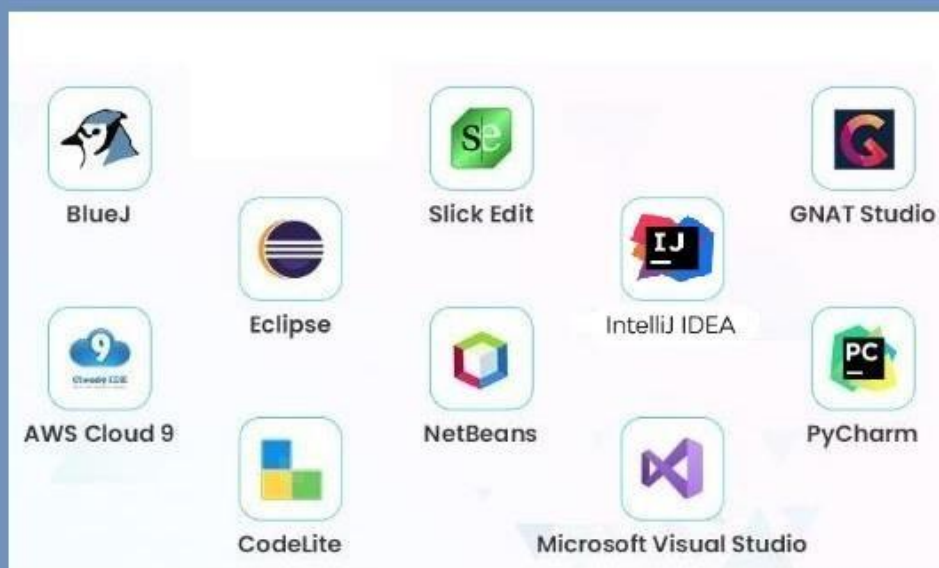


IDEs



Nombre y apellidos: Cristina Sandoval Laborde

Curso: 1º CFGS

Asignatura: Entornos de desarrollo

Índice

1.Introducción.....	1
1.1. Características comunes de los IDE	1
1.2. Ventajas generales de usar un IDE	1
2.Lenguajes de programación más usados.....	2
3.IDEs	2
3.1. IntelliJ IDEA.....	2
3.2. Eclipse.....	3
3.3 NetBeans.....	3
4. Ventajas y desventajas de los 3 IDEs	4
4.1. IntelliJ IDEA-Ventajas.....	4
4.2. IntelliJ IDEA-Desventajas	5
4.3. Eclipse-Ventajas.....	5
4.4 Eclipse - Desventajas	6
4.5 NetBeans - Ventajas.....	6
4.6 NetBeans – Desventajas	7
5.Instalación de IntelliJ IDEA	7
5.1. Descarga e instalación	7
5.2. Configuración inicial.....	9
5.3 Creación de un nuevo proyecto	9
5.3.1 Creación de un proyecto Java	9
5.3.2 Estructura del proyecto.....	10
6. Problemas que nos encontramos en la instalación.	11
7.Soluciones aportadas.....	11
8.Webgrafía	11

1.Introducción

Un Entorno de Desarrollo Integrado (IDE, por sus siglas en inglés) es un sistema de software que combina diversas herramientas de desarrollo esenciales en una sola interfaz gráfica, lo que facilita la creación de aplicaciones y programas de software. Su propósito principal es integrar las funcionalidades que un programador necesita para escribir, depurar, compilar y gestionar su código en un solo lugar, optimizando la productividad y reduciendo los errores durante el proceso de desarrollo.

1.1. Características comunes de los IDE

- Editor de código: Permite escribir y modificar código fuente, con características como resaltado de sintaxis, autocompletado y chequeo de errores en tiempo real. Esta funcionalidad ayuda a los desarrolladores a escribir código más rápido y con menos errores.
- Depurador: Herramienta que permite identificar y corregir errores (bugs) en el código. El depurador permite ejecutar el programa paso a paso, observar el estado de las variables y encontrar fallos en la ejecución del código.
- Compilador: Convierte el código fuente, generalmente escrito en un lenguaje de alto nivel, en código ejecutable o binario que pueda ser entendido por la computadora.
- Automatización de tareas: Los IDE a menudo permiten automatizar tareas como la compilación y las pruebas del código, lo que facilita la creación de versiones y el seguimiento de los cambios sin tener que hacer cada paso manualmente.
- Gestión de proyectos: Los IDE también proporcionan herramientas para gestionar archivos y recursos relacionados con un proyecto, facilitando la organización y el trabajo en equipo, especialmente cuando se trabajan con grandes cantidades de código.
- Control de versiones: Los entornos de desarrollo modernos incluyen la integración con sistemas de control de versiones como Git, lo que permite a los equipos de desarrollo trabajar de manera colaborativa y rastrear los cambios realizados en el código a lo largo del tiempo.

1.2. Ventajas generales de usar un IDE

-Mayor productividad: Al integrar todas las herramientas en un solo lugar, los IDEs eliminan la necesidad de que los desarrolladores cambien entre diferentes aplicaciones, lo que mejora la eficiencia.

-Menos errores: Al detectar errores en tiempo real y ofrecer depuradores visuales, los IDEs ayudan a reducir los errores en el código.

-Fácil aprendizaje: Para los nuevos desarrolladores, un IDE proporciona un entorno estructurado que facilita el aprendizaje de las herramientas y las buenas prácticas de programación.

2.Lenguajes de programación más usados

-Python: Usado en desarrollo web, inteligencia artificial, ciencia de datos y automatización. IDEs como PyCharm y Visual Studio Code son ideales para proyectos con frameworks como Django, Flask o bibliotecas de IA como TensorFlow.

-Java: Principalmente para aplicaciones empresariales, móviles (Android) y software de alto rendimiento. IntelliJ IDEA y Eclipse son populares debido a sus herramientas avanzadas para este lenguaje.

-JavaScript: Esencial para el desarrollo web (front-end y back-end con Node.js). WebStorm y Visual Studio Code son los IDEs más utilizados.

-C#: Usado para aplicaciones de Windows, desarrollo móvil con Xamarin y videojuegos en Unity. Visual Studio es el IDE más común para trabajar con C# y el marco .NET.

-PHP: Fundamental para aplicaciones web y sistemas de gestión de contenido. PHPStorm ofrece un excelente soporte para frameworks como Laravel y Symfony.

-C/C++: Usados en aplicaciones de alto rendimiento, como videojuegos y sistemas operativos. Code: Blocks y Eclipse son los IDEs preferidos, ideales para proyectos que requieren control detallado sobre el hardware y el rendimiento.

El lenguaje de programación que yo he elegido es JAVA, y los IDEs son IntelliJ IDEA, Eclipse y NetBeans.

3.IDEs

3.1. IntelliJ IDEA

Desarrollado por JetBrains, es un IDE destacado para el desarrollo de software en Java, que ofrece una amplia gama de herramientas para mejorar la productividad, como asistencia de código inteligente, análisis estático en tiempo real y refactorización automatizada. Integra herramientas como Maven, Gradle y Ant, y soporta tecnologías como Java EE, Spring y Android. Además, su potente sistema de depuración y análisis ayuda a detectar y corregir errores rápidamente, lo que optimiza el flujo de trabajo de desarrollo.

Algunas de las funcionalidades clave de IntelliJ IDEA incluyen:

- **Coding Assistance**: IntelliJ IDEA ofrece una asistencia de código inteligente que mejora la productividad del programador. Proporciona sugerencias y completado de código rápido y preciso.
- **Análisis estático**: El IDE realiza un análisis exhaustivo del código en tiempo real, lo que ayuda a identificar posibles errores y problemas de rendimiento
- **Refactorización automatizada**: Mejora de código sin modificar su funcionalidad.
- **Depuración avanzada**: Ofrece herramientas de depuración avanzadas, incluida la capacidad de evaluar expresiones en tiempo de ejecución, inspeccionar variables y rastrear el flujo de ejecución, lo que facilita la detección y resolución de errores de manera eficiente.

3.2. Eclipse

Eclipse es un entorno de desarrollo integrado (IDE) utilizado en la programación informática.

Eclipse fue lanzado en noviembre de 2001 por el fabricante de software IBM como sucesor del entorno de desarrollo IBM Visual Age para Java 4.0. El consorcio Eclipse liderado por IBM decidió en febrero de 2004 establecer una Fundación Eclipse independiente. Desde entonces, esta fundación se ha encargado del desarrollo del entorno de programación.

Es el segundo IDE más popular para el desarrollo de Java y, hasta 2016, fue el más popular. Eclipse está escrito principalmente en Java y su uso principal es para desarrollar aplicaciones Java, pero también se puede usar para desarrollar aplicaciones en otros lenguajes de programación a través de complementos.

Algunas de las funcionalidades clave de Eclipse son:

- Interfaz de usuario: Diseño limpio y organizado, intuitivo para principiantes.
- Editor de código: Altamente personalizable, con resaltado de sintaxis, autocompletado y sugerencias contextuales.
- Depurador: Permite rastrear y solucionar problemas en el código de manera efectiva.
- Herramientas de refactorización: Ofrecen opciones para reestructurar el código de forma segura.
- Soporte de plugins: Permite personalizar el IDE con una amplia variedad de plugins.
- Integración con control de versiones: Facilita la gestión de versiones y la colaboración en equipo.

3.3 NetBeans

NetBeans IDE es un entorno de desarrollo integrado, gratuito y de código abierto para el desarrollo de aplicaciones en los sistemas operativos Windows, Mac, Linux y Solaris.

Se trata de un editor de código fuente, junto con recursos de construcción automáticos y un depurador. También tiene la función de autocompletado inteligente de código o IntelliSense. Y en el caso de la plataforma de Netbeans IDE, ofrece un compilador y un intérprete.

NetBeans ayuda en el desarrollo de cualquier tipo de software y en sus distintas fases. A los programadores les sirve como medio para la escritura, compilación o, simplemente, para hallar errores en programas basados en los anteriores lenguajes.

Características principales de NetBeans

- Soporte para múltiples lenguajes: NetBeans es compatible con una amplia gama de lenguajes de programación, lo que te permite trabajar en diferentes proyectos sin tener que cambiar de herramienta.
- Editor de código avanzado: El editor de NetBeans cuenta con características avanzadas, como el resaltado de sintaxis, la auto-completación de código y la navegación rápida entre clases y métodos. Estas funcionalidades te ayudarán a escribir código de manera más rápida y precisa.

- **Depurador integrado:** NetBeans ofrece un poderoso depurador que te permite identificar y corregir errores en tu código de manera eficiente. Puedes establecer puntos de interrupción, inspeccionar variables y ejecutar tu programa paso a paso para encontrar fallos en tiempo de ejecución.
- **Gestión de proyectos:** NetBeans cuenta con un sistema de gestión de proyectos que te permite organizar tus archivos de manera efectiva. Puedes crear, importar y exportar proyectos fácilmente, lo que te ayudará a mantener tus proyectos ordenados y estructurados

Para programar de manera eficiente en NetBeans, es recomendable:

- Organizar el código: Utiliza una estructura clara con paquetes y carpetas.
- Aprovechar plantillas: Emplea las plantillas predefinidas para agilizar la codificación.
- Dominar atajos: Utiliza los atajos de teclado para aumentar tu velocidad.
- Depurar con regularidad: Utiliza el depurador para identificar y corregir errores.

4. Ventajas y desventajas de los 3 IDEs

4.1. IntelliJ IDEA-Ventajas

-Autocompletado inteligente

Una de las características más destacadas de IntelliJ IDEA es su autocompletado inteligente. El IDE analiza el código existente y sugiere automáticamente términos relevantes mientras escribes. Esto ayuda a ahorrar tiempo y reduce la posibilidad de cometer errores.

-Navegación de código

El depurador integrado facilita la identificación y corrección de errores en el código.

- Extensibilidad

IntelliJ IDEA es altamente extensible y permite a los desarrolladores agregar funcionalidad adicional a través de complementos y extensiones. Esto significa que puede personalizar el IDE según sus necesidades y optimizar su flujo de trabajo. Además, IntelliJ IDEA cuenta con una amplia comunidad de desarrolladores que crean y comparten complementos útiles.

- Integración con herramientas de construcción

Integración perfecta con diferentes herramientas y frameworks populares, como Maven, Gradle y Spring. Estas integraciones permiten a los desarrolladores trabajar de manera más eficiente y optimizar su flujo de trabajo al proporcionar características específicas para cada herramienta.

-Funcionalidades

Ofrece una amplia gama de complementos y plugins que permiten personalizar la experiencia de desarrollo según las necesidades de cada programador.

4.2. IntelliJ IDEA-Desventajas

-Consumo de recursos

Requiere hardware potente y ajustes adecuados para evitar ralentizaciones, especialmente en proyectos grandes.

-Curva de aprendizaje

La interfaz y la cantidad de opciones pueden resultar abrumadoras al principio, especialmente para quienes vienen de otros IDEs.

-Licencia de pago

Requiere una licencia comercial para su uso completo, aunque existe una versión de prueba gratuita.

-Menor cantidad de complementos

Aunque tiene una amplia selección, la cantidad total de complementos puede ser menor que en otros IDEs.

-Integración limitada con lenguajes de nicho

Puede ser menos eficiente con lenguajes menos comunes debido a la menor cantidad de herramientas específicas.

-Interfaz de usuario compleja

La interfaz puede parecer abrumadora al principio, requiriendo tiempo para familiarizarse con todas las opciones.

-Dependencia de JetBrains

Al ser un producto de una sola empresa, la evolución y el futuro de IntelliJ IDEA están ligados a las decisiones de JetBrains.

4.3. Eclipse-Ventajas

-Mayor productividad

El editor inteligente, la depuración eficiente y las herramientas de refactorización agilizan el desarrollo y reducen errores.

-Código de mejor calidad

Las herramientas de refactorización y la posibilidad de depurar el código permiten crear software más robusto y mantenible.

-Flexibilidad

El soporte para múltiples lenguajes y la personalización a través de plugins hacen que Eclipse sea adaptable a diferentes proyectos.

-Colaboración efectiva

La integración con control de versiones facilita el trabajo en equipo y la gestión de cambios.

-Gran comunidad

La comunidad activa de Eclipse ofrece soporte y recursos para resolver problemas y aprender nuevas habilidades.

4.4 Eclipse - Desventajas

-Curva de aprendizaje

Aunque su interfaz es intuitiva, la configuración y personalización de Eclipse pueden requerir un tiempo de aprendizaje, especialmente para usuarios principiantes.

-Rendimiento

En proyectos grandes, Eclipse puede volverse lento, especialmente si se utilizan muchos plugins.

-Fragmentación de la comunidad

La amplia gama de plugins y distribuciones puede llevar a una cierta fragmentación de la comunidad, dificultando a veces encontrar soluciones a problemas específicos.

-Mantenibilidad de proyectos

Debido a su naturaleza extensible, la configuración de un proyecto Eclipse puede volverse compleja y difícil de mantener a largo plazo.

4.5 NetBeans - Ventajas

-Código abierto

NetBeans es de código abierto y por ende gratuito, de modo que cualquiera puede descargarse esta aplicación.

-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.

4.6 NetBeans – Desventajas

- Poca disponibilidad de plugins

Si bien NetBeans cuenta con una comunidad de desarrolladores que crean plugins, la variedad y frecuencia de actualizaciones pueden ser menores en comparación con otros IDEs más populares como Eclipse o Visual Studio Code.

- Rendimiento en proyectos grandes

NetBeans puede volverse lento al trabajar con proyectos grandes y complejos, especialmente cuando se realiza una indexación completa del código. Esto se debe a que analiza todo el código para proporcionar funcionalidades como autocompletado y navegación, lo que puede consumir muchos recursos del sistema.

- Documentación limitada en algunos aspectos

La documentación oficial de NetBeans puede ser escasa o estar desactualizada en algunas áreas, como el RCP (Rich Client Platform). Esto puede dificultar la resolución de problemas y la búsqueda de información específica.

-Falta de soporte para webapps

Si bien NetBeans permite desarrollar aplicaciones web, su soporte para tecnologías como servlets, JSP y frameworks web específicos puede no ser tan completo o integrado como en otros IDEs diseñados específicamente para desarrollo web.

5.Instalación de IntelliJ IDEA

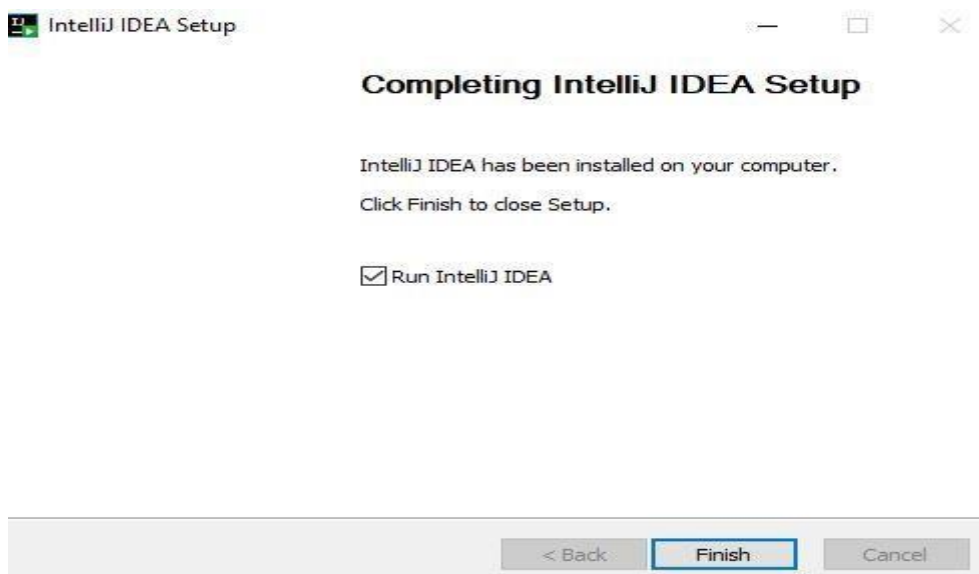
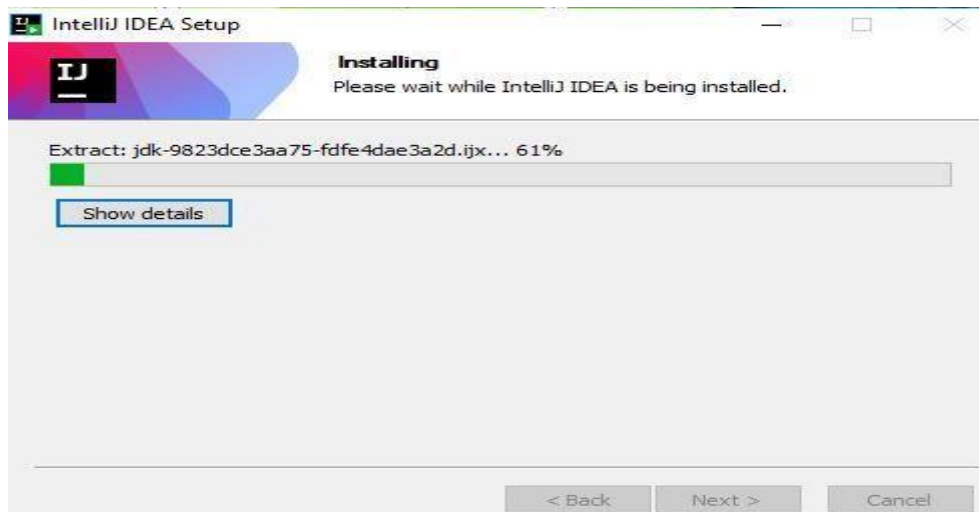
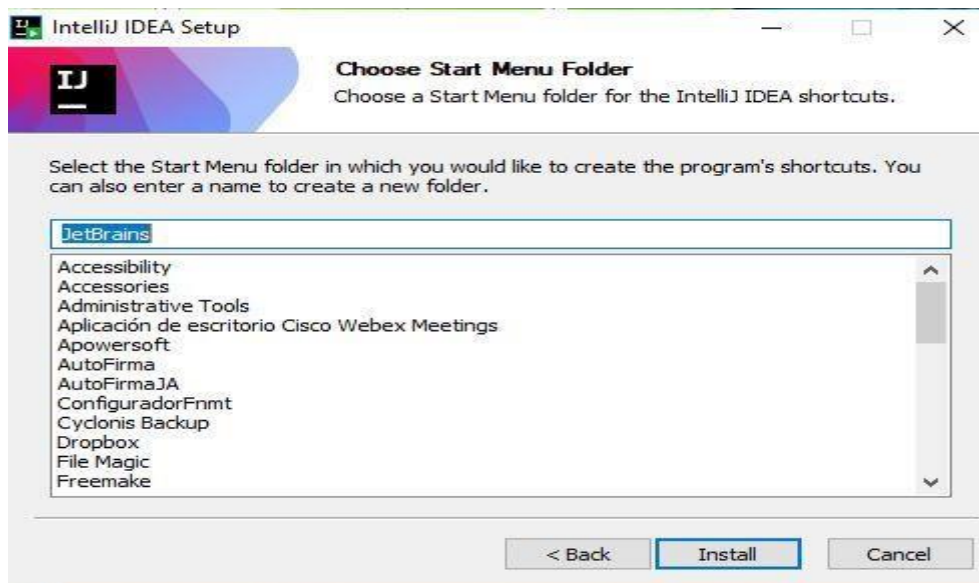
5.1. Descarga e instalación

-Visita el sitio web oficial de JetBrains: <https://www.jetbrains.com/idea/>.

-Descarga la versión adecuada de IntelliJ IDEA para tu sistema operativo (Windows, macOS o Linux).

3. Ejecuta el instalador y sigue las instrucciones en pantalla para completar la instalación.





5.2. Configuración inicial

Una vez instalada, IntelliJ IDEA te guía a través de una configuración inicial para personalizar tu entorno de desarrollo.

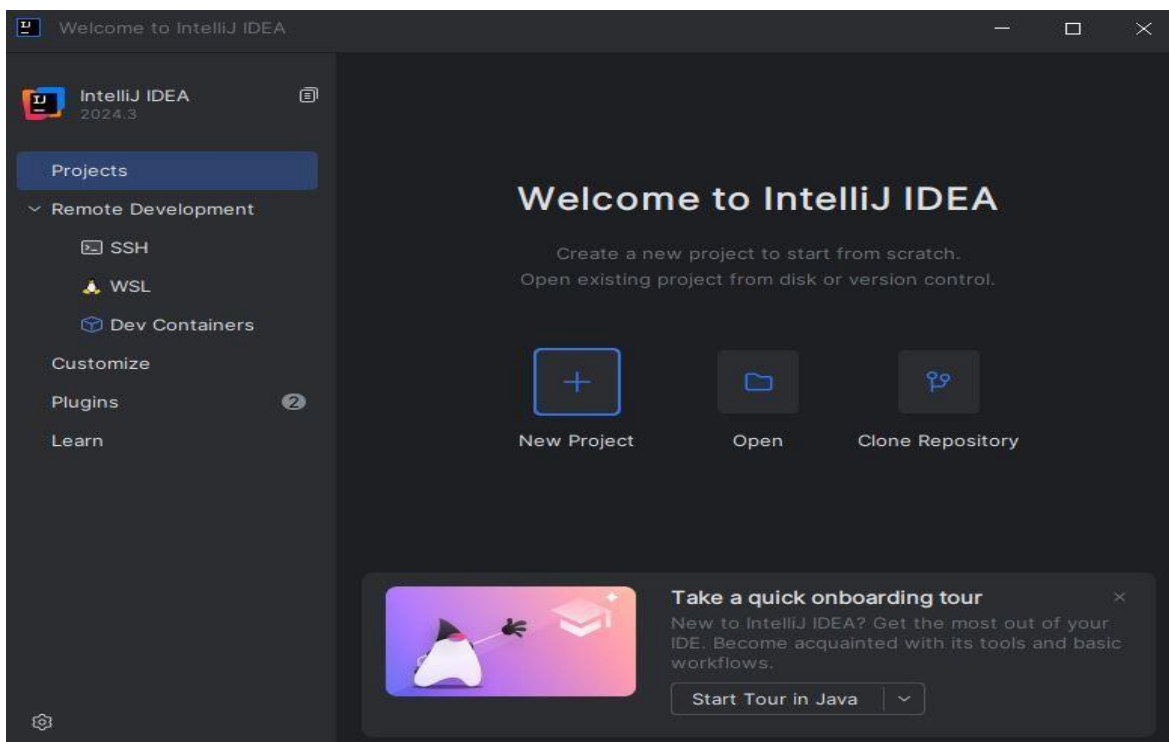
-Selecciona un tema: Elige entre el tema claro u oscuro para ajustar la apariencia del IDE a tus preferencias.

-Configuración del teclado: Elige un esquema de teclado que te resulte familiar.

-Plugins: IntelliJ IDEA ofrece una amplia gama de plugins para ampliar sus funcionalidades.

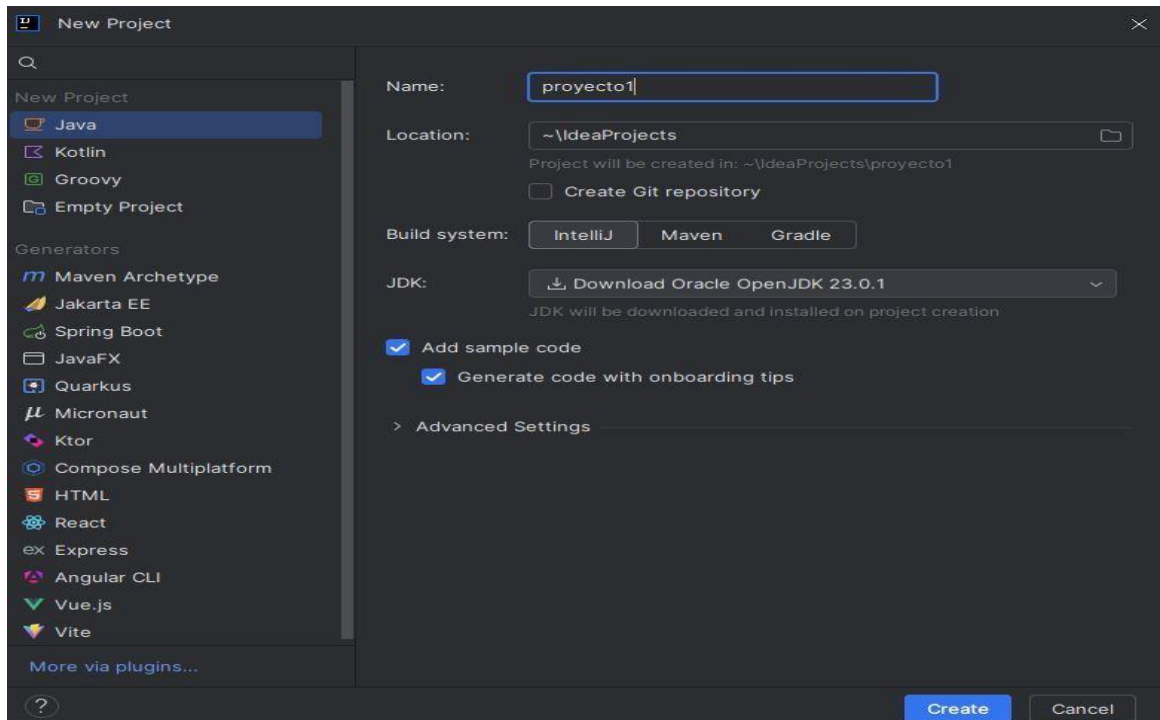
5.3 Creación de un nuevo proyecto

Una vez que IntelliJ IDEA está configurado, ya se puede empezar a crear proyectos de Java.



5.3.1 Creación de un proyecto Java

1. Haga clic en “Create New Project” en la pantalla de bienvenida o ve a “File > New > Project”.
2. Selecciona “Java” como el tipo de proyecto y elige la versión de Java que desees utilizar.
3. Asigna un nombre al proyecto y especifica la ubicación de la carpeta del proyecto.
4. Haz clic en “Finish” para crear el proyecto.

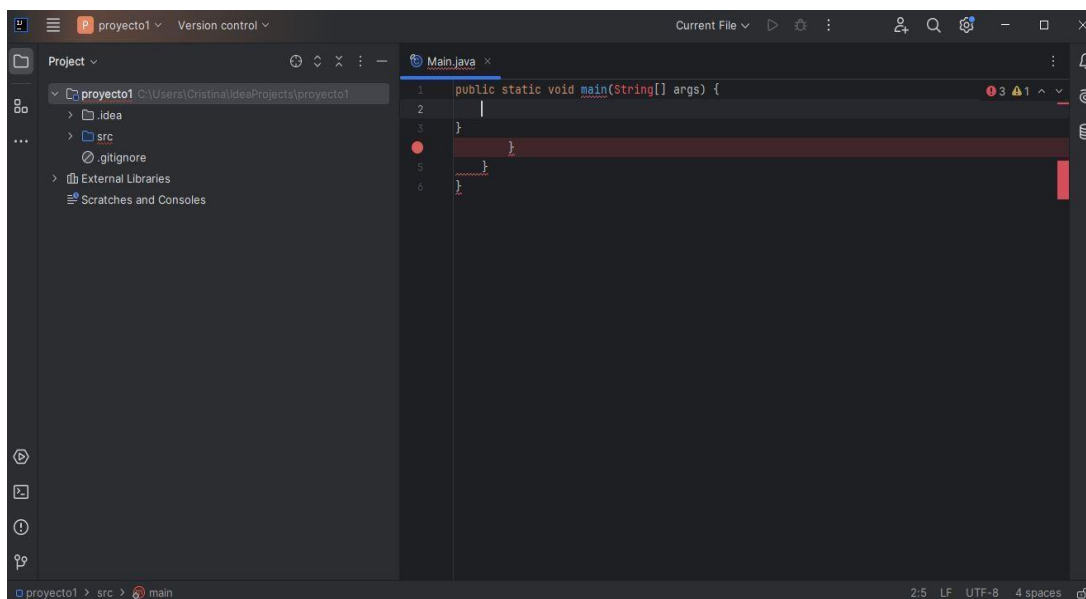


5.3.2 Estructura del proyecto

IntelliJ IDEA organiza su proyecto en una estructura de directorios lógica que facilita la gestión de los archivos de código fuente, las dependencias y los recursos.

La estructura típica de un proyecto Java incluye:

- **src**: Contiene los archivos de código fuente de Java.
- **lib**: Almacena las bibliotecas externas que necesita el proyecto.
- **resources**: Contiene archivos de recursos como archivos de propiedades, archivos de configuración y archivos de imágenes.
- **out**: Directorio de salida para los archivos compilados.



6. Problemas que nos encontramos en la instalación.

No se registraron problemas durante el proceso de instalación.

Algunos posibles errores al instalar IntelliJ IDEA pueden ser:

- Versiones incompatibles de JDK.
- Errores de instalación por falta de espacio en disco.
- Permisos insuficientes.
- Conflictos con otras versiones de IntelliJ.
- Errores en la descarga.
- Configuraciones incorrectas.

7.Soluciones aportadas.

Soluciones a los posibles problemas planteados anteriormente.

- Problemas de compatibilidad con el sistema operativo: Asegúrate que el sistema operativo es compatible con la versión de IntelliJ IDEA que estamos instalando.
- Dependencias faltantes: Verifica que tengas instaladas las dependencias necesarias, como una versión adecuada del JDK, ya que, sin ellas, IntelliJ IDEA no funcionará correctamente.
- Errores de instalación por falta de espacio en disco: Asegúrate de contar con suficiente espacio libre en el disco duro para completar la instalación.
- Problemas de red o firewall: Asegúrate de tener acceso a Internet y de que un firewall no esté bloqueando las descargas necesarias para la instalación o actualización.
- Errores de configuración o preferencias corruptas: Si tienes configuraciones previas de IntelliJ IDEA o preferencias corruptas de versiones anteriores, pueden interferir con la instalación. Asegúrate de eliminar configuraciones antiguas antes de instalar una nueva versión.
- Archivos de instalación corruptos: Si la descarga de IntelliJ IDEA fue interrumpida o el archivo descargado está dañado, puede que no se complete la instalación correctamente.

8.Webgrafía

Anandmeg. (2024, 21 de junio). *¿Qué es el IDE de Visual Studio?* Microsoft Learn. Recuperado el 30 de noviembre de 2024, de <https://learn.microsoft.com/es-es/visualstudio/get-started/visual-studio-ide?view=vs-2022>

Daniel. (2023, 24 de octubre). *IDE: ¿Qué es un entorno de desarrollo integrado?* Formación en Ciencia de Datos | DataScientest. Recuperado el 27 de noviembre de 2024, de <https://datascientest.com/es/ide>

Definición de IDE: Qué es, 5 Ejemplos, Tipos y Para qué Sirve + Sinónimo y Significado. (2023, 26 de octubre). SignificadosWeb.com. Recuperado el 27 de noviembre de 2024, de <https://significadosweb.com/definicion-de-ide-que-es-ejemplos-tipos-y-para-quesirve-sinonimo-y-significado/#>

- Diarlu, D. (2019, 15 de agosto). *22 entornos de desarrollo integrado (IDE) más populares*. Diarlu. Recuperado el 29 de noviembre de 2024, de <https://www.diarlu.com/entornos-de-desarrollo-integrado/>
- Fantino, J. (2021, 6 de noviembre). *¿Qué es Netbeans? ¡Crea aplicaciones con Java a la velocidad de la luz!* Crehana. Recuperado el 3 de diciembre de 2024, de <https://www.crehana.com/blog/transformacion-digital/que-es-netbeans/>
- García, J. (2023, 18 de mayo). *¿Qué es Eclipse? Definición | Herramienta de programación*. Tecno Simple. Recuperado el 30 de noviembre de 2024, de <https://tecnosimple.com/que-es-eclipse-definicion-herramienta-de-programacion/>
- Jcweb. (2024, 23 de enero). *NetBeans: Qué es y cómo utilizarlo para programar de forma efectiva*. Jcweb.es. Recuperado el 2 de diciembre de 2024, de <https://jcweb.es/netbeansque-es-y-como-utilizarlo-para-programar-de-forma-efectiva/>
- Kumar, C. (2024, 13 de septiembre). *13 mejores IDE para programadores para agilizar su flujo de trabajo de desarrollo*. Geekflare Spain. Recuperado el 29 de noviembre de 2024, de <https://geekflare.com/es/best-ides-for-programmers/>
- León, J. T. (2024, 2 de marzo). *El entorno de desarrollo integrado Eclipse: Todo lo que necesitas saber*. Taboada León. Recuperado el 30 de noviembre de 2024, de <https://taboadaleon.es/el-entorno-de-desarrollo-integrado-eclipse-todo-lo-que-necesitassaber/>
- López, M. (2022, 13 de abril). *¿Qué es NetBeans? Ventajas y usos*. IMMUNE Technology Institute. Recuperado el 2 de diciembre de 2024, de <https://immune.institute/blog/que-es-netbeans/>
- ¿Qué es y para qué sirve un IDE?* (2023, 31 de julio). Red Hat. Recuperado el 27 de noviembre de 2024, de <https://www.redhat.com/es/topics/middleware/what-is-ide>
- Team, S. (2024, 11 de marzo). *IDEs: Qué son, para qué sirven y los más populares*. SW Hosting. Recuperado el 29 de noviembre de 2024, de <https://www.swhosting.com/es/blog/ides-que-son-para-que-sirven-y-los-mas-populares>
- Ventajas y desventajas de las características de IntelliJ IDEA*. (2024, 23 de enero). Proscont.com. Recuperado el 30 de noviembre de 2024, de https://www.proscont.com/ventajas-y-desventajas-intellij-ideacaracteristicas/#ventajas_de_intellij_idea
- Vidal, S. (2023, 15 de julio). *¿Qué es IntelliJ IDEA?* Tecnobits. Recuperado el 30 de noviembre de 2024, de <https://tecnobits.com/que-es-intellij-idea/>