

ENTORNOS DE DESARROLLO **DAM 1**

**UT 03 INSTALACIÓN Y USO DE ENTORNOS DE DESARROLLO**

Índice

[1 ¿Qué es un IDE? 4](#_Toc148267773)

[1.1 Tipos de IDE 4](#_Toc148267774)

[1.2 Funciones y componentes de un IDE 4](#_Toc148267775)

[2 Uso de Geany 6](#_Toc148267776)

[2.1 Ambiente de trabajo 7](#_Toc148267777)

[2.2 Resaltado de sintaxis 8](#_Toc148267778)

[2.3 Plegado de código 8](#_Toc148267779)

[2.4 Autocompletado de palabras y símbolos 10](#_Toc148267780)

[2.5 Algunas características útiles adicionales 11](#_Toc148267781)

[2.6 Combinaciones de teclas 12](#_Toc148267782)

[2.7 Crear, compilar y probar nuestro código 15](#_Toc148267783)

[2.8 Ejercicios 18](#_Toc148267784)

[2.8.1 Ejercicio 1: 18](#_Toc148267785)

[2.8.2 Ejercicio 2: 19](#_Toc148267786)

[2.8.3 Ejercicio 3: 19](#_Toc148267787)

[2.8.4 Ejercicio 4: 20](#_Toc148267788)

[3 Uso de Visual Studio Code 21](#_Toc148267789)

[3.1 Ambiente de trabajo 22](#_Toc148267790)

[3.2 Cambiar el tema de color 25](#_Toc148267791)

[3.3 Instalación de extensiones 25](#_Toc148267792)

[3.4 Otras características de Visual Studio Code 27](#_Toc148267793)

[3.4.1 Edición en paralelo 28](#_Toc148267794)

[3.4.2 Selección de caja 29](#_Toc148267795)

[3.4.3 Formatear el código 30](#_Toc148267796)

[3.4.4 Más características 30](#_Toc148267797)

[3.5 Combinaciones de teclas 31](#_Toc148267798)

[3.5.1 Ejercicio 1: 32](#_Toc148267799)

[3.5.2 Ejercicio 2: 32](#_Toc148267800)

[3.5.3 Ejercicio 3: 33](#_Toc148267801)

[3.5.4 Ejercicio 4: 34](#_Toc148267802)

[3.5.5 Ejercicio 5: 34](#_Toc148267803)

[4 Uso de IntelliJ IDEA 35](#_Toc148267804)

[4.1 Instalación y configuración 35](#_Toc148267805)

[4.2 Creación de proyectos Java 37](#_Toc148267806)

[4.3 Combinaciones de teclas 41](#_Toc148267807)

[4.3.1 Ejercicio 1: 42](#_Toc148267808)

[4.3.2 Ejercicio 2: 42](#_Toc148267809)

[5 Uso de Visual Studio 43](#_Toc148267810)

[5.1 Creación de proyectos 43](#_Toc148267811)

[5.2 Combinaciones de teclas 45](#_Toc148267812)

[5.3 Otras características 45](#_Toc148267813)

[5.3.1 Ejercicio 1 46](#_Toc148267814)

# ¿Qué es un IDE?

Un entorno de desarrollo es un conjunto de procesos y herramientas de programación que utilizamos para crear software. Un entorno de desarrollo integrado o IDE es un entorno en el que los procesos y las herramientas se coordinan, y proporcionan una interfaz común para el proceso de desarrollo de software (o, al menos, para escribir el código y ejecutarlo).

Los primeros IDE nacieron a principios de los años 70, pero se hicieron muy populares a lo largo de los años 90. Su primer objetivo es ahorrar tiempo y tener más fiabilidad en los procesos de software. Proporcionan al programador algunos componentes útiles que aumentan la eficiencia y reducen el tiempo de codificación.

Algunos ejemplos de estos IDE son Geany, Visual Studio Code, NetBeans, Eclipse, Visual Studio, IntelliJ... En esta unidad nos vamos a centrar en algunos de ellos.

## Tipos de IDE

Hay dos tipos principales de IDE:

**IDE de propósito general**: pueden funcionar con muchos lenguajes de programación, y son (generalmente) más simples y limitados a las características. Ejemplos de estos IDE son Geany, Visual Studio Code, Sublime Text...

**IDE de propósito específico**: generalmente funcionan con uno (o un par de) lenguajes de programación, por lo que son entornos muy específicos y proporcionan un mayor conjunto de características para este lenguaje. Ejemplos de estos IDE son IntelliJ, Visual Studio, NetBeans, Eclipse

## Funciones y componentes de un IDE

Los entornos de desarrollo están compuestos por algunas herramientas de programación útiles que nos ayudan a cumplir nuestros objetivos.

Estas herramientas pueden ser:

* Editor de código fuente: es una parte esencial de todo IDE, ya que nos permite escribir el código fuente de nuestro programa. Por lo general, resalta la sintaxis del código, y también puede tener algunas características adicionales, como autocompletado de código, sangría automática, ayuda contextual, acceso a la documentación de la API, inserción automática de paréntesis, corchetes...
* Compiladores e intérpretes: estas herramientas se encargan de traducir nuestro código fuente a código máquina. Algunos IDE nos permiten agregar muchos compiladores o intérpretes diferentes para muchos lenguajes diferentes, como veremos más adelante.
* Depurador: se encarga de depurar nuestro código fuente. En otras palabras, nos permite examinar paso a paso la ejecución de nuestro programa, comprobar los valores de las diferentes variables y detener el programa en cualquier punto.
* Generador de GUI: simplifica la creación de interfaces gráficas de usuario (GUI), para que podamos colocar los diferentes controles (botones, listas, menús...) donde queramos.
* Control de versiones: con las herramientas de control de versiones podemos gestionar cada cambio en el código fuente de nuestros proyectos, de forma que tendremos diferentes versiones de cada archivo fuente, y podremos volver atrás y recuperar cualquier versión anterior en cualquier momento. Es una herramienta esencial para el trabajo en equipo.

# Uso de Geany

Geany es un editor de texto fácil con algunas características básicas de un IDE. Se puede descargar de forma gratuita desde su web oficial, donde encontraremos versiones para Windows y Mac.

## Instalación:

Instalaremos la versión Linux desde la Terminal:

sudo apt update

En primer lugar, instalamos java development kit:

sudo apt install default-jdk

*(que nos permite compilar desde .java a .class y ejecutar, con la máquina virtual de java, los ficheros .class)*

En segundo lugar, instalamos el compilador de c gcc:

sudo apt install gcc

A continuación, instalamos Geany:

sudo apt install geany

Compilamos con F8: Construir -> Compilar

Ejecutamos con F5: Construir -> Ejecutar

*Si en cualquier momento apt install falla con error: “bloqueo de cache… lo mantiene el proceso ####”, matar el proceso número #### con el comando:*

sudo kill -9 ####

## Ambiente de trabajo

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

El entorno de trabajo de Geany tiene las siguientes secciones:

1. Menú.
2. Barra de herramientas (opcional)
3. Barra lateral con pestañas:
   1. Documentos → Lista de documentos.
   2. Símbolos → Lista de símbolos de código.
4. Ventana Edición.
5. Ventana de mensaje, con las siguientes pestañas:

* Status → Una lista de mensajes de estado.
  1. Mensajes del compilador → del compilador.
  2. Mensajes → Mensajes generales.
  3. Garabatear (Borrador)→ Para hacer anotaciones.

1. Barra de estado, donde podemos ver, entre otras cosas, la fila y columna que estamos editando, el tipo de archivo, la codificación

## Resaltado de sintaxis

Esta es una característica muy útil cuando estamos escribiendo código, ya que podemos detectar fácilmente algunos errores tipográficos básicos a medida que escribimos el código. Además, nos ayuda a entender todo el código con un simple vistazo.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Como podemos ver en la imagen anterior, las palabras reservadas están escritas en un color diferente (azul). Si escribimos una palabra reservada y no está escrita en azul, entonces es posible que no la hayamos escrito correctamente, por lo que podemos verificarla de inmediato, sin esperar el error de compilación.

## Plegado de código

Geany proporciona un código básico de plegado. Este plegado nos permite mostrar u ocultar algunas partes del código, de modo que podamos centrarnos solo en las partes que estamos editando actualmente. Esta función se puede activar o desactivar desde *el menú Editar/Preferencias*.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Desde este menú, también podemos habilitar la posibilidad de plegar y desplegar todos los plegados anidados que están dentro del bloque de código que estamos tratando de plegar/desplegar. De forma predeterminada, esta opción está desactivada, por lo que tenemos que plegar / desplegar cada bloque de código de forma independiente.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

## Autocompletado de palabras y símbolos

Esta opción se puede configurar en el menú Edición/Preferencias, en la opción Editor, en la pestaña Finalizaciones.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Con esta función habilitada, veremos una lista de opciones adecuadas cada vez que escribamos el comienzo de una palabra reservada, por lo que podremos completarla simplemente escribiendo ***Enter***. Por defecto está habilitado para mostrar la lista cuando escribimos las 4 primeras letras de la palabra reservada. Además, también podemos configurar la finalización automática de paréntesis, llaves, corchetes, etc. Esto es realmente útil si generalmente nos olvidamos de cerrar estos símbolos.

## Algunas características útiles adicionales

Además de las principales características útiles mostradas anteriormente, también podemos destacar las siguientes características de Geany:

* Cierre automático de etiquetas XML/HTML: en el menú Editar/Preferencias, en la opción Editor, también podemos activar el cierre automático de etiquetas XML/HTML cuando escribimos la etiqueta de apertura correspondiente.
* Soporte multilenguaje: Geany nos permite elegir entre muchos lenguajes de programación diferentes, como C, Java, PHP, HTML, Python, C # y otros, con su propio autocompletado y resaltado de sintaxis. Podemos seleccionar el idioma del menú Documento > Establecer tipo de archivo, eligiendo el lenguaje de programación adecuado.
* Inicio de Geany: cuando ejecutamos Geany, todos los documentos abiertos de nuestra sesión anterior se abrirán nuevamente. Esta función se puede configurar en el menú Editar/Preferencias, en la opción General, en la pestaña Inicio.
* Clonar documentos: en el menú Documento hay una opción Clonar que nos permite no solo copiar el texto del documento actual, sino también sus propiedades, como la posición del cursor. Puede ser realmente útil para crear muchos archivos de origen con una estructura similar.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

## Combinaciones de teclas

Hay muchos atajos o enlaces de teclas disponibles en Geany, como también podemos ver en muchas otras aplicaciones. En las siguientes tablas podemos ver algunas de las más comunes. Estos accesos directos también se pueden configurar desde el menú *Preferencias de edición*, en la pestaña *Combinaciones*.

**Accesos directos a archivos**

|  |  |  |
| --- | --- | --- |
| Opción | Atajo | Acción |
| Nuevo | Ctrl-N | Crea un nuevo archivo |
| Abrir | Ctrl-O | Abre un archivo |
| Salvar | Ctrl-S | Guarda el archivo actual |
| Guardar todo | Ctrl-Mayús-S | Guarda todos los archivos abiertos |
| Cerrar | Ctrl-W | Cierra el archivo actual |
| Salida | Ctrl-Q | Salida de Geany |

**Editar accesos directos**

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Atajos de formato**

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Además de estos atajos, puedes consultar todos los posibles [aquí](https://www.geany.org/manual/current/index.html#keybindings).

## Crear, compilar y probar nuestro código

Una vez que hayamos terminado de escribir nuestro código fuente, podemos probarlo desde Geany, utilizando el menú *Construir*.

Interfaz de usuario gráfica, Aplicación, Correo electrónico

Descripción generada automáticamente

Si vamos a la opción Build /Set Build Commands, podemos ver el compilador que se ejecutará cuando intentemos compilar el programa. (*Construir → Establecer comandos de construcción)*

Siempre que creamos un nuevo documento con Geany, es importante guardarlo con un nombre y extensión apropiados antes de escribir cualquier código en él. De esta manera, Geany rellenará automáticamente los comandos de compilación y ejecución correspondientes con las herramientas adecuadas. Además, solo después de detectar el tipo de código fuente que estamos escribiendo, se activará el resaltado de código y el autocompletado

Hay compiladores de detección automática para C # y HTML (este último es un lenguaje interpretado, por lo que no tiene compilador):

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

También hay compiladores auto detectados para C y Java:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

También podemos configurar el editor para subrayar con color rojo cada error de compilación. Esta opción está disponible en el menú Editar/Preferencias, dentro de la opción Editor, en la pestaña Características.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Como podemos ver en la imagen de arriba, podemos verificar cada error de compilación en la ventana de mensajes del compilador en la parte inferior de la ventana.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

En la imagen anterior podemos ver cómo usar diferentes tipos de código fuente, y cada uno tiene su propio compilador asignado, por lo que podemos compilar cada documento con su compilador asociado.

## Ejercicios

### Ejercicio 1:

Abra Geany y cree un nuevo archivo fuente llamado Test.java con el siguiente código fuente:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Vea cómo Geany resalta el código (*Puede que no utilice los mismos colores que en el ejemplo*) Además, marque o habilite el autocompletado y pruébelo agregando una nueva línea con el código para escribir "Mundo" (simplemente copie la línea que dice "Hola" y reemplácela con "Mundo").

### Ejercicio 2:

Compile el archivo fuente del ejercicio anterior. Si no ha habido errores, ejecútelo.

### Ejercicio 3:

Cree un archivo llamado test.c con Geany. Recuerde: después de elegir *Archivo-Nuevo*, elija Archivo*-Guardar como* y guárdelo con la extensión .c, de modo que el código se resaltará a medida que lo escribamos. A continuación, escriba el siguiente código.

Forma

Descripción generada automáticamente con confianza bajaVerifique el autocompletado del código.

Compile completamente el programa con Construir → Build F9, y ejecútelo F5.

### Ejercicio 4:

Pruebe cuatro de los atajos de teclado anteriores de Geany con cualquiera de los archivos fuente creados hasta ahora.

Documente los atajos que ha utilizado, explicando su uso e insertando en la memoria capturas de pantalla del resultado.

# Uso de Visual Studio Code

Imagen que contiene dibujo

Descripción generada automáticamente

Visual Studio Code es un editor de código ligero (pero potente) disponible para Windows, MacOSX y Linux. Tiene soporte nativo para Javascript, TypeScript y Node.js, y una amplia variedad de extensiones que nos permiten trabajar con (casi) cualquier otro lenguaje de programación, como C, C#, Java, Python, PHP, Go, etc.

Puedes descargarlo desde su [web oficial](https://code.visualstudio.com/), donde también podrás encontrar algunas extensiones útiles para Java, C, C#..

Instala Java J**DK**

**También debes instalar el compilador GCC:**

Sigue las siguientes instrucciones: <https://code.visualstudio.com/docs/cpp/config-mingw>

En el punto 3, no olvide anotar el directorio de instalación.

En el punto 7, introduce ***el directorio anterior*** en la variable PATH.

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente Interfaz de usuario gráfica, Texto

Descripción generada automáticamente



Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente



NOTA: para los usuarios de Linux (Ubuntu), una vez que descargue el paquete .deb desde el sitio web oficial, debe abrir un terminal e ir a la carpeta de descarga. A continuación, escriba este comando y tendrá Visual Studio Code disponible en la sección Programación:

sudo dpkg -i <visual\_studio\_file\_name>

donde

<visual\_studio\_file\_name>

es el nombre del archivo descargado.

## Ambiente de trabajo

Una vez instalado, puede ver la pantalla de bienvenida con algunas de las opciones básicas

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Si creamos un nuevo archivo veremos el entorno de trabajo habitual, que se divide en 5 secciones.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* Editor (C): Área de edición principal. Puede abrir tantos editores como necesite.
* Barra de margen (B): Contiene información útil sobre el explorador de archivos, errores de código o advertencias, etc.
* Barra de estado (E): muestra información sobre el proyecto actual o los archivos abiertos actualmente.
* Barra de actividades (A): Tiene algunas opciones:
  + Explorador de archivos, para localizar los archivos a editar. Esta opción muestra/oculta un panel izquierdo para examinar la carpeta actual
  + Herramienta de búsqueda, para buscar algunas palabras o expresiones regulares en nuestros archivos fuente.
  + Control de código fuente (para comunicarse con herramientas de control de versiones, como repositorios Git)
  + Depurador
  + Gestor de extensiones, desde el que podemos instalar new extensiones a este IDE, y comprobar/configurar las que ya están instaladas.
* Paneles (D): Debajo del editor puede ver algunos paneles diferentes, como errores / advertencias, o un terminal para escribir algunos comandos de la carpeta actual. Este panel se puede mover ala derecha si queremos tener más espacio vertical.

Cada vez que abrimos Visual Studio Code, muestra la última instantánea antes de que se cerrara la última vez, con los mismos archivos abiertos, etc.

## Cambiar el color del tema

Si desea cambiar el color del tema predeterminado para Visual Studio Code, debe ir al menú Archivo/Preferencias (o al menú Código/Preferencias si lo está ejecutando en Mac OSX) y, a continuación, elegir la opción Tema de color. Luego, puede elegir entre una amplia variedad de opciones. Los más populares son los modos oscuros o claro de Visual Studio.

## Instalación de extensiones

Una de las características más destacadas de Visual Studio Code es que podemos mejorar su rendimiento instalando extensiones adicionales. Muchas de estas extensiones se refieren a un lenguaje de programación en particular, por lo que podemos enriquecer el IDE para un lenguaje específico.

Si queremos comenzar a codificar, necesitamos instalar la(s) extensión(es) adecuada(s) para el lenguaje con el que vamos a trabajar. Para ello, pulsamos sobre el icono de extensión en la barra de actividades (barra izquierda), y buscamos el idioma deseado en el campo de texto. Por ejemplo, si queremos trabajar con el lenguaje de programación Java, simplemente escribimos Java y luego veremos muchas extensiones relacionadas con este lenguaje.

DEBES INSTALAR una extensión útil llamada Java Extension Pack, de Microsoft, que contiene los archivos más comunes para trabajar con Java:

* Soporte de lenguaje para Java (TM) de Red Hat
* Depurador para Java
* Ejecutor de pruebas de Java
* Maven para Java

**Posible fallo: “No encuentra javac”:**

1. Desinstala Java con el gestor de programas de Windows
2. Reinicia
3. Instala Java J**DK**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Además, es posible que necesites la extensión Code Runner, una extensión general que nos permite ejecutar programas escritos en muchos lenguajes de programación diferentes, como C, Python, Java, etc.

También debe ir al menú Archivo / Preferencias / Configuración, ir a la sección Extensiones y elegir Configuración de código de ejecución. Luego, busque la opción Ejecutar en terminal y verifíquelo.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**También debes instalar el compilador GCC:**

Sigue las siguientes instrucciones: <https://code.visualstudio.com/docs/cpp/config-mingw>

En el punto 3, no olvide anotar el directorio de instalación.

En el punto 7, introduce el directorio anterior en la variable PATH.

A continuación, solo necesitamos abrir la carpeta en la que vamos a almacenar los archivos fuente. En el panel izquierdo, veremos el contenido de esta carpeta, y podremos abrir un archivo existente, o crear uno nuevo desde el menú Archivo:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Para ejecutar un solo archivo, podemos simplemente hacer clic derecho en su código fuente y elegir la opción Ejecutar código en el menú contextual (esta opción es proporcionada por la extensión Code Runner, por lo que debe instalar esta extensión para usar esta opción).

De la misma manera que hemos probado nuestro ejemplo de Java, podemos ejecutar programas en muchos otros lenguajes, como C, Python y así sucesivamente. Solo necesitamos descargar la(s) extensión(es) apropiada(s) para trabajar con este lenguaje y ejecutar los archivos fuente con Code Runner.

Sin embargo, hay algunos lenguajes concretos cuya configuración es bastante más difícil. Este es el caso de C#, por ejemplo. Necesitamos instalar el compilador apropiado (ya sea Mono o .NET), pero debemos seguir algunos pasos adicionales en cada caso para configurar todo correctamente. En este caso, puede ser una mejor opción usar Geany para archivos de código fuente único o Visual Studio para proyectos complejos.

## Otras características de Visual Studio Code

Visual Studio Code tiene las mismas características vistas antes con Geany con respecto a:

* Resaltado de sintaxis
* Plegado de código
* Autocompletado de código mediante IntelliSense
* Cierre automático de etiquetas XML/HTML

La configuración se puede cambiar en Archivo / Preferencias / Configuración.

Pero, además de estas características básicas, Visual Studio Code tiene algunas características más avanzadas que Geany no tiene.

### Edición en paralelo

Esta característica nos permite editar dos o más archivos fuente al mismo tiempo, teniendo cada uno en una columna diferente. De esta forma, podremos editarlos y compilarlos sin tener que cerrar ninguno de ellos. Esta opción se activa haciendo clic en el icono de 2 columnas en la barra superior derecha.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Si queremos compilar alguno de estos archivos, como tenemos más de un documento activo, se mostrará un menú emergente para elegir el archivo a compilar.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

### Selección de caja

Si queremos hacer este tipo de selección, debemos mantener pulsada la tecla Mayús+Alt mientras seleccionamos la(s) columna(s) deseada(s) con el ratón.

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

### Formatear el código

Hay una opción para formatear el código seleccionado actualmente (o todo el archivo). Esta opción se encuentra en el menú contextual del editor (haciendo clic derecho en el panel del editor), con el nombre Formatear documento. Formatea el código de acuerdo con el lenguaje que estamos usando. Por ejemplo, para un programa Java tendremos este formato:

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente con confianza media

Podemos ver que las llaves de apertura se han colocado junto a la línea que crea el bloque (en lugar de ponerlas en la siguiente línea). Este es el formato que suelen tener los archivos fuente de Java.

### Más características

Hay algunas otras características adicionales que podríamos mencionar:

* Autoguardado: por defecto, VS Code nos pide que guardemos manualmente nuestros cambios, pero hay una opción en el menú Archivo llamada Auto Save. Si está habilitado, el IDE guardará periódicamente nuestro trabajo (o cada vez que el editor pierda el foco).
* Salida caliente: cada vez que salgamos de VS Code sin guardar cambios, este estado se guarda automáticamente, y cuando volvamos a abrir el IDE, mantendremos todos nuestros cambios no guardados.
* Sangría de detección automática: cada vez que abrimos un archivo fuente, se detecta y utiliza automáticamente su propia sangría, en lugar de la establecida por defecto en VS Code. Esta opción se muestra (y se puede modificar) en la barra de estado (tamaño de tabulación).

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

## Combinaciones de teclas

En la pantalla de bienvenida tenemos un enlace con una lista completa de accesos directos para VS Code. Algunos de ellos son similares a otros editores, y algunos otros son particulares para este IDE.

Tabla

Descripción generada automáticamente

### Ejercicio 1:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteAbra Visual Studio Code y cree un nuevo archivo de código fuente denominado Test.java con el siguiente código fuente:

* Instale las extensiones indicadas.
* Vea cómo VS Code resalta el código.
* Compile y ejecute el programa **dándole al play, arriba a la derecha.**

Interfaz de usuario gráfica, Aplicación, Correo electrónico

Descripción generada automáticamente

### Ejercicio 2:

Cree un archivo denominado test.c con Visual Studio Code con el siguiente contenido:

Forma

Descripción generada automáticamente con confianza baja

Compile y ejecute el programa con el botón play:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

### Ejercicio 3:

Pruebe las características de Visual Studio, como la **selección de caja** y el **cursor múltiple**, con cualquiera de los archivos mencionados en ejercicios anteriores.

Para ello, tendrás que extenderlos de manera que, al menos, tengas 10 líneas de código efectivamente ejecutables.

### Ejercicio 4:

Cree un archivo llamado HolaMundoWeb.html e introduzca el siguiente código html:

Captura de pantalla de computadora

Descripción generada automáticamente

Guarda los cambios y abre el archivo con un navegador web

¿Qué ocurre si cambio el contenido de la etiqueta <title>?

¿Qué ocurre si cambio el número que hay junto a la h en la etiqueta <h1>?

¿Qué tengo que cambiar para que el mensaje mostrado en la página web sea “Hola querido amigo”?

### Ejercicio 5:

Configura el editor VSC para que se vea en modo claro y en modo oscuro.

Instala los siguientes plugins en VSC y muestra su uso:

* Html preview
* Auto rename tag

# Uso de IntelliJ IDEA

Un dibujo con letras

Descripción generada automáticamente con confianza baja

IntelliJ IDEA es un IDE multiplataforma desarrollado por JetBrains. Fue lanzado por primera vez en 2001 como IntelliJ, y fue uno de los primeros IDE con navegación avanzada y refactorización de código. Podemos considerarlo un IDE de propósito específico, ya que se centra en unos pocos grupos de lenguajes de programación.

Está disponible en dos versiones: Community Edition (gratuita) y Ultimate Edition (comercial). La principal diferencia entre ellos se puede encontrar en los idiomas y sistemas de control de versiones compatibles. Por ejemplo, la versión de la comunidad no permite PHP o Javascript. En nuestro caso, como vamos a trabajar con Java Virtual Machine, necesitamos descargar la edición Community, que es gratuita y de código abierto. Esta es la página de [descarga](http://www.jetbrains.com/idea/download/).

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

## Instalación y configuración

En cuanto al proceso de instalación:

* En Windows, tenemos un asistente paso a paso que nos guía a través de todo el proceso de instalación. Podemos elegir la carpeta de instalación (o simplemente dejar la que tenemos configurada por defecto), y si queremos crear un acceso directo en el escritorio. También podemos marcar una opción para agregar un menú contextual para abrir el proyecto IntelliJ desde el explorador. Esta opción puede ser muy útil.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

* En Mac OSX, tenemos un instalador que nos pide que arrastremos la aplicación a la carpeta Aplicaciones.
* En Linux, descargamos un archivo tar.gz que debemos descomprimir. Dentro de la carpeta principal hay una subcarpeta bin. Debemos entrar en esta carpeta desde un terminal y ejecutar el comando. /idea.sh para lanzar el IDE. La primera vez que lo ejecutemos, creará un acceso directo en algún lugar del menú de aplicaciones, por lo que podemos iniciar IntelliJ usando este acceso directo a partir de ese momento.

La primera vez que ejecutamos *IntelliJ*, nos permite importar configuraciones anteriores, si teníamos alguna versión anterior instalada. Si no, podemos elegir "*No importar configuraciones*". A continuación, podemos elegir el tema de la interfaz de usuario ...

Texto

Descripción generada automáticamente

A continuación, nos permite definir un script para lanzar programas desde la línea de comandos, pero podemos saltarnos este paso y pasar al siguiente, en el que podemos optar por instalar algunos plugins adicionales. También podemos dejar este paso con su configuración predeterminada y comience a usar IntelliJ.

## Creación de proyectos Java

Desde la pantalla de bienvenida, podemos elegir entre:

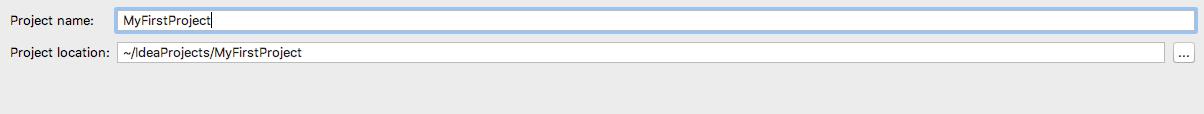
* Creación de un nuevo proyecto
* Abrir un proyecto existente desde nuestro ordenador
* Obtener un proyecto de un repositorio remoto mediante sistemas de control de versiones (VCS)

Entonces, si queremos crear un nuevo proyecto, elegimos la opción *Nuevo* proyecto, y luego especificamos que queremos crear un proyecto Java, desde el panel izquierdo:

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

A continuación, podemos elegir una plantilla para nuestro proyecto con el fin de generar algún código por defecto, pero, como no tenemos ninguna plantilla anterior, podemos saltarnos este paso, y pasar al siguiente, en el que debemos especificar un nombre y ubicación del proyecto (podemos dejar la ubicación predeterminada si queremos):



NOTA: de forma predeterminada, IntelliJ almacena los proyectos dentro de una carpeta llamada IdeaProjects dentro de nuestra carpeta de inicio. A cada proyecto se le asigna una subcarpeta dentro de esta carpeta principal.

Luego, hacemos clic en Finalizar y veremos nuestro proyecto. Si hacemos clic en la pestaña del proyecto a la izquierda, podemos ver la estructura de la carpeta del proyecto y crear elementos (archivos fuente) en ella.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ahora, vamos a crear nuestro primer archivo fuente. Haga clic derecho en la carpeta src y elija New > Java Class.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Luego, debemos especificar el nombre de la clase. Por ejemplo, Hello.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Se creará un nuevo archivo, y podremos editarlo en el área principal. Podemos dejar un código como este:

Interfaz de usuario gráfica, Texto, Aplicación, Teams

Descripción generada automáticamente

Podemos ejecutar el archivo haciendo clic derecho sobre él y eligiendo la opción "Ejecutar Hello.main()":

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Luego, podemos verificar los resultados en el terminal incrustado en la parte inferior de la ventana:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

## Combinaciones de teclas

En la tabla siguiente puede ver algunos de los enlaces de teclas o accesos directos más comunes disponibles para IntelliJ IDEA en sistemas Windows.

|  |  |  |
| --- | --- | --- |
| Atajo |  | Acción |
| Ctrl + Mayús + N | | Abra un nuevo archivo. |
| Ctrl + | N | Abra cualquier clase. |
| Ctrl + Barra espaciadora | | Código completo. |
| Ctrl + Mayús + Barra espaciadora | | Finalización inteligente de código. |
| Ctrl + | S | Guardar archivo. |
| Ctrl + O | | Métodos de sobrescritura. |
| Ctrl + | Yo | Implementar todo. |
| Ctrl + / | | Línea Comentar / Descomentar. |
| Ctrl + | D | Línea duplicada. |
| Ctrl + | Z | Deshacer la última acción. |
| Ctrl + Mayús + Z | | Rehacer la última acción deshecha. |
| Ctrl + | F | Mostrar cuadro de diálogo de búsqueda. |
| Ctrl + | R | Mostrar cuadro de diálogo de reemplazo. |
| Ctrl + | F9 | Compilar proyecto. |
| Mayús + F10 | | Ejecutar proyecto. |
| Mayús + F9 | | Depurar. |
| F7 |  | Paso a la función (en modo *de depuración* ). |
| F8 |  | Línea siguiente (en modo *de depuración* ). |
| F9 |  | Detenga la depuración. |
| Ctrl + | F8 | Crear punto de interrupción. |
| Ctrl + Mayús + F12 | | Maximice el panel del editor. |

La mayoría de estos accesos directos también están disponibles en Linux. Con respecto a los sistemas MacOSX, debe reemplazar la tecla Ctrl con la clave Cmd. Puedes encontrar más atajos aquí.

### Ejercicio 1:

Cree un nuevo proyecto Java con IntelliJ denominado Test y copie este código en una clase denominada Test:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Luego, ejecute el programa para verificar si todo está bien.

### Ejercicio 2:

Explica la utilidad de los siguientes 4 atajos de teclado, mostrando además una captura:

* Ctrl + Z
* Ctrl + /
* Ctrl + D
* Mayus + F10

# Uso de Visual Studio

Un dibujo de una cara feliz

Descripción generada automáticamente con confianza baja

Visual Studio fue un IDE creado en 1997. Inicialmente fue creado para desarrollar aplicaciones C++ y Visual Basic bajo Windows, pero ha evolucionado y ahora incluye algunos otros lenguajes, como C#, ASP.NET... En general, puede desarrollar cualquier tipo de aplicación soportada por la plataforma .NET, pero también hay algunos otros lenguajes que también son compatibles, como Java o Python, entre otros.

Existen diferentes distribuciones para Visual Studio: Community, Professional o Enterprise. El primero es gratuito, y puedes descargarlo [aquí](https://visualstudio.microsoft.com/es/vs/). Está disponible para sistemas Windows y MacOSX. Una vez que ejecute el instalador, debe elegir su carga de trabajo. Por ejemplo, si está deseando desarrollar aplicaciones de escritorio o de consola con C# y formularios Windows Forms, puede elegir el desarrollo de escritorio .NET.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

## Creación de proyectos

Si queremos crear un nuevo proyecto, vamos al menú Archivo > Nuevo proyecto. Luego, generalmente elegimos una aplicación de consola de C #:

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

En el formulario inferior, debemos especificar el nombre y la ubicación del proyecto (podemos dejar la ubicación predeterminada). Luego, se mostrará un nuevo proyecto, con un archivo fuente inicial predeterminado llamado Programa.cs, con algún código predeterminado ya escrito en él:

Texto

Descripción generada automáticamente

Para ejecutar el programa, simplemente hacemos clic en el botón Inicio en la barra de herramientas, o presionamos F5, o Ctrl + F5 si desea que el programa se detenga después de terminar, antes de cerrar el terminal.

## Combinaciones de teclas

En las tablas siguientes puede ver algunos de los accesos directos o combinaciones de teclas más comunes de Visual Studio (en Windows).

|  |  |  |
| --- | --- | --- |
| Atajo |  | Acción |
| Ctrl + Mayús + N | | Crear un nuevo proyecto. |
| Ctrl + | N | Cree un nuevo archivo. |
| Ctrl + | S | Guarde el archivo actual. |
| Ctrl + Mayús + S | | Guarde todos los archivos abiertos. |
| Ctrl + C / V / X | | Copiar, pegar y cortar texto. |
| Ctrl + | Con | Deshacer la última acción. |
| Ctrl + | Y | Rehacer la última acción deshecha. |
| Ctrl + | F | Mostrar cuadro de diálogo de búsqueda. |
| Ctrl + | H | Mostrar cuadro de diálogo de reemplazo. |
| Atajo |  | Acción |
| Ctrl + | L | Quitar línea. |
| Ctrl + R | | Cambie el nombre del elemento seleccionado. |
| F5 / Ctrl + F5 | | Ejecute la aplicación con/sin depuración. |
| Ctrl + Mayús + B | | Vuelva a generar el proyecto. |
| F11 / F10 | | Paso a la función / siguiente paso (en modo *de depuración* ). |
| Mayús + F11 | | Función de salida (en modo *de depuración* ). |

## Otras características

Hay otras configuraciones que se pueden configurar desde el menú Herramientas > Opciones. Por ejemplo, podemos mostrar/ocultar los números de línea de la subsección Editor de texto, para cada lenguaje específico (C#, Basic...), o las líneas verticales de sangría para ver fácilmente los límites de una instrucción if o for...

### Ejercicio 1

Crea un programa de tipo consola, que ejecute una serie de instrucciones tipo “Hola Mundo” en el lenguaje de programación C#.

El mensaje que se debe mostrar en la consola debe ser el siguiente:

“Hola Miguel, soy (alumno) y estoy programando en C#”

Sustituye (alumno) por tu nombre y apellidos.