

TRABAJO DE INVESTIGACIÓN ED02

1.-Introducción.

En la presente unidad de trabajo se ha estudiado como instalar y usar un entorno de desarrollo.

Se ha visto:

- **Cómo instalarlo.**
- **Configurar y personalizar el entorno de desarrollo.**
- **Crear un proyecto, insertar código y ejecutarlo.**
- **Añadir y eliminar un plugin.**

2.-Práctica.

Busca cuales son los IDE más usados actualmente (que sirvan para programar en Java).

Elige uno de ellos y haz una guía de usuario para manejar dicho entorno. No puede ser ni NetBeans ni Eclipse.

IntelliJ Idea

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

- Community Edition: libre descarga.
- Ultimate Edition: que cuesta más de 500 dólares de suscripción anual, según el plan que contratado, aunque ofrece una prueba gratis durante 30 días para 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.

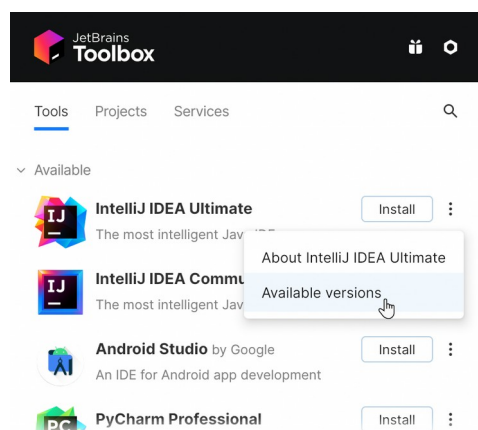
Además ofrece integración con GIT lo que permite trabajar con diferentes versiones de nuestro software sin afectar a la línea maestra de nuestro desarrollo.

▫ Cómo instalarlo

La aplicación **JetBrains Toolbox** es la herramienta recomendada para instalar productos JetBrains. Úselo para instalar y administrar diferentes productos o varias versiones del mismo producto, actualizar y revertir cuando sea necesario y eliminar fácilmente cualquier herramienta.

La aplicación Toolbox mantiene una lista de todos sus proyectos para abrir rápidamente cualquier proyecto en el IDE y la versión correctos.

1. **Descargue el instalador .exe** desde la página web de la aplicación Toolbox (<https://www.jetbrains.com/toolbox-app/>).
2. **Ejecute el instalador** y siga los pasos del asistente.
3. Después de ejecutar la aplicación Toolbox, haga clic en su icono Icono de la aplicación Toolbox en el área de notificación y **seleccione qué producto desea instalar**.
4. Para instalar una versión específica, haga clic en **⋮** y seleccione Versiones disponibles .



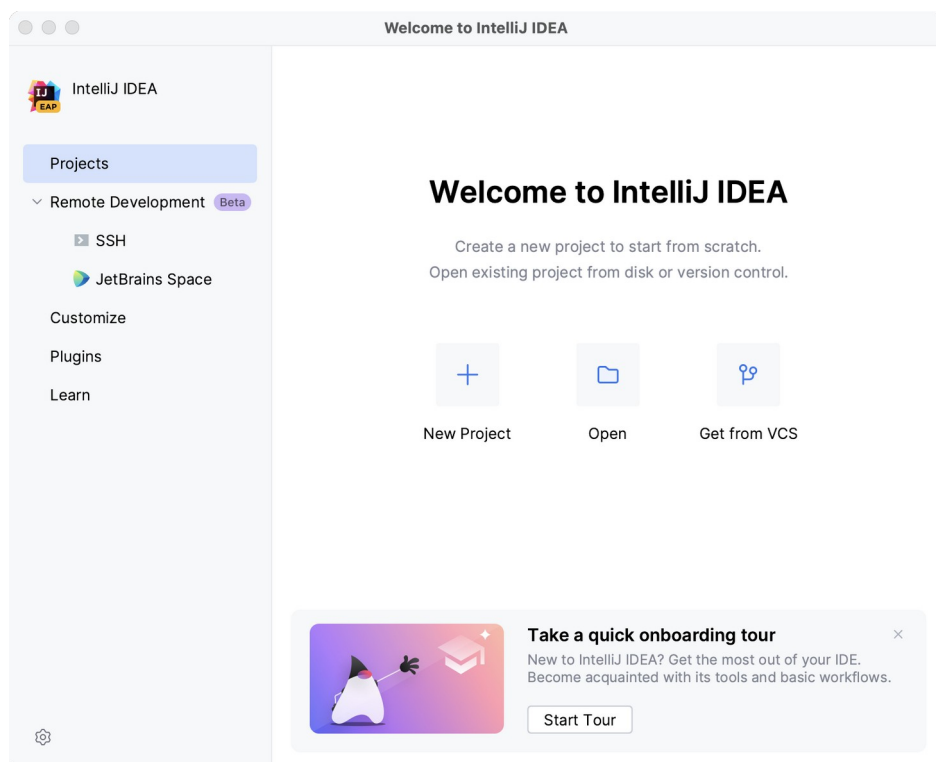
Inicie sesión en su cuenta JetBrains desde la aplicación Toolbox y activará automáticamente las licencias disponibles para cualquier IDE que instale.

También se encuentran disponibles la instalación independiente, la instalación silenciosa para Windows y la instalación como un paquete instantáneo en Linux.

Para ejecutar por primera vez IntelliJ IDEA se puede realizar a través de la aplicación Toolbox (También para ejecutar cualquier producto JetBrains o, en caso de haber realizado una instalación independiente, su ejecución dependerá del sistema operativo en el que se haya realizado:

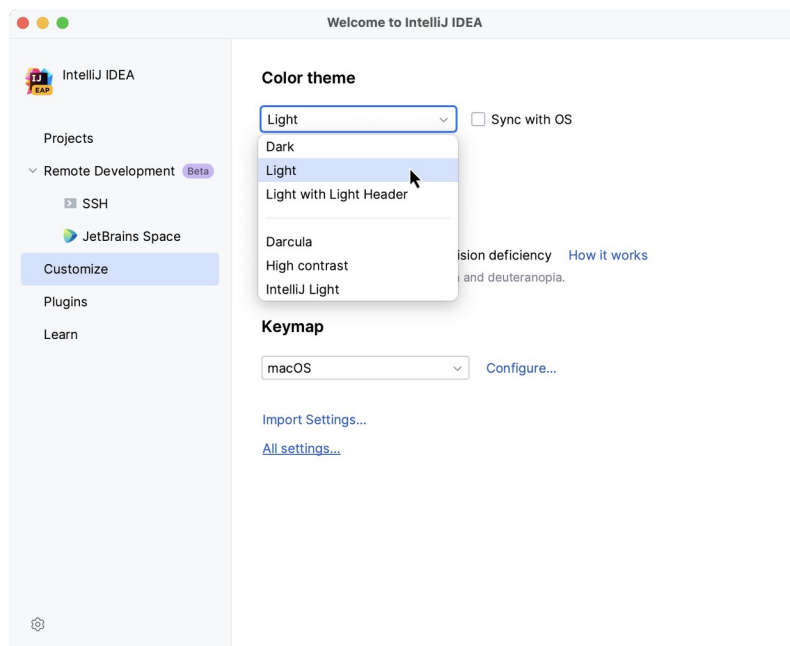
- En Windows: Para ejecutar IntelliJ IDEA, búsquelo en el menú Inicio de Windows o utilice el **acceso directo del escritorio**. También puede ejecutar el script por lotes del iniciador o el ejecutable en el directorio de instalación en bin .
- En Mac OS: Ejecute la aplicación IntelliJ IDEA desde el **directorio de aplicaciones , Launchpad o Spotlight**.
- En Linux: **Ejecute el script de shell idea.sh** en el directorio de instalación en bin . También puede utilizar el **acceso directo del escritorio** si se creó durante la instalación.

Una vez que inicie IntelliJ IDEA, verá la pantalla de bienvenida , el punto de partida para trabajar con el IDE y configurar sus ajustes. Esta pantalla también aparece cuando cierra todos los proyectos abiertos. Utilice las pestañas del lado izquierdo para cambiar al cuadro de diálogo de bienvenida específico.

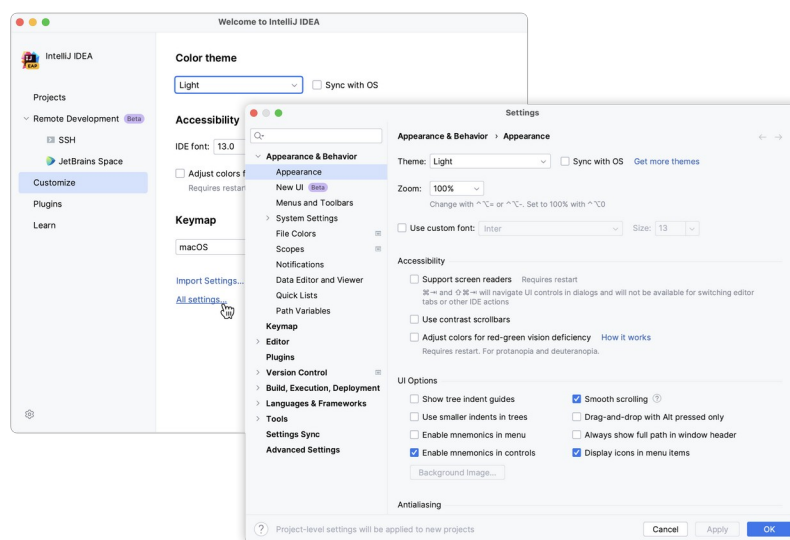


Configurar y personalizar el entorno de desarrollo

Haga clic en **Personalizar** y seleccione otro **Tema de color** o seleccione la casilla de verificación Sincronizar con el sistema operativo para usar el tema predeterminado de su sistema. Aquí también puede configurar los ajustes de accesibilidad o seleccionar otro mapa de teclas.



Seleccione **Importar** configuración si ya tiene una copia de la configuración requerida que desea aplicar. Alternativamente, haga clic en Todas las configuraciones para abrir el cuadro de diálogo de configuración. La configuración que modifique en este momento se convertirá en la nueva configuración predeterminada para tus proyectos y el IDE.



Haga clic en **Aprender** para familiarizarse más con los accesos directos, las funciones y los flujos de trabajo que ofrece el IDE. Para obtener más información, consulte Aprenda las funciones de IDE .

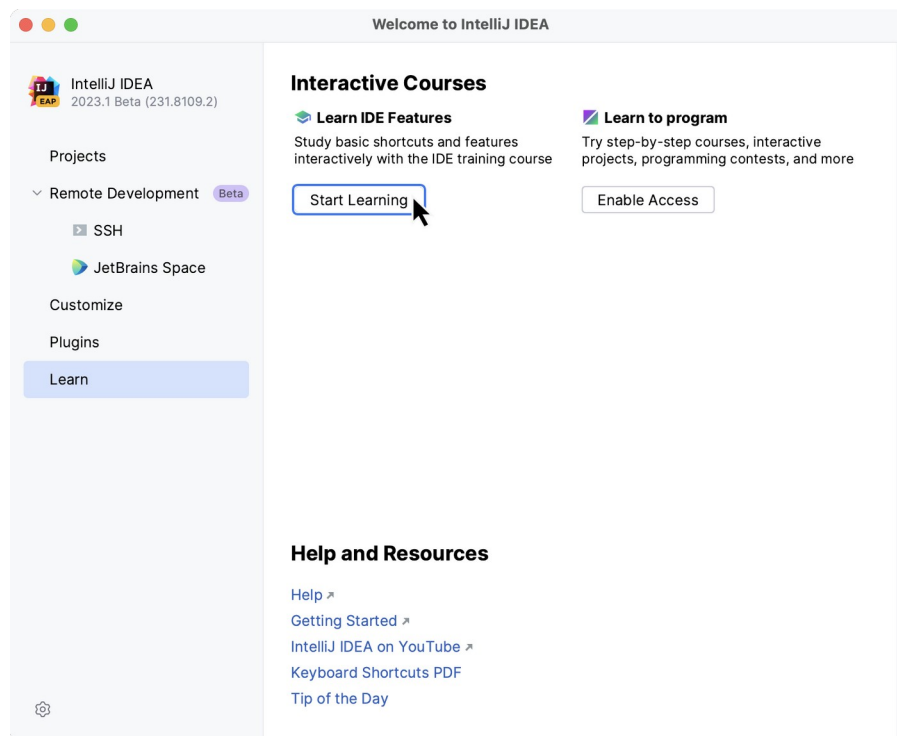
IntelliJ IDEA incluye un **Entrenador de funciones integrado**. Este curso de capacitación interactivo sobre los conceptos básicos de IDE puede ayudarlo a aprender cómo realizar algunas tareas comunes, como refactorizar código, navegar por proyectos y más.

El curso se divide en varios tutoriales, cada uno con una serie de lecciones que le enseñan cómo realizar una tarea específica. Puede iniciar el curso o un tutorial específico desde la pantalla de bienvenida o desde la ventana de la herramienta Aprender .

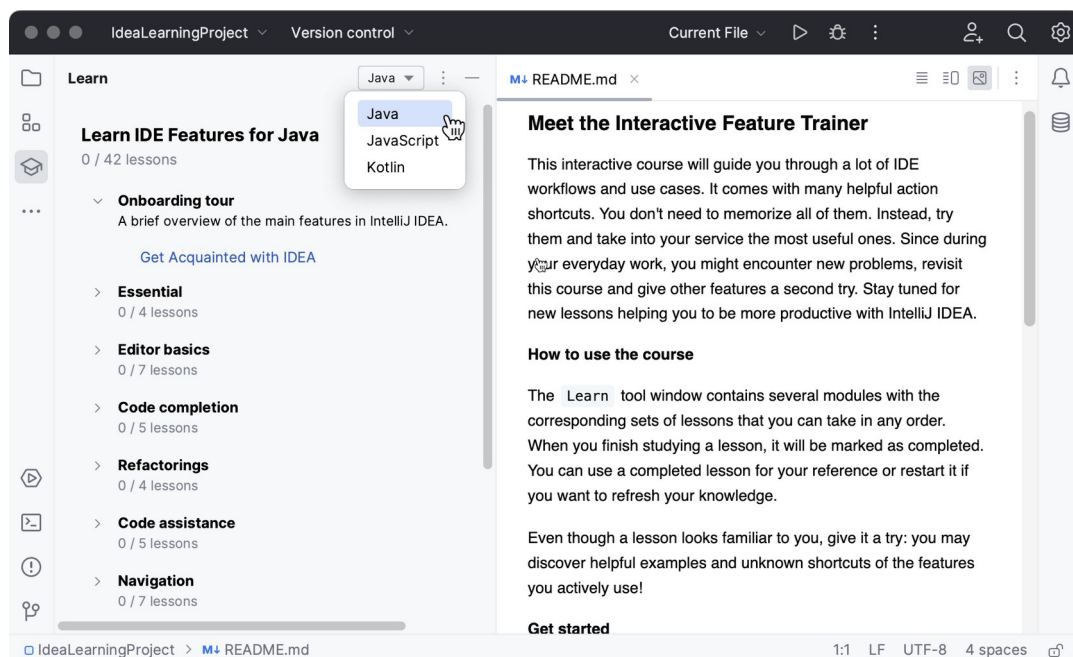
Para completar una lección, debes terminar todas las tareas que contiene. Sin embargo, no es necesario seguir todas las lecciones y tutoriales; puede optar por realizar solo las que considere más relevantes;

Antes de empezar, asegúrese de que el complemento incluido **IDE Features Trainer** esté **habilitado** en la pestaña Instalado de Configuración | Página de complementos y de estar utilizando la nueva interfaz de usuario .

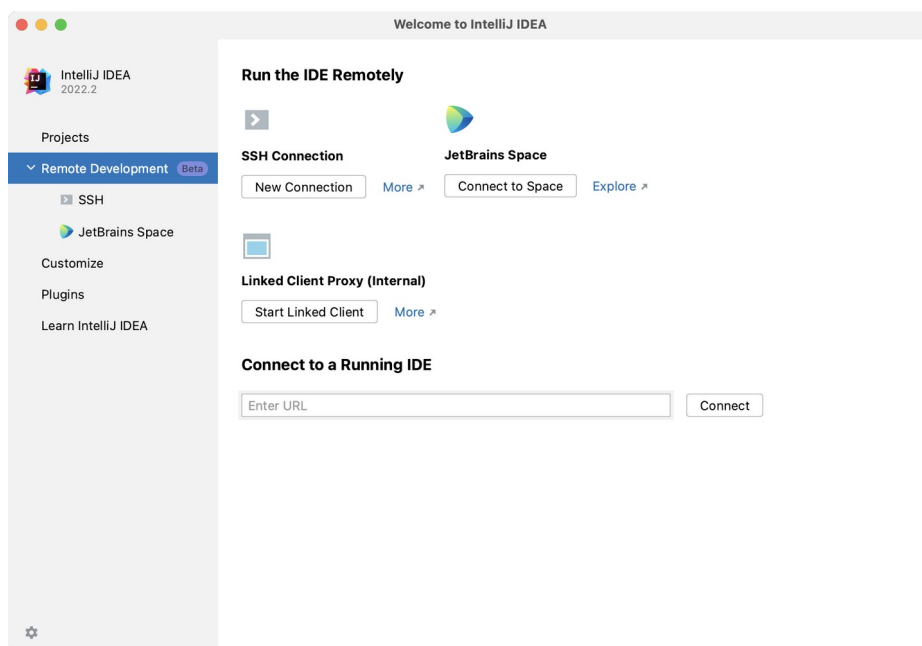
Para iniciar el curso de aprendizaje haga clic en **Comenzar a aprender** en **Aprender características del IDE** .



IntelliJ IDEA genera un proyecto de aprendizaje y abre el contexto que corresponde a la primera lección del tutorial seleccionado.



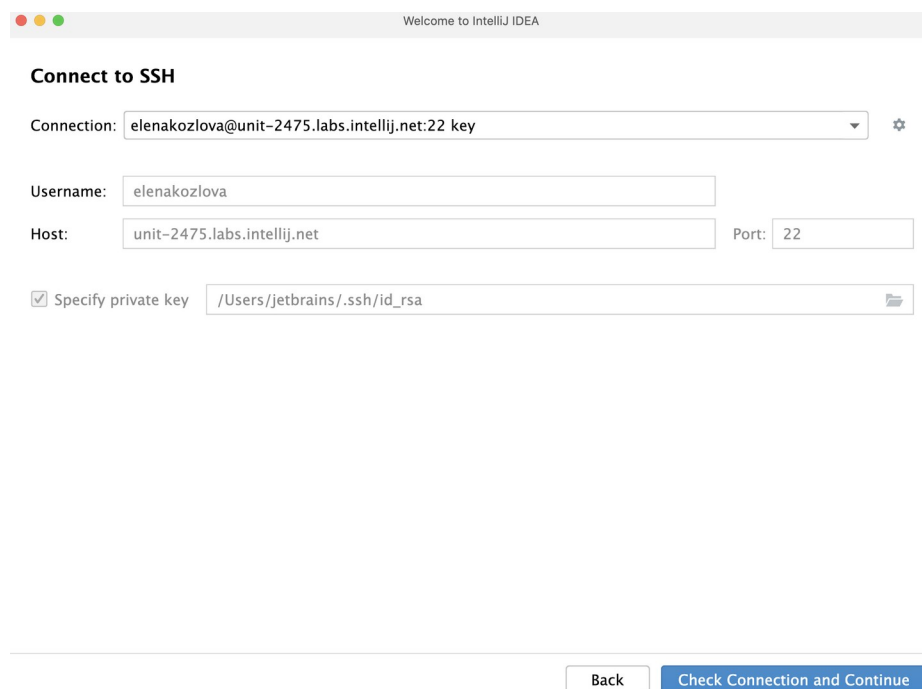
Haga clic en **Desarrollo remoto** para configurar un punto de entrada para conectarse a un servidor remoto y codificar, ejecutar, depurar e implementar sus proyectos de forma remota.



Antes de empezar, Asegúrese de tener **habilitado el complemento Remote Development Gateway**.

En la sección **Ejecutar el IDE de forma remota**, haga clic en **Conexión SSH** (Si ya tiene el IDE ejecutándose en el servidor remoto y tiene un enlace de conexión, puede usar la sección Conectarse al IDE en ejecución).

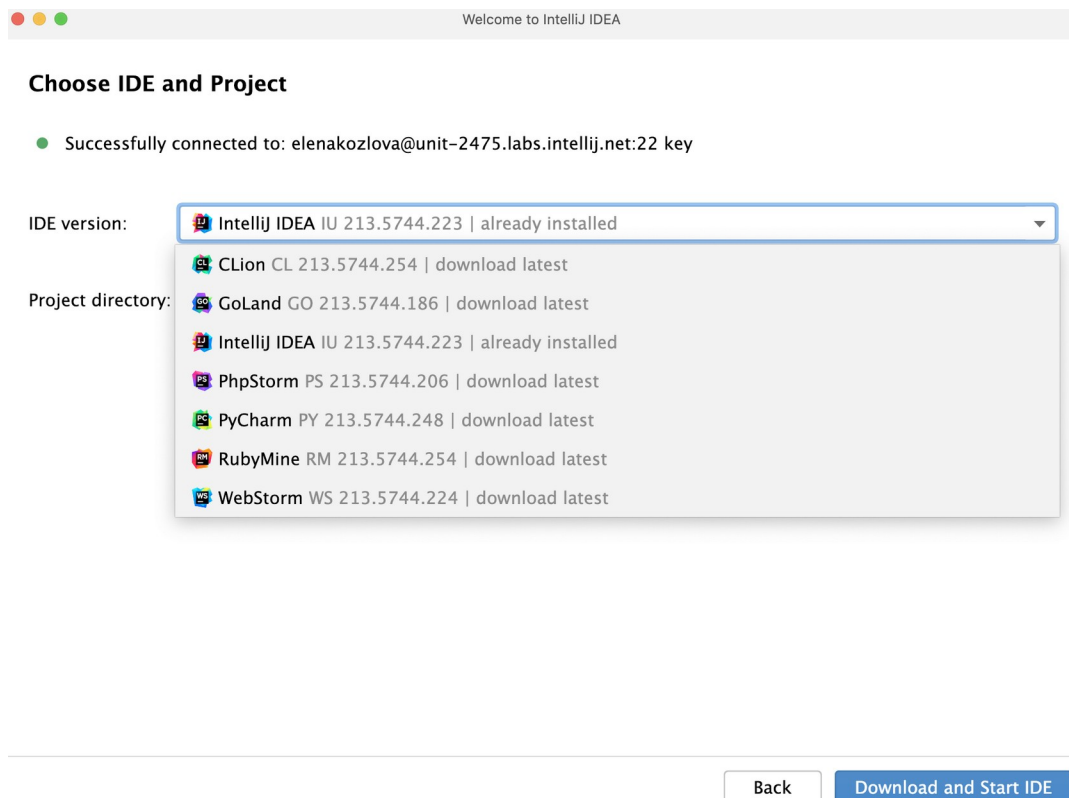
Configure los parámetros de conexión del servidor remoto y haga clic en **Verificar conexión** y continuar para asegurarse de que la conexión al servidor remoto sea exitosa.



En la siguiente página del asistente, en el campo **Versión de IDE**, seleccione la fuente del IDE remoto que desea utilizar.

Utilice una de las siguientes formas de obtener un instalador IDE:

- **Recuperar automáticamente desde el almacenamiento de los instaladores de JetBrains:** variante predeterminada .
- Obsérvelo del **almacenamiento interno de su empresa**. En este caso, debe hacer clic en Otras opciones y seleccionar Usar enlace de descarga . Es útil si las máquinas remotas no tienen una conexión a Internet con los sitios web de JetBrains o si su organización utiliza algunas compilaciones personalizadas.
- Sube **desde tu máquina local**. En este caso, haga clic en Otras opciones y seleccione Cargar archivo de instalación . Debe obtener el .tar.gz archivo IDE del sitio web de JetBrains con anticipación.



Haga clic en **Descargar e iniciar IDE**. IntelliJ IDEA inicia JetBrains Gateway, que descarga el backend del IDE, inicia y abre JetBrains Client con su proyecto remoto.

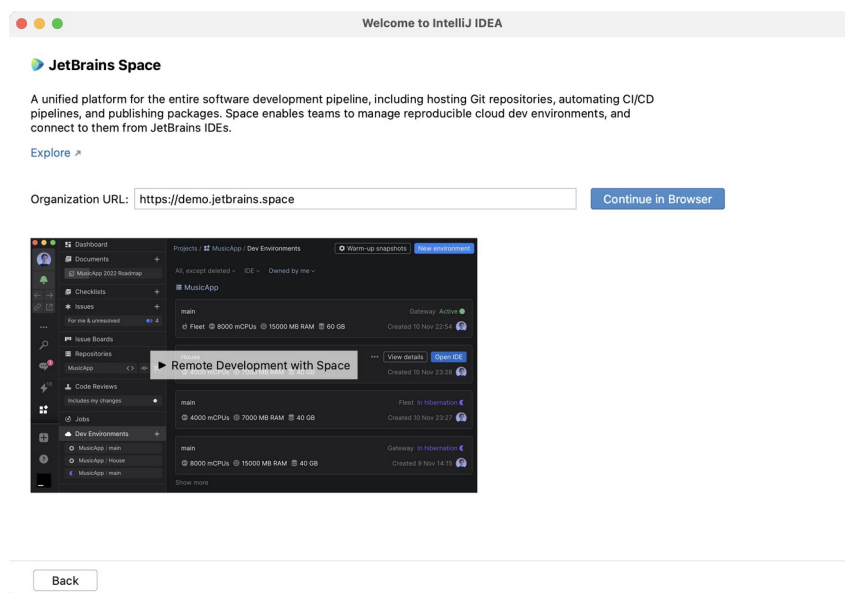
También se puede configurar su backend IDE para que se inicie directamente en **WSL2**.

Puede también **conectarse a Space** para automatizar todos los procesos de configuración de su entorno de desarrollo, que puede utilizar para el desarrollo de software.

Space automatiza los siguientes procesos:

- Creando una máquina virtual remota
- Iniciar una imagen de Docker con bibliotecas y herramientas adecuadas
- Descarga y prepara el IDE backend para que puedas abrir fácilmente tu proyecto y sumergirte en la codificación de inmediato.

Para conectarse a Space, **seleccione JetBrains Space** y haga clic en **Conectar al espacio** e ingrese la url de su **organización** y haga clic en Continuar en el navegador.

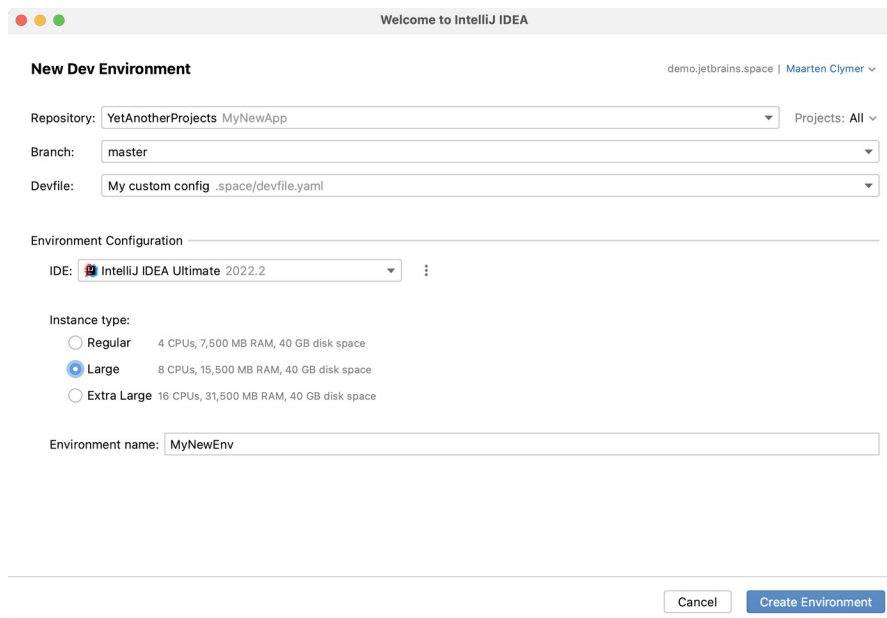


En la ventana del navegador que se abre, haga clic en Aceptar para otorgar los permisos necesarios. Si aún no ha iniciado sesión en la aplicación web de Space, también se le pedirá que ingrese su nombre de usuario y contraseña de Space. Luego será redirigido nuevamente a su IntelliJ IDEA.

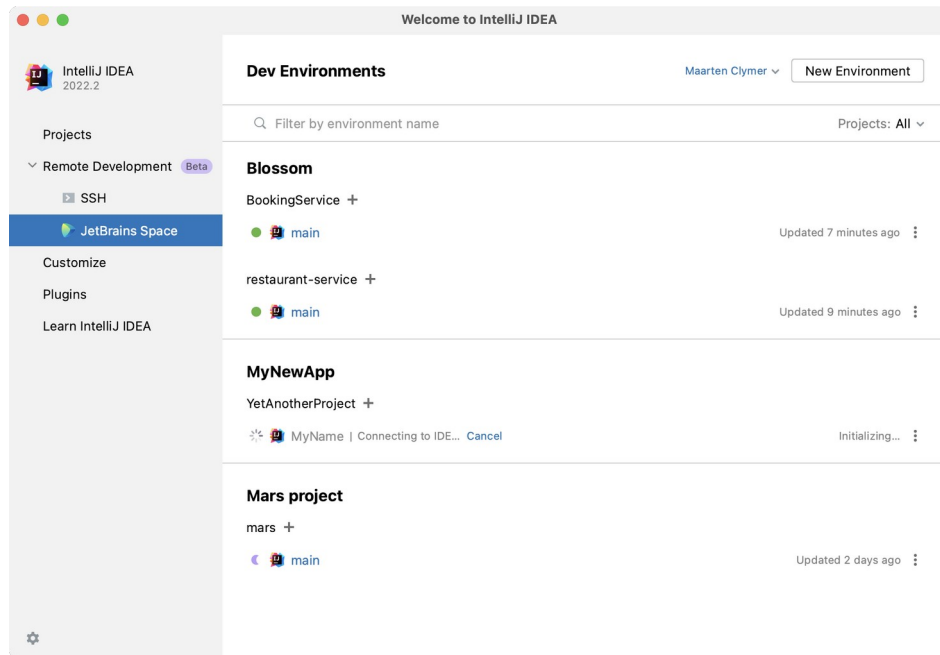
Haga clic en **Nuevo entorno** y configure un nuevo entorno de desarrollo (omite este paso si su proyecto ya tiene el entorno de desarrollo necesario):

- Seleccione un **repositorio** y una **rama** donde desee crear un entorno de desarrollo.
- En la **lista Devfile** , seleccione un archivo devfile (si existe en su proyecto Space).
- En la **lista de IDE** , seleccione un IDE para usar en el entorno de desarrollo.
- En la **lista Tipo de instancia** , seleccione un tipo de máquina virtual según los recursos que necesita y los créditos de cálculo .
- En el campo **Nombre del entorno** , introduzca un nombre para el entorno para distinguirlo de otros entornos.

Haga clic en **Crear entorno** para completar la creación del entorno.



En la lista de entornos de desarrollo, **seleccione un entorno al que desee conectarse.**



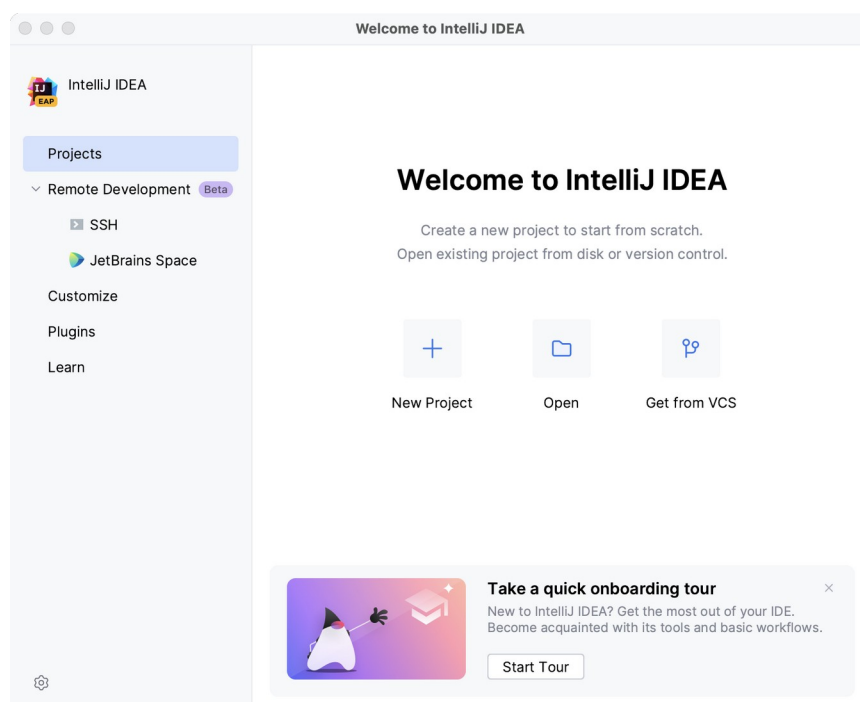
Esto iniciará una conexión al IDE remoto. Una vez que la conexión se establezca correctamente, IntelliJ IDEA abrirá JetBrains Client con su proyecto Space donde podrá seguir trabajando.

≡ Crear un proyecto, insertar código y ejecutarlo

Cuando inicie IntelliJ IDEA, le mostrará la **pantalla de bienvenida**.

Desde el cuadro de diálogo Bienvenido a IntelliJ IDEA , puede hacer lo siguiente:

- Crear un nuevo proyecto
- Crea tu primera aplicación Java
- Abrir un proyecto
- Obtener un proyecto de un sistema de control de versiones

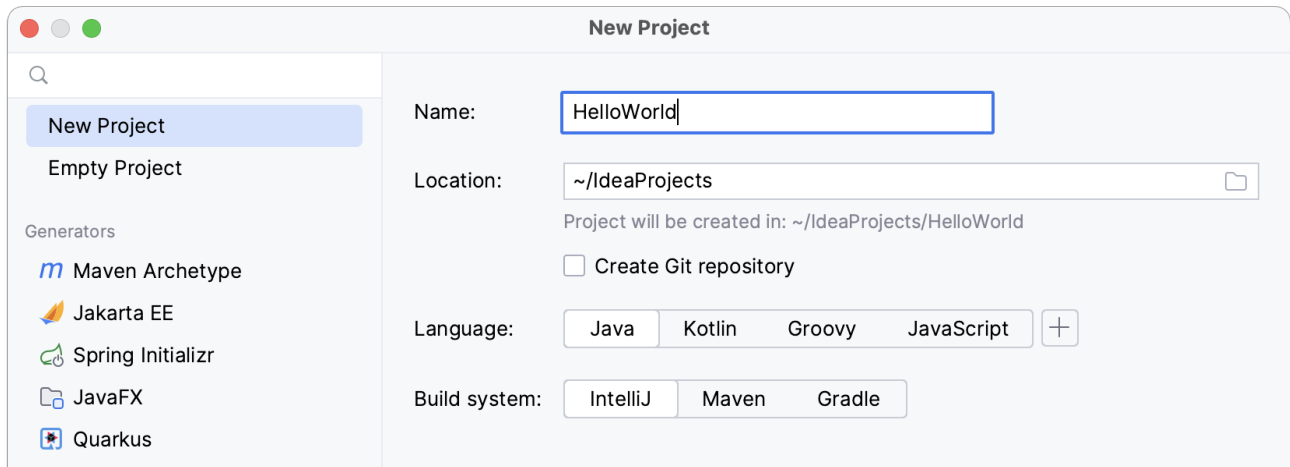


Haga clic en **Nuevo proyecto**. En el asistente de Nuevo proyecto , seleccione **Nuevo proyecto** de la lista de la izquierda.

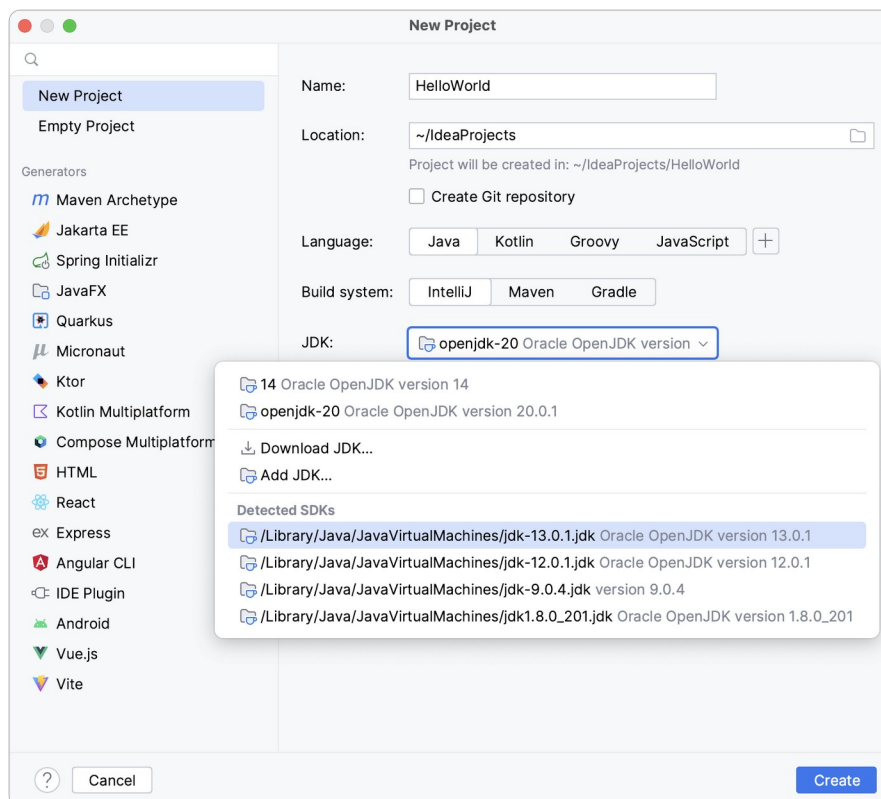
Asigne un nombre al proyecto (por ejemplo HelloWorld) y cambie la ubicación predeterminada, si es necesario.

No vamos a trabajar con sistemas de control de versiones en este tutorial, así que deje la opción Crear repositorio Git desactivada.

Asegúrese de que **Java** esté **seleccionado en Idioma** e IntelliJ en Sistema de compilación .



Para desarrollar aplicaciones Java en IntelliJ IDEA, necesita el SDK de Java (JDK). Si el JDK necesario ya está definido en IntelliJ IDEA, selecciónelo de la lista de JDK . Si el JDK está instalado en su computadora, pero no está definido en el IDE, seleccione **Agregar JDK** y especifique la ruta al directorio de inicio del JDK (por ejemplo, /Library /Java /JavaVirtualMachines /jdk-20.0.1.jdk).



Si no tiene el JDK necesario en su computadora, seleccione **Descargar JDK**. En el siguiente cuadro de diálogo, especifique el proveedor de JDK (por ejemplo, OpenJDK), la versión, cambie la ruta de instalación si es necesario y haga clic en Descargar .

Deje la **opción Agregar código de muestra deshabilitada** y haga clic en **Crear**. Después de eso, el IDE creará y cargará el nuevo proyecto por usted.

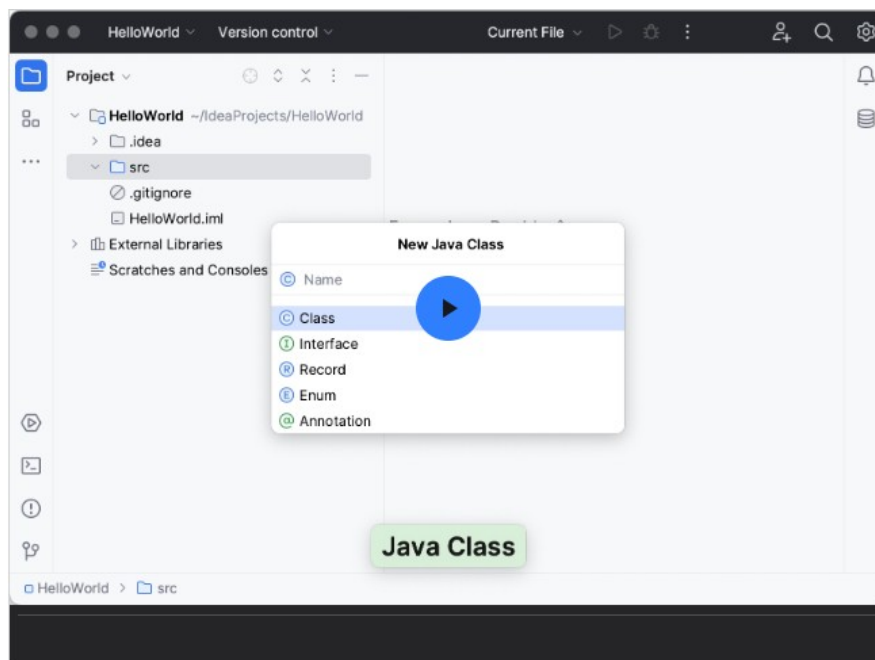
Los paquetes se utilizan para agrupar clases que pertenecen a la misma categoría o proporcionan una funcionalidad similar, para estructurar y organizar aplicaciones grandes con cientos de clases.

En la **ventana de herramientas Proyecto**, haga clic con el botón derecho en la **carpeta src**, seleccione **Nuevo (o presione)** y luego seleccione **Clase Java**.

En el **campo Nombre**, escriba `com.example.helloworld.HelloWorld` y haga clic en **Aceptar**.

IntelliJ IDEA crea el `com.example.helloworld` paquete y la `HelloWorld` clase. Junto con el archivo, IntelliJ IDEA ha generado automáticamente algunos contenidos para su clase. En este caso, el IDE ha insertado la declaración del paquete y la declaración de clase.

Esto se hace mediante plantillas de archivos. Dependiendo del tipo de archivo que cree, el IDE inserta el código inicial y el formato que se espera que esté en todos los archivos de ese tipo.



Agregue el método main() usando plantillas en vivo. Coloque el símbolo de intercalación en la cadena de declaración de clase después del corchete de apertura `{` y presione Shift Enter (A diferencia de `,` inicia una nueva línea sin romper la actual).

Escriba `main` y seleccione la plantilla que inserta la `main()` declaración del método.

A medida que escribe, IntelliJ IDEA sugiere varias construcciones que se pueden utilizar en el contexto actual.



Las plantillas activas son fragmentos de código que puedes insertar en tu código. maines uno de esos fragmentos. Por lo general, las plantillas activas contienen bloques de código que utiliza con más frecuencia. Usarlos puede ahorrarle algo de tiempo ya que no tiene que escribir el mismo código una y otra vez. Después de la `main()` declaración del método, IntelliJ IDEA coloca automáticamente el cursor en la siguiente línea.

Llamemos a un método que imprima texto en la salida estándar del sistema.

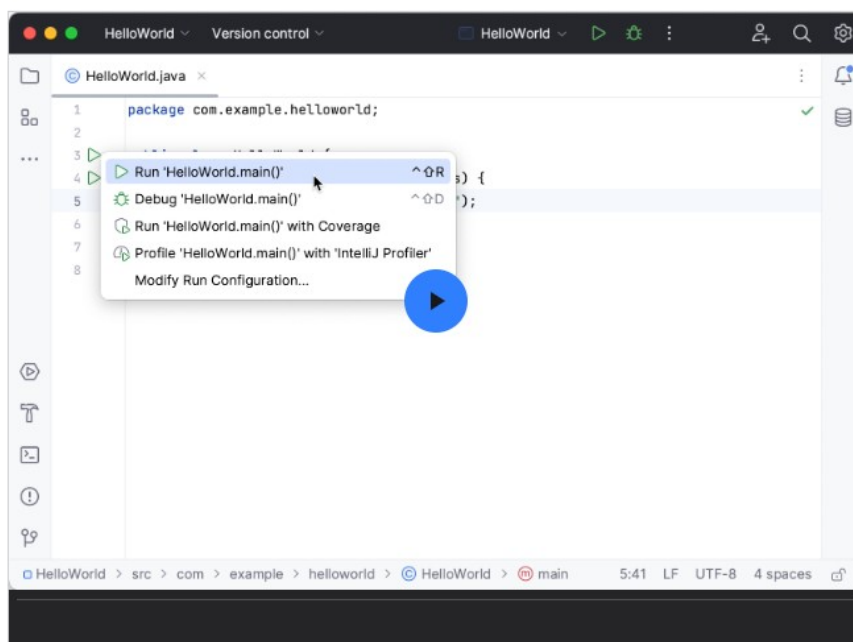
- Escriba `System.out` y **seleccione la System class** de la lista de sugerencias de finalización de código (es del `java.lang` paquete estándar).
- Presione para insertar la selección con un punto final. `Ctrl .`
- Escriba `println`, seleccione `out` y presione nuevamente `Ctrl .`
- Escriba `p`, seleccione el método `println(String x)` y presione `Enter`. IntelliJ IDEA le muestra los tipos de parámetros que se pueden usar en el contexto actual. Esta información es para su referencia.
- Tipo `"`. La segunda comilla se inserta automáticamente y el signo de intercalación se coloca entre las comillas. Tipo `Hello World!`



La finalización de código básico analiza el contexto alrededor de la posición actual del cursor y proporciona sugerencias a medida que escribe. Puede abrir la lista de finalización manualmente presionando `Ctrl Space`.

Las clases Java válidas se pueden compilar en código de bytes. Puede **compilar y ejecutar** clases con el `main()` método **directamente desde el editor** usando el ícono de flecha verde en el margen.

Haga clic ▶ en el canal y seleccione **Ejecutar 'HelloWorld.main()'** en la ventana emergente. El IDE comienza a compilar su código.



Cuando se completa la compilación, se abre la ventana de la herramienta **Ejecutar** en la parte inferior de la pantalla.

La primera línea muestra el comando que IntelliJ IDEA utilizó para ejecutar la clase compilada. La segunda línea muestra la salida del programa: `Hello World!`. Y la última línea muestra el código de salida `0`, que indica que salió exitosamente.

Si su código no es correcto y el IDE no puede compilarlo, la ventana de la herramienta Ejecutar mostrará el código de salida correspondiente.

Cuando hace clic en Ejecutar , IntelliJ IDEA crea una configuración de ejecución especial que realiza una serie de acciones. Primero, construye su aplicación. En esta etapa, **javac compila su código fuente** en código de bytes JVM.

Una vez que javac termina la compilación, coloca el código de bytes compilado en el directorio de salida , que está resaltado en amarillo en la ventana de herramientas Proyecto .

Después de eso, la **JVM ejecuta el código de bytes**.

IntelliJ IDEA analiza automáticamente el archivo que está abierto actualmente en **el editor y busca diferentes tipos de problemas**: desde errores de sintaxis hasta errores tipográficos. El **widget de Inspecciones** en la esquina superior derecha del editor le permite ver rápidamente todos los problemas detectados y observar cada problema en detalle.

☞ Añadir y eliminar un plugin

Haga clic en **Complementos** en el panel izquierdo y descargue e instale o desinstale complementos adicionales desde **JetBrains Marketplace** (<https://plugins.jetbrains.com/idea>) .

