
 <p>PROVINCIA DE BUENOS AIRES DIRECCIÓN GRAL. DE CULTURA Y EDUCACIÓN DIRECCIÓN DE FORMACIÓN PROFESIONAL</p>	 <p>Centro de formación profesional N° 402 de La Plata</p>
<p>Programación Bajo Entornos Avanzados Nivel I Curso: 4043</p>	<p>Apunte 1: Introducción a Java y NetBeans</p>

¿Por qué muchos de los desarrolladores de software eligen Java?

Java ha sido probado, ajustado, ampliado y probado por toda una comunidad. Más de nueve millones de los desarrolladores de Java la convierte en la comunidad de desarrollo de mayor tamaño y más activa del planeta. Con su versatilidad, eficacia y portabilidad, Java se ha convertido en incomparable para los desarrolladores, ya que les permite:

- Escribir software en una plataforma y ejecutarla virtualmente en otra.
- Crear programas que se puedan ejecutar en un explorador y acceder a servicios Web disponibles.
- Desarrollar aplicaciones de servidor para foros en línea, almacenes, encuestas, procesamiento de formularios HTML y mucho más.
- Combinar aplicaciones o servicios que utilizan el lenguaje Java para crear aplicaciones o servicios con un gran nivel de personalización.
- Escribir aplicaciones potentes y eficaces para teléfonos móviles, procesadores remotos, productos de consumo de bajo coste y prácticamente cualquier otro dispositivo con un latido digital.

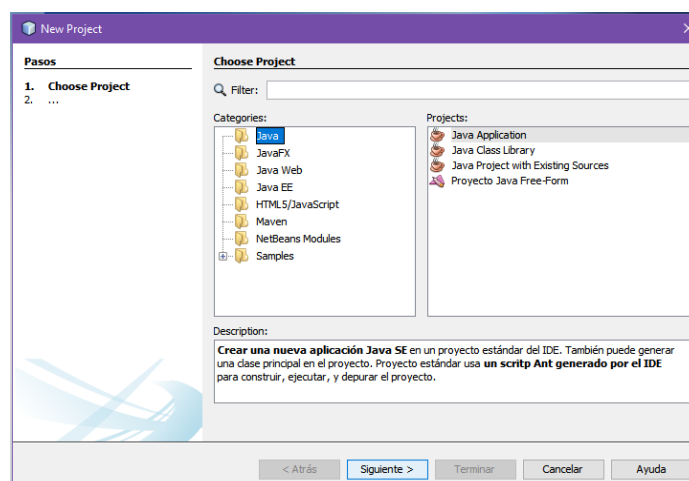
¿Qué software vamos a utilizar para programar en Java?

NetBeans

NetBeans es un entorno de desarrollo integrado libre, principalmente utilizado para la programación Java, aunque existe una gran cantidad de módulos para extenderlo. **NetBeans IDE** es un producto **libre y gratuito, sin restricciones de uso**. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio del año 2000 y continúa siendo el patrocinador principal de los proyectos.

Para instalarlo, instalar previamente el JDK correspondiente a tu S.O (32b o 64b) <http://www.oracle.com/technetwork/es/java/javase/downloads/index.html> y luego podés ir a la página de descargas de NetBeans: <http://netbeans.org/downloads/> o descargar la versión que ya trae el JDK (sólo en inglés): <http://www.oracle.com/technetwork/java/javase/downloads/jdk-netbeans-jsp-142931.html> , seleccionás el idioma, la plataforma y listo.

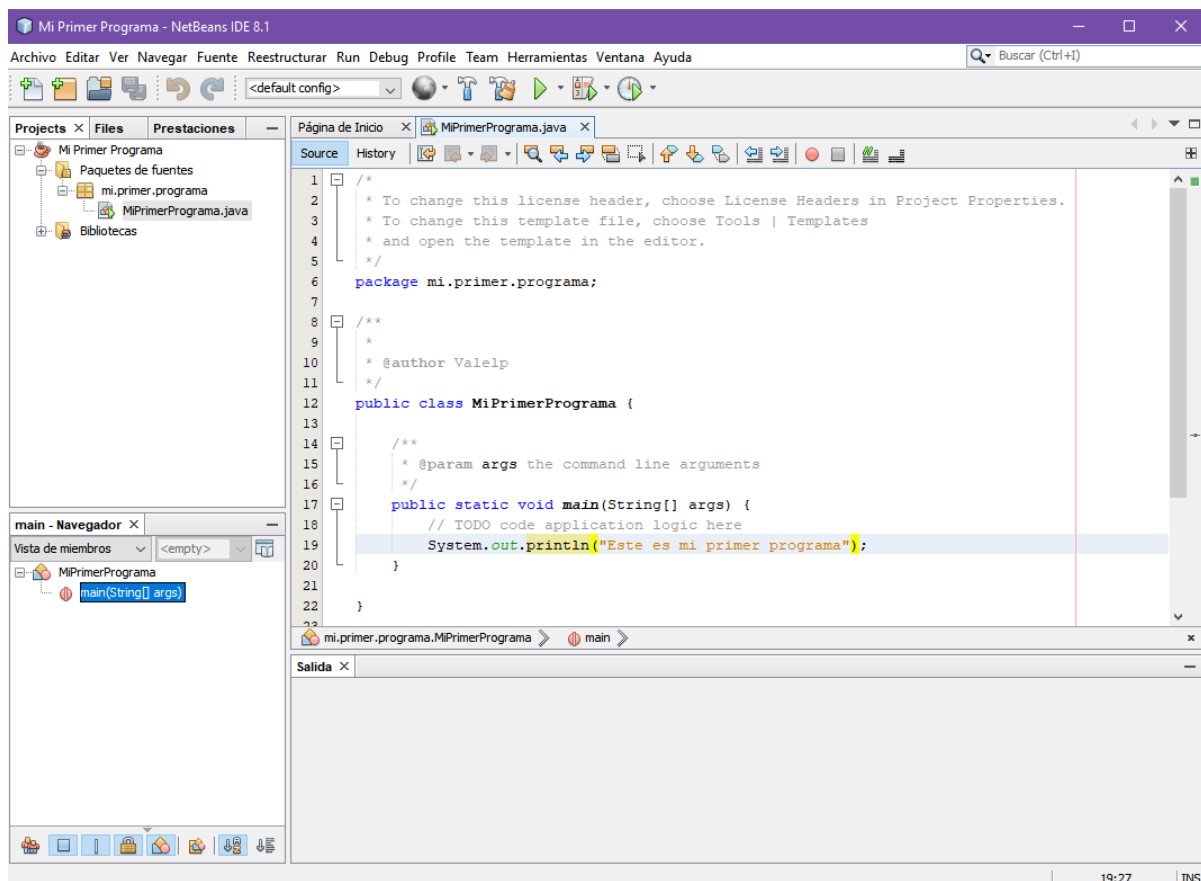
1. Para comenzar, **crearemos un nuevo proyecto**. Para ello, debemos ir a File/New Project. Nos aparecerá una imagen como la siguiente:





2. A continuación, haremos click en Siguiente y podremos darle un nombre al proyecto.

Vamos a llamarlo "Mi Primer Programa" y, la clase principal, será "miprimerprograma.MiPrimerPrograma". El nombre de la clase principal se asigna automáticamente, pero podés cambiarlo.

3. Una vez creado el proyecto, vamos a ver una ventana con lo siguiente:



 PROVINCIA DE BUENOS AIRES DIRECCIÓN GRAL. DE CULTURA Y EDUCACIÓN DIRECCIÓN DE FORMACIÓN PROFESIONAL	 Centro de formación profesional N° 402 de La Plata
Programación Bajo Entornos Avanzados Nivel I Curso: 4043	Apunte 1: Introducción a Java y NetBeans

En la ventana de arriba a la izquierda es donde estarán situados los archivos del proyecto con el que estemos trabajando.

El archivo **MiPrimerPrograma.Java** que se crea al iniciar el proyecto, es el siguiente:

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package miprimerprograma;

/** * * @author valelp */

public class MiPrimerPrograma {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    }

}

```

4. Explicación del código:

4.1. Como podemos observar, en la línea número 10, **declaramos una clase**. Más adelante veremos que, en Java, todo debe estar contenido en una clase. Una clase en Java es donde se definen los atributos y los métodos para los objetos que se van a crear a partir de esa clase. ¿Qué es un atributo? Un atributo es una propiedad que tiene la clase. ¿Qué es un método? Un método define una función de la clase, es decir, lo que hace una clase. Declaramos una clase de la siguiente manera:

```
public class Nombre_clase{ ... }
```

4.2 Debemos tener una **función main**, la cual declararemos de la siguiente manera:

```
public static void main(String[] args){ ... }
```

- **Public** indica que el método main() es público.
- **Void** indica que main() no devuelve ningún valor.
- El método main() debe aceptar siempre como parámetro un **vector de Strings**.
- La **parte principal del programa** irá dentro del main(), es decir, el **TO DO** (“para hacer”).



4.3 Para utilizar otras clases, debemos importarlas antes de nuestra propia clase. Por ejemplo:

```
import java.util.*;

public class Nombre_clase{ public static void main(String[] args){ ... } }
```

Pero, todavía no hemos escrito el Mi Primer Programa, que al fin y al cabo, es el objetivo de este tutorial.

Para escribir en la consola, debemos utilizar lo siguiente:

 PROVINCIA DE BUENOS AIRES DIRECCIÓN GRAL. DE CULTURA Y EDUCACIÓN DIRECCIÓN DE FORMACIÓN PROFESIONAL	 Centro de formación profesional N° 402 de La Plata
Programación Bajo Entornos Avanzados Nivel I Curso: 4043	Apunte 1: Introducción a Java y NetBeans

```
//Para escribir sin saltar de línea System.out.print("mensaje");
```

```
//Para escribir y saltar de línea System.out.println("mensaje");
```

En nuestro caso, el mensaje será "Mi Primer Programa".

Si quisiéramos imprimir una variable con un cierto valor, deberemos utilizar lo siguiente:

```
//Para escribir sin saltar de línea System.out.print(variable);
```

```
//Para escribir y saltar de línea System.out.println(variable);
```

Para concatenar cadenas o variables, utilizaremos un símbolo de suma, el '+'.

```
//Para escribir sin saltar de línea System.out.print("mensaje1 "+" mensaje2 "+ variable);
```

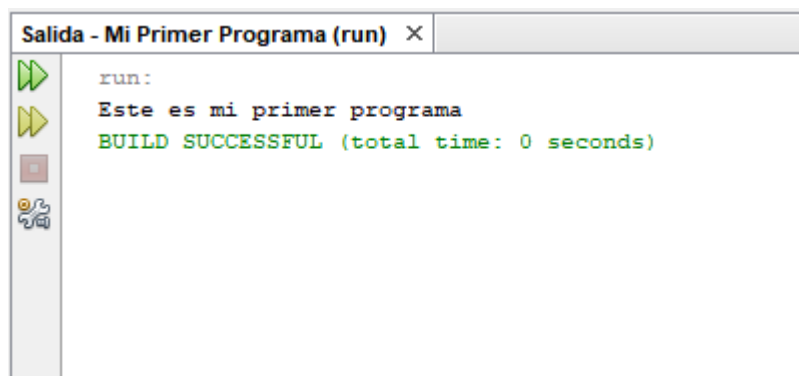
```
//Para escribir y saltar de línea System.out.println("mensaje1 "+" mensaje2 "+ variable);
```

Por lo tanto, nuestro código quedará finalmente así:

```
package miprimerprograma;

/**
 * @author valelp
 * Tutorial 1 Java: Mi Primer Programa
 */
public class MiPrimerPrograma {
    public static void main(String[] args) {
        System.out.println(";Mi primer programa!");
    }
}
```

Si le damos a ejecutar, obtendremos lo siguiente en la consola:



```
run:
Este es mi primer programa
BUILD SUCCESSFUL (total time: 0 seconds)
```