



Curso 2023/24

Tema-02 (2ªEval)

Instalación y uso de Entornos de Desarrollo Integrado (IDEs).

Tema-02(2ªEval)

Contenido

1.- CONCEPTO DE ENTORNOS DE DESARROLLO. EVOLUCIÓN HISTÓRICA:	2
1.1.- ¿Qué es un Entorno de Desarrollo Integrado (IDE)?	2
1.2.- Evolución Histórica.	2
2.- COMPONENTES DE UN IDE:	3
3.- PRINCIPALES ENTORNOS DE DESARROLLO DE ESCRITORIO:	4
3.1.- Code::Blocks.	4
3.2.- Visual Studio Code.	4
3.3.- Visual Studio.	5
3.4.- Eclipse.	5
3.5.- NetBeans.	6
3.6.- IntelliJ IDEA.	6
4.- ENTORNOS DE DESARROLLO DE MÓVILES Y ONLINE:	7
4.1.- IDE Desarrollo móvil.	7
4.2.- IDE Online (En la nube o Web).	7
5.- VENTAJAS E DESVENTAJAS DE LOS IDE:	8
5.1.- Ventajas.	8
5.2.- Desventajas.....	8
6.- INSTALACIÓN DE LOS PRINCIPALES ENTORNOS DE DESARROLLO:	9
6.1.- Entornos de desarrollo para C/C++ y Web.	9
6.2.- Entornos de desarrollo para Java.	9
7.- COMPARATIVA DE LOS PRINCIPALES ENTORNOS DE DESARROLLO:	10
8.- BIBLIOGRAFÍA:	11

1.- CONCEPTO DE ENTORNOS DE DESARROLLO. EVOLUCIÓN HISTÓRICA:

1.1.- ¿Qué es un Entorno de Desarrollo Integrado (IDE)?.

Un **Entorno de Desarrollo Integrado - IDE (Integrated Development Environment)** es una aplicación informática que está compuesta por un conjunto de **procedimientos y herramientas** de programación que van a **facilitar la tarea del desarrollo** de aplicaciones al programador obteniendo resultados con mayor rapidez.

Integra herramientas de **compilación, ejecución y testeo** de aplicaciones pudiendo estar pensada para **utilizarse con un único lenguaje** de programación **o bien** puede dar cabida a **varios lenguajes**. Posee un **Interfaz de Usuario (UI)** que facilita su uso. Además, permite la instalación de **extensiones (plugins)** que añaden nuevas funcionalidades.

1.2.- Evolución Histórica.

En las **décadas de utilización de la tarjeta perforada** como **sistema de almacenamiento** el concepto de Entorno de Desarrollo Integrado sencillamente no tenía sentido.

Los **programas** estaban **escritos con diagramas de flujo** y **entraban** al sistema **a través de** las **tarjetas perforadas**. **Posteriormente, eran compilados**.

El **primer lenguaje** de programación que **utiliza un IDE fue el BASIC** (que fue el primero en abandonar también las tarjetas perforadas o las cintas de papel).

Éste primer IDE **estaba basado en consola** de comandos exclusivamente (normal por otro lado, si tenemos en cuenta que hasta la década de los 90 no entran en el mercado los sistemas operativos con interfaz gráfica). Sin embargo, el uso que hace de la gestión de archivos, compilación, depuración... es perfectamente compatible con los IDE actuales.

A nivel popular, el **primer IDE** puede considerarse que fue el **IDE llamado Maestro**. Nació a principios de los 70 y fue instalado por unos 22.000 programadores en todo el mundo. Lideró el campo durante los años 70 y 80.

El uso de los entornos integrados de desarrollo **se ratifica y afianza en los 90** y hoy en día contamos con infinitud de IDE, tanto de licencia libre como no. El uso de un IDE u otro depende de la elección del usuario. Podemos decir que los más utilizados actualmente son los que aparecen en la siguiente tabla:

Tabla de los IDE más relevantes hoy en día:		
Entorno de desarrollo	Lenguajes que soporta	Tipo de licencia
NetBeans	C/C++, Java, JavaScript, PHP, Python.	De uso público.
Eclipse	Ada, C/C++, Java, JavaScript, PHP.	De uso público.
Microsoft Visual Studio.	Basic, C/C++, C#.	Propietario.
C++ Builder.	C/C++.	Propietario.
JBuilder.	Java.	Propietario.

2.- COMPONENTES DE UN IDE:

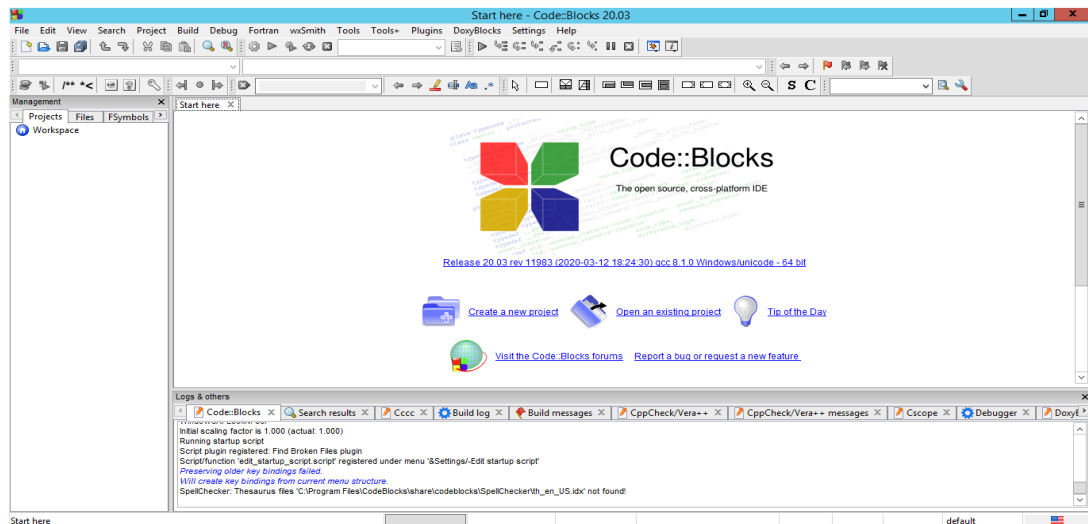
Un **IDE** normalmente **consiste en un editor de texto** donde escribir el código con resaltado de sintaxis y corrector sintáctico, **un compilador y/o intérprete**, **un depurador**, **un control de versiones**, **un constructor de interfaz gráfica (GUI)**, entre otras funcionalidades.

- **Editor de texto:** es la parte que nos permite escribir el código fuente del programa, ofrece funciones propias de la edición como copiar, cortar, pegar o buscar. Además, es capaz de reconocer, resaltar y cambiar los colores de las variables, las cadenas de caracteres, las palabras reservadas, las instrucciones, etc. haciendo que el código fuente sea mucho más visual, cómodo y se podrán reconocer los errores a simple vista.
- **Compilador:** Es el encargado de traducir el código fuente tecleado por el programador y escrito en un lenguaje de alto nivel a un programa escrito en lenguaje de bajo nivel llamado lenguaje máquina, capaz de ser interpretado y ejecutado por la máquina. El proceso de traducción se conoce como compilación.
- **Intérprete:** Se diferencian de los anteriores en que realizan la traducción a medida que se van ejecutando las instrucciones. Normalmente, no guardan el resultado de dicha traducción. Son más lentos que los compiladores debido a la necesidad de traducir el programa mientras se ejecutan, pero a cambio, son más flexibles como entornos de programación y depuración y permiten ofrecer al programa interpretado un entorno no dependiente de la máquina donde se ejecuta, sino del propio intérprete (lo que se conoce comúnmente como máquina virtual).
- **Depurador (Debugger):** Es el encargado de depurar y limpiar los errores en el código fuente de un programa informático. Permite examinar paso a paso, instrucción a instrucción la ejecución de un programa y examinar las distintas situaciones y cambios que se produzcan en las variables del programa o en los registros del procesador. El depurador va a permitir detener el programa en cualquier punto de ruptura para examinar la ejecución.
- **Constructor de interfaz gráfica:** Esta herramienta de programación simplifica la creación de interfaces gráficas de usuario permitiendo al diseñador colocar los controles (botones, listas, menús, etc.), utilizando un editor WYSIWYG de arrastrar y soltar. Algunos IDE incorporan estas herramientas con el plugin correspondiente, como es el caso de Eclipse.
- **Control de versiones:** Permiten controlar los cambios que se realizan sobre las aplicaciones. De esta manera, se obtendrán revisiones y versiones de las aplicaciones en un momento dado de su desarrollo.

3.- PRINCIPALES ENTORNOS DE DESARROLLO DE ESCRITORIO:

3.1.- Code::Blocks.

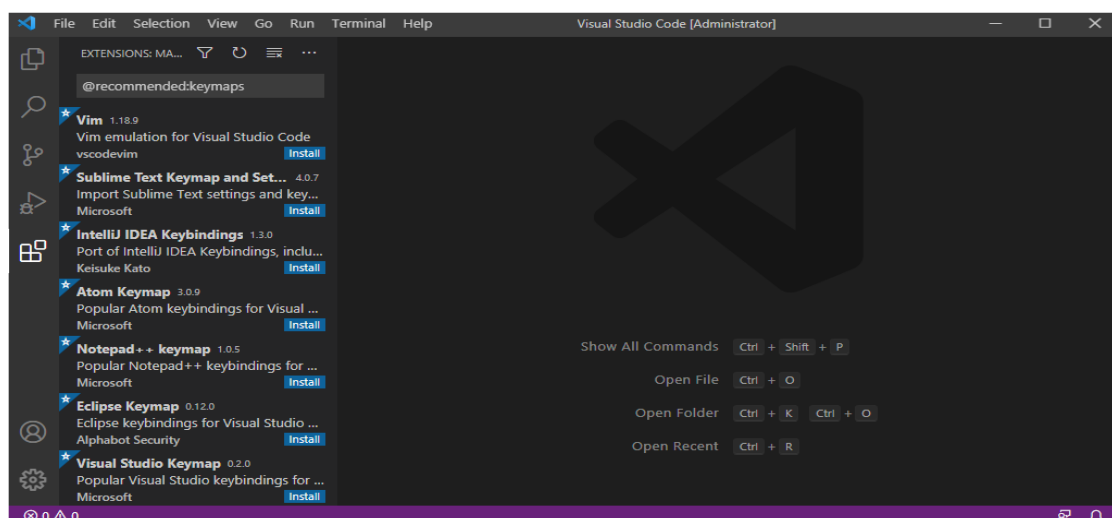
- Es un IDE de código abierto orientado a C/C++ y Fortran.
- Es un entorno totalmente configurable, orientado a satisfacer las necesidades más exigentes de los usuarios.
- Soporta múltiples compiladores, sobre todo GCC, (Colección de Compiladores de GNU).
- Soporte multiplataforma, (funciona con varios sistemas operativos o dispositivos).



Obtener Code::Blocks: [Enlace](#)

3.2.- Visual Studio Code.

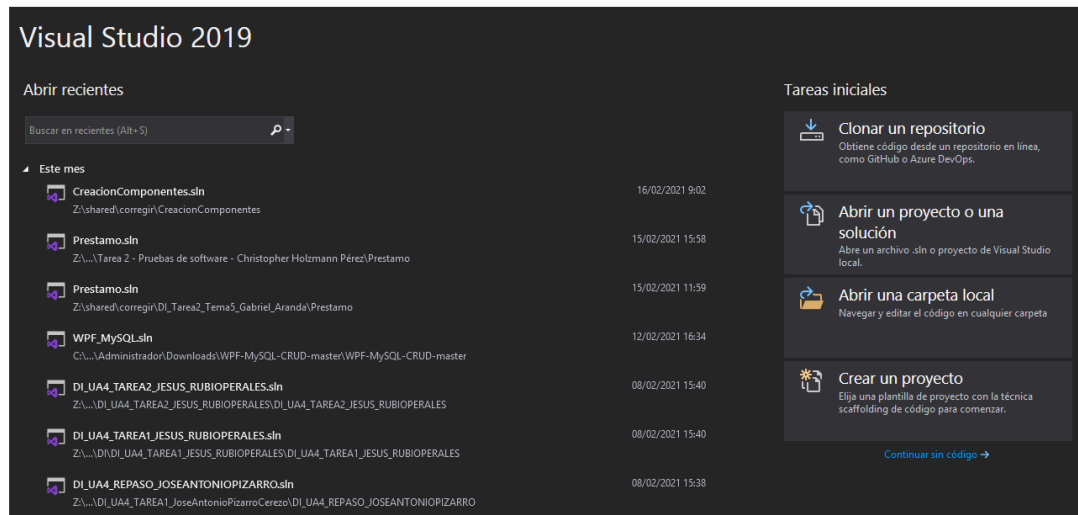
- IDE desarrollado por Microsoft para Windows, Linux y Mac OS.
- Es un IDE para desarrolladores web y JavaScript, con extensiones admite casi todos los lenguajes.
- Posee soporte nativo para GIT.
- Altamente personalizable (extensiones).
- Soporte multilenguaje.



Obtener Visual Studio Code: [Enlace](#)

3.3.- Visual Studio.

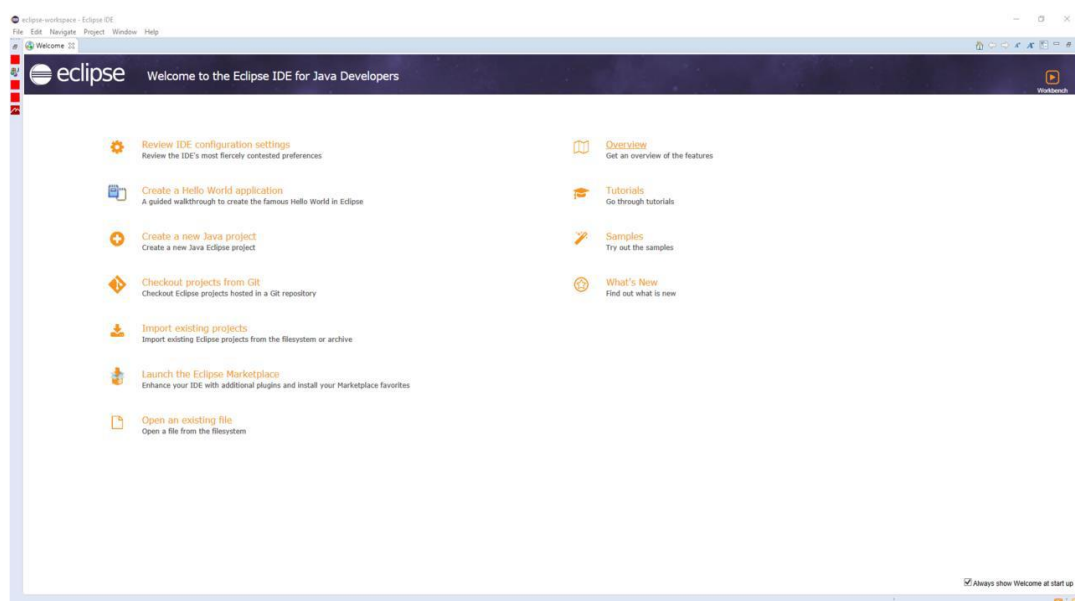
- El “hermano mayor” de Visual Studio Code.
- Posee soporte multilenguaje (PHP, C, C++, C#...).
- Únicamente funciona en entornos Windows (no compensa virtualizar).
- Existe una versión Community gratuita, (es necesario tener una cuenta registrada).



Obtener Visual Studio: [Enlace](#)

3.4.- Eclipse.

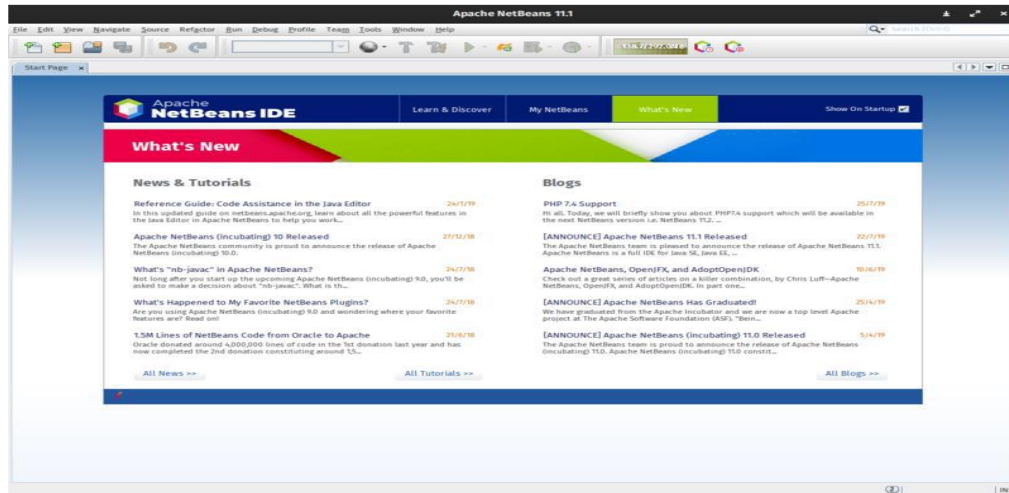
- Creado originalmente por IBM y ahora mantenido por Eclipse Foundation.
- Es un entorno gratuito de código libre.
- ES el IDE más utilizado para el desarrollo de aplicaciones Java.
- Permite añadir nuevas funcionalidades con extensiones.



Obtener Eclipse: [Enlace](#)

3.5.- NetBeans.

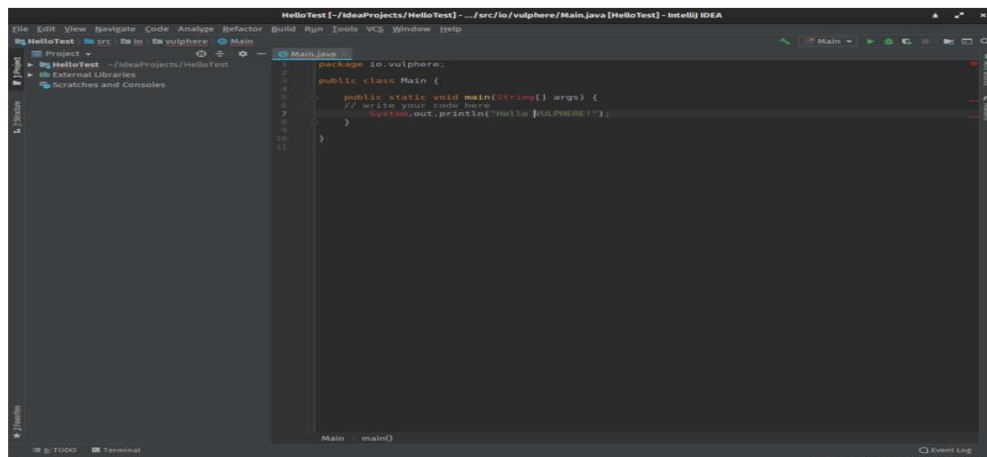
- Enfocado mayoritariamente al desarrollo de aplicaciones Java.
- También es personalizable, al igual que Eclipse.
- Es un entorno mucho más pesado que Eclipse.
- Permite añadir nuevas funcionalidades con extensiones.



Obtener NetBeans: [Enlace](#)

3.6.- IntelliJ IDEA.

- Desarrollado por JetBrains.
- Este IDE sirve para los siguientes lenguajes: JAVA, con Node JS, PHP, Python, Ruby, Sass, TypeScript, AngularJS, CoffeeScript, CSS, HTML, JavaScript, LESS, etc. .
- Posee una versión para la comunidad (community) y una versión de pago por uso.
- Soporte multiplataforma, (funciona con varios sistemas operativos o dispositivos).
- Soporte multilenguaje y soporta múltiples frameworks de programación.
- Permite añadir nuevas funcionalidades con extensiones.



Obtener IntelliJ IDEA: [Enlace](#)

4.- ENTORNOS DE DESARROLLO DE MÓVILES Y ONLINE:

4.1.- IDE Desarrollo móvil.

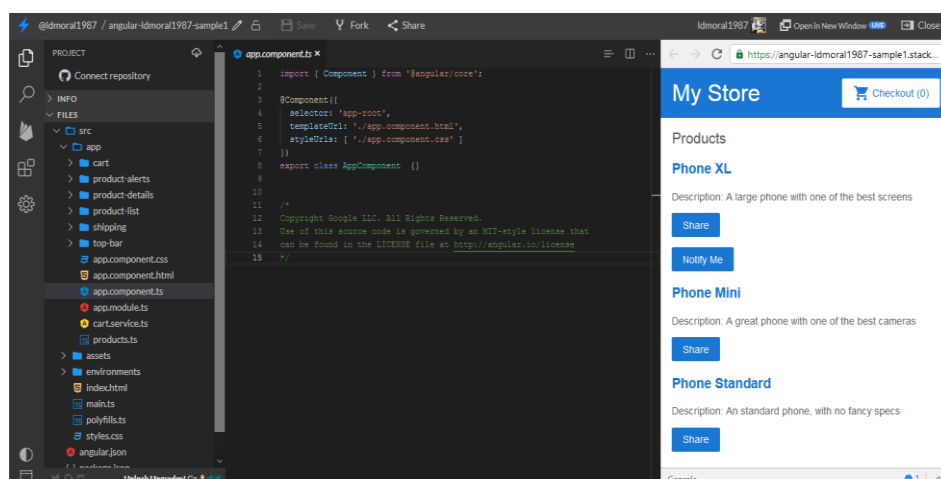
IDE diseñados específicamente para crear e implementar aplicaciones móviles. Algunos IDE están totalmente centrados en dispositivos móviles, mientras que otros permiten a los usuarios agregar complementos específicos para ampliar la funcionalidad al desarrollo móvil.

- Algunos ejemplos: Android Studio, AppCode, Qt Creator, Programming y Xcode.

4.2.- IDE Online (En la nube o Web).

IDE a los que pueden acceder los usuarios autorizados desde cualquier lugar del mundo mediante una conexión a Internet estándar. Los desarrolladores utilizan IDE en la nube para escribir, editar y compilar código directamente en el navegador para prescindir de la necesidad de descargar software en sus equipos locales.

- Poseen casi las mismas funcionalidades de un IDE convencional.
- Fomentan el trabajo colaborativo y los repositorios comunes.
- Muchas empresas están apostando por usar este tipo de entornos.
- La mayoría de estos entornos tienen un estilo parecido a Visual Studio Code.
- Casi todos tienen planes de precio gratuitos.
- Algunos ejemplos: StackBlitz , Repl.it , Codeanywhere, AWS Cloud9 o CodePen.



5.- VENTAJAS E DESVENTAJAS DE LOS IDE:

5.1.- Ventajas.

- Disponer de una interfaz de usuario integrada con todas las herramientas de desarrollo necesarias.
- Facilitar la escritura de código, gracias a características como la edición automática del código fuente, la finalización de código inteligente, la verificación de la sintaxis y el resaltado de sintaxis (a través de negritas, itálicas o diferentes colores de fuente que hacen el código más legible).
- Proporcionar herramientas para depurar errores y probar el código. Cualquier entorno de desarrollo integrado permite seguir el código, línea por línea, conforme se pone en marcha para inspeccionar su comportamiento.
- Simplificar el proceso de compilación. El entorno de desarrollo es capaz de convertir el código en un lenguaje simplificado que los sistemas operativos puedan entender.
- Integrar otras funcionalidades, como sistemas de control de versiones o herramientas de gestión de proyectos.
- Simplifica la creación de aplicaciones de bases de datos.
- No requiere un conocimiento detallado de la base de datos.

5.2.- Desventajas.

- IDE utiliza una interfaz gráfica, por lo que requiere más memoria y potencia de procesamiento.
- Algunos IDE no permiten a los usuarios interactuar con la base de datos directamente.

6.- INSTALACIÓN DE LOS PRINCIPALES ENTORNOS DE DESARROLLO:

6.1.- Entornos de desarrollo para C/C++ y Web.

- Se recomienda usar **Visual Studio Code** para programar en C/C++ y Web.
- Proceso de instalación de **Visual Studio Code**:
 1. Acceder a la web de **Visual Studio Code** → [Enlace](#).
 2. Descargar la última versión con el sistema operativo que desea instalar.
 3. Ejecutar el instalador, con las opciones por defecto.
 4. Iniciar por primera vez el programa para terminar de configurarlo.
 5. Añade las extensiones (plugins) que desee.

6.2.- Entornos de desarrollo para Java.

- Se recomienda usar **Eclipse o Netbeans** para programar en Java.
- Proceso de instalación de **Eclipse**:
 1. Antes de instalar Eclipse, tiene que bajar e instalar el kit de desarrollo de Java (JDK).
 2. Acceder a la web de **Eclipse** → [Enlace](#).
 3. Descargar la última versión con el sistema operativo que desea instalar.
 4. Ejecutar el instalador, elige **Eclipse IDE for Developers** con las opciones por defecto.
 5. Iniciar por primera vez el programa para terminar de configurarlo.
 6. Añade las extensiones (plugins) que desee.
- Proceso de instalación de **Netbeans**:
 1. Antes de instalar Netbeans, tiene que bajar e instalar el kit de desarrollo de Java (JDK).
 2. Acceder a la web de **Netbeans** → [Enlace](#).
 3. Descargar la última versión con el sistema operativo que desea instalar.
 4. Ejecutar el instalador, con las opciones por defecto.
 5. Iniciar por primera vez el programa para terminar de configurarlo.
 6. Añade las extensiones (plugins) que desee.

7.- COMPARATIVA DE LOS PRINCIPALES ENTORNOS DE DESARROLLO:

Lenguaje	¿Qué es?	Ventajas	Desventaja	Sistema O.
HTML	Lenguaje utilizado para la creación de páginas web. HTML significa "hypertext mark-up language", que significa, "lenguaje para el formato de documentos de hipertexto",	Sencillo que permite describir hipertexto. Archivos pequeños. Lenguaje de fácil aprendizaje. Lo admiten todos los exploradores	Lenguaje estático. La interpretación de cada navegador puede ser diferente. Lento para ejecutar archivos pesados	Se pueden ver en todos los sistemas operativos, pero la interpretación varía según el navegador que esté utilizando
Scratch	Es una aplicación informática destinada principalmente a los niños y les permite explorar y experimentar, mediante el uso de una sencilla interfaz	Es un programa gratuito, de software libre. Es perfecto para enseñar y aprender a programar. Es multilenguaje	Es un programa muy pesado y es un poco complicado manejarlo	Está disponible para varios sistemas operativos, Windows, Mac y Linux.
Java	Es un lenguaje orientado a objetos, de una plataforma independiente, fue desarrollado por la compañía SUN Microsystems ahora es propietario ORACLE.	Se pueden realizar distintos aplicativos Puede desarrollar aplicaciones de escritorio que se ejecutan en forma independiente Soporta el desarrollo de aplicaciones móviles	Esperar la actualización siguiente para que sea más rápido.	Sirve para todos los sistemas operativos y si no es la versión adecuada para dicho sistema, la aplicación java se encarga de descargas o actualizar para un excelente desempeño en el pc.
SQL	Lenguaje de consulta estructurada; es un lenguaje específico del dominio que da acceso a un sistema de gestión de bases de datos relacionales que permite especificar diversos tipos de operaciones en ellos.	Es un sistema de gestión de base de datos. Es útil para manejar y obtener datos de la red de redes. Nos permite olvidarnos de los ficheros que forman la base de datos.	Utiliza mucho la memoria RAM para las instalaciones y utilización de software. No se puede utilizar como prácticas porque se prohíben muchas cosas, tiene restricciones en lo particular. La relación, calidad y el precio esta muy debajo comparado con oracle	solo está disponible para sistemas operativos Windows de Microsoft
JavaScript	Es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas.	Lenguaje de scripting seguro y fiable. Los script tienen capacidades limitadas, por razones de seguridad. El código Javascript se ejecuta en el cliente.	Código visible por cualquier usuario. El código debe descargarse completamente. Puede poner en riesgo la seguridad del sitio	Se pueden ver en todos los sistemas operativos, pero la interpretación varía según el navegador que esté utilizando
C	Es un lenguaje orientado a la implementación de Sistemas Operativos, concretamente Unix es el lenguaje de programación más popular para crear software de sistemas, aunque también se utiliza para crear aplicaciones.	El lenguaje C ha sido pensado para ser altamente transportable Acceso a memoria de bajo nivel mediante el uso de punteros. Interrupciones al procesador con uniones. Un conjunto reducido de palabras clave.	Encapsulación. Funciones anidadas, aunque GCC tiene esta característica como extensión. Soporte nativo para programación multihilo y redes de computadores.	Se pueden ver en todos los sistemas operativos, pero la interpretación varía según el navegador que esté utilizando
CSS	Es el lenguaje utilizado para describir la presentación de documentos HTML o XML, esto incluye varios lenguajes basados en XML como son XHTML o SVG	El lenguaje CSS ofrece una amplia gama de herramientas de composición más potentes que HTML. Con CSS se evita tener que recurrir a "trucos" para conseguir algunos efectos	Si hay problemas o limitaciones de compatibilidades, el navegador aplicará el formato predeterminado y nuestro trabajo de composición habrá sido inútil.	Para todos los sistemas operativos Algo que suele ocurrir mucho, es que el diseñador elija una fuente que no suele ser estándar para sus diseños
PHP	PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas	Es un lenguaje multiplataforma. Orientado al desarrollo de aplicaciones web	Como es un lenguaje que se interpreta en ejecución La ofuscación es una técnica que puede dificultar la lectura del código	Se usa principalmente para la interpretación del lado del servidor, páginas web y CMS
Maquina	Es el sistema de códigos directamente interpretable por un circuito microprogramable, como el microprocesador de una computadora o el microcontrolador de un autómata.	Posibilidad de cargar (transferir un programa a la memoria) sin necesidad de traducción posterior, lo que supone una velocidad de ejecución superior a cualquier otro lenguaje de programación.	Dificultad y lentitud en la codificación. Poca fiabilidad. Gran dificultad para verificar y poner a punto los programas	Para todo tipo de maquina ya que este va integrado en el microprocesador
Ensamblador	Lenguaje de programación de bajo nivel. Consiste en un conjunto de mnemónicos que representan instrucciones básicas para los computadores, microprocesadores, microcontroladores y otros circuitos integrados programables.	1.Velocidad 2.Eficiencia de tamaño 3.Flexibilidad	1.Tiempo de programación 2.Programas fuente grandes 3.Peligro de afectar recursos inesperadamente	Para todo tipo de maquina ya que este va integrado en el microprocesador

8.- BIBLIOGRAFÍA:

- ❖ **Libro de Texto de Entornos de desarrollo Ed. Paraninfo**
- ❖ **Cursos de entornos de desarrollo de la web fp-informática**
- ❖ **Wikipedia**