácilmente de un documento a otro. Asimismo, puedes realizar un análisis comparativo dividiendo la pantalla para varios archivos en una sola ventana. Además, puedes sincronizar ambos documentos y desplazarte en ambos al mismo tiempo activando la función de

sincronizar el desplazamiento vertical. Es *completamente gratuito*. También se ha traducido a más de 80 idiomas que los usuarios pueden editar para adaptarlo a sus necesidades o agregar uno que aún no esté disponible.

★ Atom



Ha sido creado por el equipo de GitHub, se trata de un editor de código open-source con una comunidad activa.

Su diseño es personalizable, permitiendo a los desarrolladores adaptarlo a sus preferencias. Además, Atom ofrece integración con Git y GitHub directamente desde su interfaz, facilitando el seguimiento de cambios.

Actualmente el proyecto se encuentra descontinuado, así que ya no tiene más actualizaciones, pero muchos usuarios se resisten a dejar de utilizarlo. Si vas a empezar a usar un editor de código, es algo que tienes que tener en cuenta.

Los desarrolladores pueden programar juntos utilizando la herramienta Teletype, disponible en versión beta.

Además, este está disponible para la edición multiplataforma, por lo que los usuarios también pueden trabajar en todos los sistemas operativos.



Los usuarios de este editor también pueden optar por la función de división de pantalla para trabajar en varios proyectos a la vez. La interfaz de usuario del editor presentará varios paneles para comparar y editar el código entre archivos.

Atom también viene con una función de navegador de sistema de archivos para navegar y abrir fácilmente un solo archivo, un proyecto completo o varios proyectos en una sola ventana.

Este también ayuda a los usuarios a programar más rápido utilizando las funciones de autocompletar y buscar y reemplazar. Simplemente busca, obtén una vista previa y reemplaza líneas de texto a medida que escribes en un archivo o en todos los proyectos.

Es posible elegir entre miles de paquetes de código abierto para agregar más características y funcionalidades al editor de texto. La mejor parte es que Atom se puede descargar completamente gratis.

#### ★ Emacs



Se trata de un programa clásico para la edición de código por línea de comandos en sistemas GNU/Linux. Es un editor para situaciones en las que debamos editar texto plano. Es un editor de código clásico de GNU, la organización que se encarga de dar



soporte al Kernel de Linux con todo un ecosistema de software libre para completar el sistema operativo. Este editor es por tanto mantenido por la Free Software Foundation.

Actualmente cuenta con versiones para los sistemas operativos Linux, Windows y Mac. Es también conocido por GNU Emacs y nos sirve para editar código de distintas maneras, tanto mediante la consola o terminal como por interfaz gráfica mediante ventanas

Este editor de texto avanzado es más similar a Visual Studio Code que al Bloc de Notas, y funciona tanto en entornos gráficos como en modo texto (de hecho, es famoso por sus largas listas de atajos de teclado, que permiten usar todas sus funciones prescindiendo por completo del mouse).





## Entornos de desarrollo integrado.

### ¿Qué son?

Un entorno de desarrollo integrado, o IDE por sus siglas en inglés (Integrated Development Environment), es una herramienta esencial para programadores. Es una especie de taller completo donde un carpintero tiene todas sus herramientas a mano para crear muebles: sierras, martillos, lijas, etc.

Algunas de las características que debe de tener un ide son:

Combinarlo todo en un solo lugar: En vez de tener varias aplicaciones abiertas para escribir código, compilarlo, depurarlo y más, un IDE te ofrece todas estas funcionalidades en una sola interfaz.

Facilitar la escritura de código: Los IDEs incluyen características como autocompletado, resaltado de sintaxis y refactorización de código, que te ayudan a escribir código más rápido y con menos errores.

Ayudar a depurar tu código: Cuando tu código no funciona como esperas, un IDE te proporciona herramientas para encontrar y solucionar los problemas de manera eficiente.

Integración con otras herramientas: Muchos IDEs se integran con sistemas de control de versiones (como Git), bases de datos y otras herramientas que los desarrolladores utilizan a diario.



## Veamos 5 entornos de desarrollo integrado

### ➤ Visual Studio



Es una de las aplicaciones más conocidas tanto dentro como fuera del mundillo de la programación de aplicaciones- Visual Studio es una aplicación que tradicionalmente siempre ha sido de pago, hasta que, tras la compra de GitHub por parte de Microsoft, lanzó Visual Studio Community, una versión gratuita totalmente funcional de esta aplicación.

La historia de Visual Studio comenzó en 1989, cuando Microsoft lanzó su primer entorno de desarrollo integrado llamado Visual Basic. Esta herramienta se convirtió rápidamente en una de las más populares para el desarrollo de aplicaciones de Windows y estableció los cimientos para lo que se convertiría en Visual Studio.

El primer lanzamiento de Visual Studio fue en 1997 y se denominó Visual Studio 97. Esta versión incluía un entorno de desarrollo integrado para varias plataformas, incluyendo Windows y el lenguaje de programación Visual Basic. Desde entonces, Microsoft ha lanzado regularmente nuevas versiones de Visual Studio, cada una con nuevas características y capacidades significativas para ayudar a los desarrolladores a crear aplicaciones de manera más rápida y eficiente.



Con Visual Studio podemos crear aplicaciones para cualquier plataforma, tanto de escritorio como móvil, crear interfaces de usuario web en C#, editar páginas de ASP.NET y mucho más. Además, también nos permite compilar, depurar errores de funcionamiento y probar aplicaciones de .NET y C++ en entornos Linux.

Una de las funciones más interesantes que nos ofrece Visual Studio es que hace uso de la inteligencia artificial para completar líneas de código en tiempo real. Gracias a ello podremos reducir el tiempo de desarrollo de las aplicaciones, ya que no se basa en predicciones, sino que aprende sobre la marcha que estamos haciendo y a donde queremos llevarlo, lo que nos permite programar de forma más segura y precisa.

Todos los cambios que hagamos en el código de la aplicación quedan almacenados en un registro de cambios, junto con la persona que los ha realizado si estamos trabajando en un proyecto de forma colaborativa. También nos permite analizar el funcionamiento de un código en tiempo real, ideal para comprobar el funcionamiento sin esperar a su compilación.

*Algunas de las **características** más relevantes de Visual Studio son:*

Entorno Integrado: Visual Studio proporciona un entorno de desarrollo integrado que permite a los programadores escribir, depurar y probar el código en un mismo lugar, lo que facilita el proceso de desarrollo y permite ahorrar tiempo.

Lenguajes de Programación: Este software es compatible con múltiples lenguajes de programación, lo que lo hace flexible y adaptable a diferentes necesidades. Algunos de los lenguajes de programación compatibles son C++, C#, F#, Visual Basic y Python.

Integración de Herramientas: Visual Studio cuenta con integración de herramientas de terceros, lo que permite a los desarrolladores utilizar herramientas adicionales para complementar el proceso de desarrollo, como sistemas de control de versiones o pruebas automatizadas.



Herramientas de depuración avanzadas: El IDE ofrece potentes herramientas de depuración que permiten a los desarrolladores rastrear y solucionar problemas en sus aplicaciones de manera eficiente. Incluye características como puntos de interrupción, seguimiento de pila, inspección de variables y análisis de rendimiento para facilitar la depuración y optimización del código.

Integración con Git y control de versiones: Visual Studio ofrece una sólida integración con el sistema de control de versiones Git. Los desarrolladores pueden administrar fácilmente su código fuente, realizar seguimiento de cambios, crear ramas y fusionar código directamente desde el IDE.

Soporte para inteligencia artificial y aprendizaje automático de Visual Studio: Incluye herramientas y extensiones para el desarrollo de aplicaciones basadas en inteligencia artificial y aprendizaje automático. Los desarrolladores pueden crear y entrenar modelos de aprendizaje automático, así como implementar soluciones de IA en sus aplicaciones.

➤ Apache NetBeans





NetBeans es otro entorno de desarrollo integrado basado en código abierto. Gracias a su enfoque podemos desarrollar nuestras aplicaciones como módulos (como un componente de un software). Lo cual resulta sumamente cómodo pues podemos reutilizar nuestro código para diferentes proyectos las veces que queramos sin mayor complicación. NetBeans es un muy buen IDE para Java pero también podemos desarrollar en otros lenguajes como PHP, C, C++ e incluso HTML 5.

Entre sus detractores, hay quien comenta que el desarrollo de una GUI (interfaz gráfica de usuario) sobre NetBeans es algo complicado y que no cuenta con la ingente cantidad de plugins y extensiones.

Apache NetBeans incluye un editor de texto que analiza el código tanto sintáctica como semánticamente, lo que permite evitar errores básicos de programación que nos puede hacer perder muchas horas. Es multiplataforma, con aplicaciones disponibles para todos los sistemas operativos compatibles con Java como Windows, Mac, Linux, OSX y BSD

A través de su página web tenemos a nuestra disposición un gran número de tutoriales tanto básicos como avanzados, con los que saldremos de dudas cuando nos atascamos con alguna función. Además, incluye un servidor de lista de correo donde periódicamente envían información sobre nuevas funciones, tutoriales y demás.

La descarga y el uso de Apache NetBeans es completamente gratuito a través de su web.

*Algunas de sus **características** son:*

Código abierto: NetBeans es de código abierto y por ende gratuito, de modo que cualquiera puede descargarse esta aplicación. Para ello, es necesario visitar la web oficial de Apache NetBeans.

Multiplataforma: Se puede emplear en distintos dispositivos, así como ejecutarse en diferentes sistemas operativos (Mac OS, Windows, Linux y Solaris).



Manejo automático de la memoria: Para aquellos programas con C o C++, la administración de la memoria se puede realizar de forma automática.

Multilenguaje: Como venimos diciendo, NetBeans permite desarrollar aplicaciones multilenguaje. Es decir, no solo opera con Java; sino que NetBeans puede utilizar otros lenguajes como PHP, C o Ruby, entre otros.

Desarrollo de interfaces gráficas de usuario (GUI): NetBeans incluye un diseñador visual que facilita la creación de interfaces gráficas de usuario de forma rápida y sencilla, tanto para aplicaciones de escritorio como web.

Integración con herramientas de desarrollo: NetBeans se integra con una amplia variedad de herramientas de desarrollo, como sistemas de control de versiones como Git,, servidores de aplicaciones, bases de datos, y más.

*\*Como curiosidad comentar que NetBeans y Java pertenecían a la misma compañía, de ahí que este IDE se encuentre orientado principalmente a dicho lenguaje de programación.*

➤ Eclipse



Es un IDE muy popular de código abierto y ampliamente extendido. Fue desarrollado originalmente por IBM aunque actualmente cuenta con una gran comunidad que le da soporte con una gigantesca biblioteca de plugins. Es multiplataforma (Windows, Linux y Mac) y dispone de nuevas versiones cada año. Con Eclipse es posible desarrollar todo

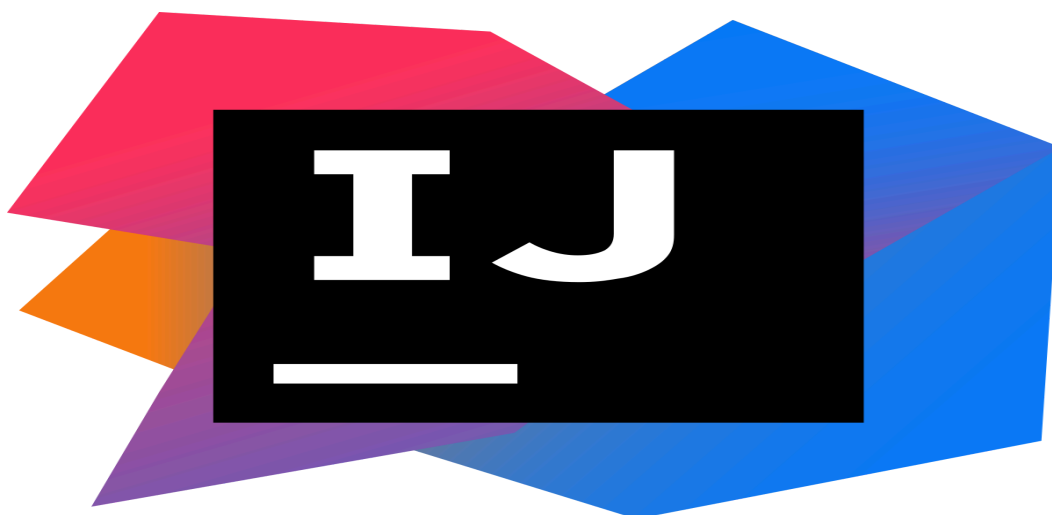


tipo de aplicaciones para web, escritorio o móviles a través de su soporte multi-lenguaje usando Java, C, C++, JSP, perl, Python, Ruby y PHP.

Su mayor punto fuerte reside en su asistente y capacidad para depurar errores. A medida que vas programando Eclipse lanza recomendaciones que nos ayudan a optimizar nuestro código.

Se puede descargar el IDE Eclipse desde su página web donde también podemos descargar un gran número de herramientas adicionales para sacar el máximo partido a Eclipse.

➤ IntelliJ Idea



Desarrollado por Jet Brains es un IDE que cuenta con dos versiones.

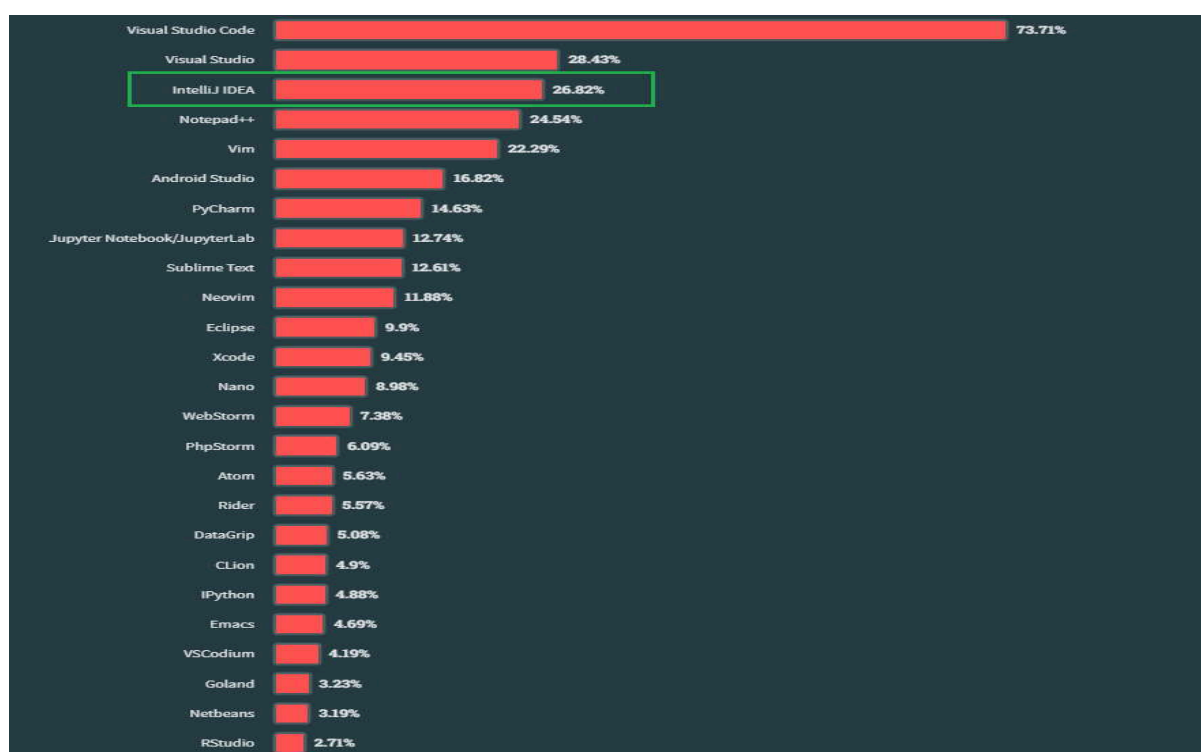
La Community Edition, libre descarga, y la versión Ultimate Edition que cuesta la friolera de más de 500 dólares de suscripción anual, según el plan que contratemos, pero que podemos probar gratis durante 90 días si lo que queremos es echarle un vistazo a todo lo que puede ofrecer.



Con IntelliJ podemos programar sobre diferentes lenguajes y proporciona soporte para trabajar con JAVA , Node JS, PHP, Python, Ruby, Sass, TypeScript, AngularJS, CoffeeScript, CSS, HTML, JavaScript, LESS, etc.

Además ofrece integración con GIT lo que permite trabajar con diferentes versiones de nuestro software sin afectar a la línea maestra de nuestro desarrollo.

La encuesta anual 2023 más importante de desarrolladores de Stackoverflow lo sitúa muy por arriba de otros editores e IDEs específicos de Java famosos, como Eclipse y Netbeans:



### Características de IntelliJ IDEA:

#### Soporte de Lenguajes de Programación

Java: Soporte completo, con análisis de código en tiempo real, refactorización avanzada y depuración.

Kotlin: Desarrollado por JetBrains, tiene una integración excelente.





Otros lenguajes: Groovy, Scala, JavaScript, TypeScript, SQL, y muchos más.

### Análisis de Código y Refactorización

Inspecciones de Código: Detecta errores y problemas en el código mientras se escribe.

Refactorización Inteligente: Incluye renombrar, mover, cambiar firma de método, extraer método, y muchas otras operaciones.

Corrección Rápida: Sugerencias automáticas para corregir errores.

### Navegación y Búsqueda

Navegación Avanzada: Permite saltar rápidamente entre clases, métodos, y archivos.

Búsqueda Eficiente: Herramientas de búsqueda poderosas como "Buscar en Todo el Proyecto".

### Integración con Herramientas y Tecnologías

Control de Versiones: Integración con Git, SVN, Mercurial, y otros sistemas de control de versiones.

Herramientas de Construcción: Soporte para Maven, Gradle, Ant.

Contenedores y Virtualización: Integración con Docker y Kubernetes.

Frameworks: Soporte para Spring, Hibernate, Java EE, Grails, y muchos otros.

### Depuración y Pruebas

Depurador Integrado: Herramientas avanzadas de depuración con capacidades de analizar expresiones, ver valores de variables y más.

Pruebas Unitarias: Soporte integrado para JUnit, TestNG, Spock, y otras frameworks de prueba.

Pruebas de Integración: Soporte para pruebas de integración y funcionales.



### Desarrollo Web y Móvil

Frontend: Soporte para HTML, CSS, JavaScript, TypeScript, React, Angular, Vue.js.

Backend: Soporte para frameworks como Spring Boot, Micronaut, Quarkus.

Desarrollo Móvil: Soporte para Android.

### Productividad y Automatización

Completado de Código Inteligente: Completado de código avanzado que entiende el contexto.

Generación de Código: Generación automática de constructores, getters/setters, y otros patrones de código.

Plantillas de Código: Uso de plantillas personalizables para acelerar la escritura de código.

Macros y Atajos: Grabación de macros y personalización de atajos de teclado.

### Interfaz de Usuario

Interfaz Intuitiva: Personalización de temas, disposición de ventanas y otros aspectos de la interfaz.

Editor Inteligente: Herramientas como plegado de código, múltiples cursores y edición en bloques.

### Extensibilidad

Plugins: Gran ecosistema de plugins para añadir funcionalidades adicionales, desde integración con servicios hasta herramientas específicas de desarrollo.

### Herramientas Integradas

Base de Datos: Herramientas de administración y desarrollo de bases de datos, con soporte para SQL y noSQL.



Terminal: Terminal integrada para ejecutar comandos directamente desde el IDE.

Herramientas de Análisis: Soporte para análisis de rendimiento y perfiles de memoria.

### Versiones y Licencias

Community Edition: Gratuita y de código abierto, adecuada para proyectos básicos.

Ultimate Edition: Licencia paga con características avanzadas para desarrollo profesional, empresarial y web.

#### ➤ BlueJ



Este ide está presente en innumerables cantidades de vídeos y tutoriales en YouTube. La principal razón es que en cierto momento James Gosling, el creador de Java, dijo que BlueJ era uno de sus IDE IDEs favoritos. Esto desencadenó que de pronto en muchas escuelas y academias se usará BlueJ para dar clase pues era el IDE del momento.

Como los IDE anteriores de esta lista, es multiplataforma. Podemos usarlo sobre Windows, macOS y GNU/Linux y cuenta con una gran documentación que lo hace especialmente indicado para estudiantes.

Su principal característica es que nos permite generar una vista UML de todos los objetos que componen nuestra aplicación.



BlueJ fue desarrollado para apoyar la enseñanza y el aprendizaje de la programación orientada a objetos y como resultado, su diseño difiere de otros entornos de desarrollo. La pantalla principal muestra gráficamente la estructura de clases de una aplicación en desarrollo (en un diagrama muy parecido a UML) y los objetos pueden ser creados y probados interactivamente

Combinado con una interfaz de usuario simple, esta facilidad de interacción permite experimentar de manera fácil con los objetos en desarrollo. Los conceptos de la orientación a objetos (clases, objetos, comunicación a través de llamadas a métodos) son representadas visualmente en el diseño de interacción en la interfaz.

#### Principales características:

Representación de orientación a objetos: los conceptos de clases y objetos son presentados en forma visual.

Simplicidad de la interfaz: La interfaz gráfica de usuario es más simple que en ambientes de alta escala profesionales, y por ello más fácil de aprender.

Interacción con objetos: Los programadores son capaces de experimentar con objetos creándose en el banco de objetos («Object Bench») y usando métodos individuales interactivamente (incluyendo paso de parámetros e inspección de resultados).

El «Code Pad»: El code pad es una herramienta que evalúa instantáneamente expresiones y frases arbitrarias escritas en Java.

Testeo de regresión: BlueJ soporta pruebas de regresión gracias a su integración con JUnit. Además de la escritura a mano de clases JUnit classes, se pueden grabar tests interactivos y se pueden crear a partir de ellos casos de prueba JUnit.

Soporte de trabajo en grupo: BlueJ proporciona soporte simple para trabajo grupal vía un subconjunto de funcionalidad CVS y Subversion.

Soporte de Java ME: Los proyectos Java ME (Micro Edition) pueden ser desarrollados e implementados desde BlueJ.



Sistema de extensiones flexibles: Las extensiones ( Plug-Ins) se pueden desarrollar usando una API de extensiones públicas para extender la funcionalidad del entorno básico del programa.

Creación de archivos Jar y applets



## Herramientas online que permiten desarrollar aplicaciones

Aunque parece evidente vamos a ponerle algo de contexto a lo que significa una herramienta online que permite desarrollar aplicaciones.

Una herramienta online que permite desarrollar aplicaciones es como un taller digital donde puedes construir tu propia aplicación, sin necesidad de ser un experto en programación.

Imagina que estás construyendo una casa con bloques de Lego. En lugar de bloques, utilizas componentes prediseñados que arrastras y sueltas para crear la estructura de tu aplicación. Estos componentes pueden ser botones, campos de texto, imágenes, etc. Además, puedes personalizarlos y conectarlos entre sí para crear la funcionalidad que deseas.

Una vez que hemos contextualizado este apartado veamos algunos tipos de herramientas.

### → Plataformas sin código (No-code) y de bajo código (Low-code)

No-code: Son herramientas que permiten crear aplicaciones sin necesidad de escribir una sola línea de código. Utilizan interfaces visuales intuitivas, como arrastrar y soltar, para construir aplicaciones completas. Están diseñadas para usuarios sin conocimientos técnicos.

Low-code: Son similares a las plataformas no-code, pero ofrecen un mayor nivel de personalización. Permiten utilizar algunos elementos de código para adaptar la aplicación a necesidades más específicas. Son ideales para desarrolladores que quieren acelerar el proceso de desarrollo o para usuarios con conocimientos básicos de programación



## Características

Intuitivas: Su interfaz es similar a la de un programa de diseño gráfico, lo que facilita su uso.

Rápidas: Permiten desarrollar aplicaciones en un tiempo mucho menor que con la programación tradicional.

Accesibles: Están diseñadas para personas sin conocimientos técnicos.

## Ejemplos de uso

*Creación de aplicaciones móviles y web simples.*

*Desarrollo de prototipos.*

*Automatización de tareas.*

*Construcción de sitios web dinámicos.*

### Algunas de estas aplicaciones son:

Bubble: Una de las plataformas más conocidas para crear aplicaciones web sin escribir código. Ofrece una interfaz visual intuitiva para diseñar, conectar datos y crear funcionalidades complejas.

Adalo: Similar a Bubble, Adalo te permite construir aplicaciones móviles y web de forma visual, conectando diferentes bases de datos y servicios.

Glide: Especializada en transformar hojas de cálculo de Google Sheets en aplicaciones móviles. Ideal para crear prototipos rápidos o aplicaciones sencillas.

AppGyver: Plataforma low-code que permite crear aplicaciones móviles y web personalizadas, con una amplia gama de componentes y conectores.



### → Plataformas de desarrollo web

Son herramientas poderosas que te permiten crear sitios web sin necesidad de ser un experto en programación. Estas plataformas ofrecen interfaces visuales intuitivas y una amplia gama de funcionalidades para que puedas dar vida a tus ideas.

#### **Existen dos tipos principales:**

Plataformas de construcción de sitios web: Son ideales para crear sitios web simples y rápidos. Ofrecen plantillas prediseñadas y opciones de personalización limitadas. Un ejemplo sería Google Sites.

Plataformas de desarrollo web visual: Ofrecen un mayor nivel de personalización y control sobre el diseño y la funcionalidad de tu sitio web. Son adecuadas para proyectos más complejos.

Algunas **características comunes de las plataformas de desarrollo web** son las siguientes:

Editor visual: Permite arrastrar y soltar elementos para diseñar la interfaz de tu sitio web.

Plantillas prediseñadas: Ofrece una amplia variedad de plantillas para diferentes tipos de sitios web.

Personalización: Permite personalizar el diseño, el contenido y las funcionalidades de tu sitio web.

Integraciones: Se pueden integrar con otras herramientas y servicios, como formularios de contacto, análisis web y pagos en línea.

Hospedaje: Muchas plataformas incluyen el hospedaje web.

CMS (Sistema de gestión de contenidos): Facilita la gestión y actualización del contenido de tu sitio web.





## → Frameworks de desarrollo web

**¿Qué son?** Un framework, o un marco de trabajo, es una plantilla para desarrollar softwares de manera más rápida y eficiente. Solo necesitas escribir el código que se encargue de la lógica de tu app. No tendrás que reinventar la rueda y gastar tiempo en tareas básicas de desarrollo, como crear clases, manejar objetos y definir funciones típicas. Un framework define la estructura de tu futuro proyecto y proporciona las herramientas necesarias que puedes usar como bloques de construcción. Puede incluir:

Bibliotecas y módulos para tareas específicas. Se encargan de funciones como manipulación de datos, gestión de sesiones, seguridad, autenticación, etc.

Patrones de diseño recomendados para estructurar el código. Sirven para ejecutar operaciones típicas de desarrollo y crear una arquitectura coherente en la aplicación.

Herramientas de desarrollo. Permiten generar y depurar el código, realizar pruebas unitarias, manejar bases de datos, etc.

Estándares de codificación. Aseguran que los archivos y directorios se organicen de la misma manera, mientras que el código se mantenga consistente y legible. Además, establecen el formato unificado para los nombres de las variables.

El objetivo principal de un framework es ahorrar tiempo y esfuerzo a los desarrolladores. Los frameworks permiten aprovechar soluciones existentes eliminando la necesidad de escribir código desde cero para funciones comunes y repetitivas. Esto agiliza el desarrollo, ayuda a minimizar errores y a crear un código más limpio. Además, los frameworks son reutilizables y pueden ser aplicados en múltiples proyectos



## Ventajas y desventajas de los frameworks

### *Ventajas*

- ☐ Los frameworks proporcionan una estructura clara con una funcionalidad predefinida, lo que permite reducir los costos de desarrollo y lanzar apps rápidamente.
- ☐ Mantienen un estilo de codificación coherente en todo el proyecto. Esto facilita la colaboración entre desarrolladores y permite mantener el código a largo plazo.
- ☐ Gracias a componentes y bibliotecas reutilizables, los desarrolladores pueden usar la base de código y hacer modificaciones sencillas para adaptarla a su proyecto.
- ☐ Muchos frameworks implementan actualizaciones de seguridad y mejoras continuas.
- ☐ Los marcos de trabajo populares suelen contar con una comunidad activa de programadores, que comparten conocimientos, brindan soporte y crean recursos útiles, como documentación, tutoriales y ejemplos de uso

### *Desventajas*

- ☐ Estudiar la estructura, convenciones y características de cada framework toma mucho tiempo. Más aún si eres principiante.
- ☐ La arquitectura predefinida impone ciertas restricciones. Es difícil personalizar una aplicación si el proceso de desarrollo no es flexible.
- ☐ Algunos frameworks incluyen una gran cantidad de funcionalidades y componentes que pueden resultar innecesarios si tienes una app simple. La sobrecarga de estos elementos lleva a un mayor consumo de recursos.



## Tipos y ejemplos de frameworks

Los tipos de frameworks varían según el propósito, el lenguaje de programación y el área de aplicación. Algunos de los frameworks más populares son:

### Apps móviles

*React Native* (JavaScript), permite desarrollar aplicaciones móviles nativas para iOS y Android con una base de código común.

*Flutter* (Dart), es un framework de código abierto elaborado por Google. Ayuda a crear interfaces de usuario en aplicaciones para iOS y Android.

### Aplicaciones de escritorio

*Electron* (JavaScript), usa tecnologías web como HTML, CSS y JavaScript para construir aplicaciones nativas para Windows, macOS y Linux.

*Qt* (C++), permite crear aplicaciones multiplataforma para escritorio y dispositivos móviles. Además, ofrece herramientas y bibliotecas para diseñar interfaces gráficas de usuario.

### Videojuegos

*Unity* (C#), permite desarrollar juegos 2D y 3D para una variedad de plataformas, incluyendo PCs, consolas y dispositivos móviles.

*Unreal Engine* (C++), es un framework avanzado para elaborar juegos realistas de alta calidad.

### Aplicaciones empresariales

*Spring* (Java), ofrece módulos de funcionalidad diversa, como gestión de bases de datos, acceso remoto a servidores, autorización y autenticación.

*ASP.NET* (C#), está elaborado por Microsoft para construir aplicaciones empresariales en la plataforma .NET.



### Aplicaciones web

*Django* (Python), es un framework de alto nivel que sigue el patrón de diseño Modelo-Vista-Controlador (MVC).

*Ruby on Rails* (Ruby), promueve el principio de convención sobre la configuración. Es cuando el desarrollo se basa en las convenciones de programación establecidas en vez de las configuraciones definidas por el programador.

*Laravel* (PHP), cuenta con una sintaxis simple y una gran cantidad de herramientas para el desarrollo de aplicaciones web modernas.

*Angular* (JavaScript), está desarrollado por Google para crear aplicaciones web de una sola página (SPA).

*Vue.js* (JavaScript), se orienta a la construcción de interfaces interactivas y reutilizables.

### **Los frameworks de desarrollo web también se puede dividir en dos grupos:**

Back End. Se aplican para desarrollar aplicaciones web del lado del servidor con facilidad. Permiten crear formularios y páginas, así como controlar bases de datos y peticiones HTTP. Usan lenguajes de programación como Python, Ruby, PHP, Javascript, Java, etc. Node.js y Flask son unos ejemplos de estos frameworks.

Front End. Están diseñados para agilizar el desarrollo, aumentar la funcionalidad y la interactividad de las aplicaciones web del lado del cliente. Se componen con JavaScript, HTML y CSS. Entre los frameworks de Front End más populares están Bootstrap y jQuery.



### **Bibliografía utilizada:**

[7 Mejores Editores de Código para tu Trabajo de Desarrollo ✓](#)

[Los 17 mejores editores de código](#)

[Emacs, el editor de texto libre con vocación de sistema operativo: sus 'extensiones' más usadas suplen toda clase de aplicaciones](#)

<https://medium.com/@orlidev/los-mejores-entornos-de-desarrollo-integrado-ide-explicados-con-la-magia-de-escandalosos-587e8a048eef>

[Mejores entornos de desarrollo, IDE, para programar en cualquier lenguaje](#)

[Qué es Visual Studio | Definición, historia y ventajas](#)

[Novedades y características destacadas en Visual Studio 2022](#)

[Los IDE más usados en programación JAVA | ifGeekThenNTTDATA](#)

[¿Qué es NetBeans? Ventajas y usos - IMMUNE Institute](#)

[1 - IDE IntelliJ IDEA](#)

[BlueJ un IDE para aprender Java de manera interactiva y visual](#)

[Frameworks: Marcos de trabajo para programadores](#)