

```
.mail{  
  background: url(../img/mailico.png) no-repeat center;  
  display: inline-block;  
  width: 12px;  
  height: 14px;  
  float: left;  
  margin: 2px 7px 0 0;  
}  
em.phone{  
  background: url(../img/phoneico.png) no-repeat center;  
  display: inline-block;  
  width: 20px;  
  height: 18px;  
  float: left;  
  margin: 3px 8px 0 0;  
}
```

VARIABLES Y ESTRUCTURAS DE CONTROL EN LA PROGRAMACIÓN ORIENTADA A OBJETOS: JAVA

ELEMENTOS BÁSICOS DE UN LENGUAJE DE PROGRAMACIÓN ORIENTADO A OBJETOS



ELEMENTOS BÁSICOS DE UN LENGUAJE DE PROGRAMACIÓN ORIENTADO A OBJETOS

ESTRUCTURA DE CONTENIDOS

INTRODUCCIÓN.....	3
1. LENGUAJES ORIENTADOS A OBJETOS.....	3
1.1 Conceptos básicos.....	4
1.2 Características.....	5
1.3 Los más nombrados.....	6
2 ENTORNO DE TRABAJO.....	10
2.1 ¿Qué lo compone?.....	10
2.2 ¿Cómo se obtiene?.....	11
2.3 Ejemplo completo.....	12
GLOSARIO.....	18
REFERENCIAS BIBLIOGRÁFICAS.....	19
CRÉDITOS.....	20

ELEMENTOS BÁSICOS DE UN LENGUAJE DE PROGRAMACIÓN ORIENTADO A OBJETOS

INTRODUCCIÓN



La programación de sistemas a lo largo de los años ha venido evolucionando, de manera que en la gran mayoría de las empresas se requieren soluciones a la medida, las cuales contribuyan a la optimización del recurso humano y tecnológico, permitiendo la automatización de los procesos, de acuerdo al objeto de cada organización.

En este material de formación se conocerá, en términos generales, la información referente a los conceptos básicos y las características de los lenguajes de programación orientados a objetos; asimismo, se abordará el proceso de instalación del software necesario para usarlo como entorno de desarrollo durante todo el avance del programa.

1. LENGUAJES ORIENTADOS A OBJETOS

La programación se puede enfocar teniendo en cuenta un concepto denominado paradigma, el cual hace énfasis específicamente en la forma de realizar las cosas al momento de escribir las secuencias de código que conforman los programas.



Quizás el concepto es fácilmente entendible, identificando también lo que no es la programación orientada a objetos, la cual hace mucho tiempo dio sus primeros pasos en el mundo de la programación. Es posible recordar que la codificación se iniciaba en una primera línea y de allí hacia abajo se desplegaban todas las instrucciones que se deseaba fueran cumplidas por cada programa, y en caso de un fallo, o de requerir alguna modificación, era necesario repasar todo el código para identificar cómo corregir o cómo cambiar lo que se requería.

Dicho paradigma denominado programación secuencial o estructurada, ha sido reemplazada, o mejor, modernizada, a través del paradigma orientado a objetos, que como su nombre lo indica, centra la escritura de los programas en los objetos que lo componen, y de esta manera permite dividirlos en una especie de

ELEMENTOS BÁSICOS DE UN LENGUAJE DE PROGRAMACIÓN ORIENTADO A OBJETOS

subprogramas que se comunican entre sí para conformar un todo, haciendo mucho más fácil el proceso de corrección de fallas y de actualización de las funcionalidades.

1.1 Conceptos básicos

Para comprender mejor la programación orientada a objetos, es necesario identificar los conceptos básicos, tal y como se muestran en la siguiente figura:



Cuando se va a proceder con la programación de un sistema bajo el paradigma orientado a objetos, el primer paso es efectuar algo llamado abstracción, lo cual consiste básicamente en identificar los componentes del sistema, sus características y funcionalidades. Dichos componentes son fácilmente identificables a través del análisis de sustantivos; es decir, analizando aquellos elementos que harán parte del sistema y que pueden definirse como personas, animales o cosas (estos serán los que se denominen objetos).

Teniendo en cuenta lo anterior, es necesario precisar en lo siguiente:

- Objeto:** elemento que hace parte del sistema y del cual se pueden definir ciertos datos y funcionalidades; de allí se puede inferir que los objetos están compuestos por atributos y métodos, donde los atributos corresponden a dichos datos y los métodos a las funcionalidades mencionadas.
- Clase:** es un modelo que almacena la estructura básica de un objeto; es decir, se definen en esta los atributos y métodos que van a componer los objetos.
- Instancia:** consiste en cada uno de los objetos creados a partir de una clase.

EJEMPLO PRÁCTICO



Para comprender mejor la conceptualización que se acaba de plasmar en cuanto a la programación orientada a objetos, se ilustra el siguiente ejemplo:

Se requiere apoyar la gestión institucional de un centro veterinario generando una solución informática que permita administrar las historias clínicas, el registro de usuarios, de médicos, pacientes, formas de pago y servicios ofrecidos.

El primer paso consiste entonces en efectuar la abstracción e identificar los componentes que intervienen en la creación de dicho sistema; los objetos podrían ser: historias clínicas, usuarios, médicos, pacientes, formas de pago y servicios ofrecidos; esto teniendo en cuenta que en la descripción del requerimiento, son

ELEMENTOS BÁSICOS DE UN LENGUAJE DE PROGRAMACIÓN ORIENTADO A OBJETOS

esas palabras las que hacen alusión a sustantivos y de las cuales se pueden identificar **datos (atributos)** y **comportamientos (métodos)**.

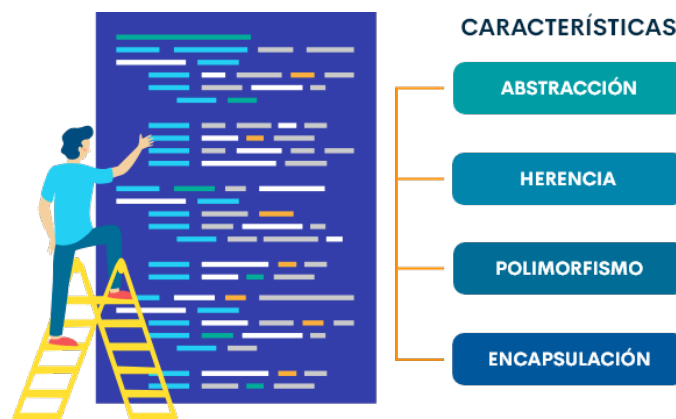
En la siguiente figura se observa la abstracción realizada para el ejemplo:

PERSONA		PACIENTE		PAGO	
ATRIBUTOS	cedula nombre apellido Telefono	ATRIBUTOS	nombre raza fechaNacimiento	ATRIBUTOS	fecha concepto valor
MÉTODOS	crearPersona() modificarPersona() EliminarPersona() ConsultarPersona()	MÉTODOS	crearPaciente() modificarPaciente() EliminarPaciente() ConsultarPaciente()	MÉTODOS	insertarPago() consultarPago()

HISTORIA		SERVICIO	
ATRIBUTOS	numero detalle	ATRIBUTOS	codigo descripcion fechaNacimiento
MÉTODOS	crearHistoria() modificarHistoria() consultarHistoria()	MÉTODOS	crearServicio() modificarServicio() EliminarServicio() ConsultarServicio()

1.2 Características

Continuando con la teorización respecto a la programación orientada a objetos, es importante tener claras las características en que se fundamenta, las cuales pueden observarse en siguiente figura:



ELEMENTOS BÁSICOS DE UN LENGUAJE DE PROGRAMACIÓN ORIENTADO A OBJETOS

1. **Abstracción:** como ya se vio en el apartado anterior, la abstracción consiste en el proceso de identificación de los componentes del sistema, que pasan a ser los objetos y a su vez sus características y funcionalidades, reconociendo estas como atributos y métodos.
2. **Herencia:** es el proceso mediante el cual unos objetos mantienen sus propios atributos y funcionalidades, pero también toman atributos o funcionalidades de otros; para comprender mejor, se continúa con el ejemplo del centro veterinario, en donde se identifica una clase que es personas, que sería la clase padre, pero se requiere otra clase denominada médicos, llamada la clase hijos, dado que los veterinarios van a tener atributos y métodos diferentes, por lo tanto heredan algunos atributos y métodos de la clase padre, pero tienen también los suyos propios.
3. **Polimorfismo:** se trata de la característica que permite asignar las mismas instrucciones a diferentes objetos, teniendo en cuenta que cada uno de ellos las va a ejecutar de acuerdo a sus particularidades internas. Para ejemplificarlo, en el sistema del centro veterinario, cuando el médico registra el alta de un paciente que se encuentra hospitalizado, suceden tres cosas: primero se informa a la secretaria para que genere la orden de pago, segundo, se informa en pantalla al veterinario que el alta fue registrada y tercero, se informa al propietario para que recoja al paciente; como puede observarse, en las tres cosas se efectúa la misma acción, la cual es informar internamente a cada uno; esos informes se realizan de manera diferente, pero en realidad se está dando la misma orden a varios objetos.

1.3 Los más nombrados

La lista de lenguajes orientados a objetos es larga; sin embargo, por su popularidad, preferencias personales, facilidad de utilización, abundancia de información de referencia, entre otros, naturalmente existen aquellos que son más conocidos, los cuales pueden observarse en la siguiente figura:



Según Aburruzaga, Medina y Palomo (2006):

- » **C++:** fue inventado por Bjarne Stroustrup mientras trabajaba para los laboratorios Bell de AT&T, en el año 1985 aproximadamente. Su nombre deriva del operador incremento del lenguaje C; se podría decir que C++ es un C incrementado, un paso más en el C.
- » **C#:** fue creado por Anders Hejlsberg, se trata de una mejora sustancial a los lenguajes que lo preceden C y C++, maneja una sintaxis similar a la de Java y cuenta con una librería de clases muy completa y bien diseñada.
- » **Python:** creado por Guido Van Rossum, consiste en un lenguaje de código abierto y muy fácil de utilizar, no es compilado, sino que trabaja por medio de un intérprete interactivo para mayor facilidad del programador; aunque suele ser mencionado en los estilos de programación orientada a objetos, acepta múltiples paradigmas.
- » **Java:** creado por James Gosling, guarda grandes parecidos con los lenguajes C y C++, difiere en el



ELEMENTOS BÁSICOS DE UN LENGUAJE DE PROGRAMACIÓN ORIENTADO A OBJETOS

hecho de que es completamente orientado a objetos; una de sus bondades es que permite incluir programas directamente en páginas web, lo cual brinda una interesante interactividad entre el usuario y el sistema.

En cuanto a la forma de ejecución del lenguaje, el código fuente se convierte en un código de mayor entendimiento para la máquina llamado bytecodes, los cuales se ejecutan a través del intérprete.

Gracias a esta conversión del código a bytecodes, las soluciones programadas bajo este lenguaje son utilizables en cualquier sistema operativo, debido a que esto consiste en un formato entendible por cualquier máquina, sin importar la arquitectura que maneje.

» Aplicación.

Es un programa grande e independiente que puede ejecutarse directamente en el equipo; las instrucciones de dicho programa son escritas dentro del método principal, adicionalmente, es posible acceder a los archivos del computador y que se ejecuten a través del (JRE) Java Runtime Environment.

Las aplicaciones hechas en Java pueden ser mostradas en pantalla de dos formas: en modo texto haciendo uso del símbolo del sistema, y en modo gráfico, por medio de la clase Frame.

A continuación, se ilustra un ejemplo de una aplicación que devuelve el contenido de una cadena de caracteres, ejecutada en modo texto, y escrita utilizando únicamente el bloc de notas.

```
public class Primera {  
    public static void main(String[] args) {  
        System.out.println("Me gusta JAVA y estoy listo para comenzar");  
    }  
}
```

Una vez digitado el código, se guarda el archivo con el mismo nombre de la clase y con extensión **.java** en una ruta conocida (la ruta para el ejemplo es **D:\CursoJava**).

Posteriormente, se debe ejecutar dicho archivo con el fin de generar los bytecodes; para ello se abre una ventana del símbolo del sistema, presionando las teclas Win+R, se digita el texto **cmd** y se presiona el botón Aceptar.

Se accede a la ruta donde se grabó el archivo **.java** y se ejecuta a través del comando **javac**; las órdenes serían las siguientes:

- » **d:** para cambiar al disco D, que es donde está la carpeta requerida.
- » **cd CursoJava** para entrar a la carpeta.
- » **javac Primera.java** para generar los bytecodes.

ELEMENTOS BÁSICOS DE UN LENGUAJE DE PROGRAMACIÓN ORIENTADO A OBJETOS

En el símbolo del sistema se ve del siguiente modo:

```
Microsoft windows [versión 6.3.9600]
<c> 2013 Microsoft Corporation. Todos los derechos reservados.

C:\Users\user>d:

D:\>cd CursoJava

D:\CursoJava>javac Primera.java

D:\CursoJav>
```

Abriendo la ruta que contiene el archivo **Primera.java** se puede observar que luego de ejecutarlo en el símbolo del sistema con el comando **javac**, apareció un nuevo archivo con extensión **.class**, dicho archivo es el que contiene los bytecodes.



Ahora para obtener el resultado final de la aplicación creada, en el mismo símbolo del sistema, se debe ejecutar el archivo sin extensión, con el comando **java** e inmediatamente se observa en la siguiente línea, el resultado devuelto.

```
Microsoft windows [versión 6.3.9600]
<c> 2013 Microsoft Corporation. Todos los derechos reservados.

C:\Users\user>d:

D:\>cd CursoJava

D:\CursoJava>javac Primera.java

D:\CursoJav>java Primera
Me gusta JAVA y estoy listo para comenzar

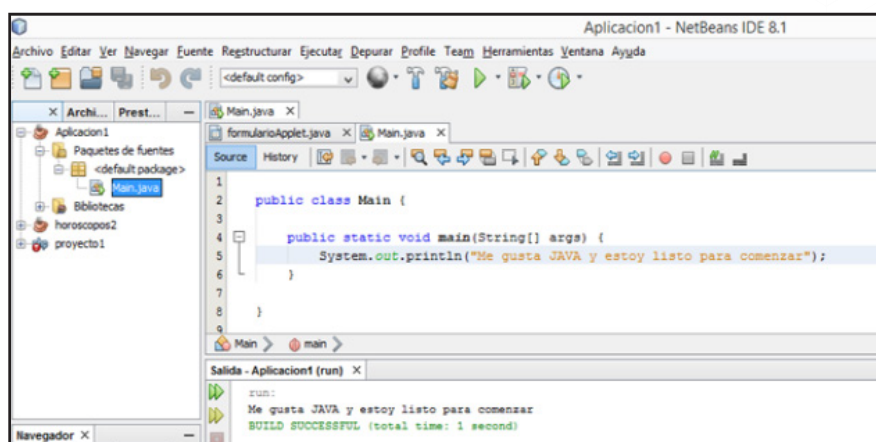
D:\CursoJava>
```


ELEMENTOS BÁSICOS DE UN LENGUAJE DE PROGRAMACIÓN ORIENTADO A OBJETOS

Escribir cada línea de código sin ningún tipo de asistente, ayuda al programador a familiarizarse cada vez más con la sintaxis del lenguaje; asimismo, es importante tener en cuenta que existiendo las herramientas que facilitan el proceso, se reduce la posibilidad de errores y se optimiza el tiempo del programador, por lo que vale la pena utilizarlas.

Emplear un entorno de desarrollo integrado como por ejemplo NetBeans, hace que el programador no necesite conocer tan al pie de la letra las diferentes secuencias de código que debe ir escribiendo para cada programa, dado que esta herramienta, entre muchas otras facilidades, realiza sugerencias y marca las posibles fallas dentro del código escrito.

Para entender mejor lo explicado anteriormente, a continuación, se observa en la siguiente figura, el ejemplo visto con el archivo **Primera.java** resuelto dentro del NetBeans:



Basta con crear un nuevo proyecto, incluir una clase principal y allí escribir el código que imprima el texto; posteriormente se presiona el botón ejecutar y en la parte inferior sale el resultado, es decir que se realizó lo mismo que en el ejemplo con el símbolo del sistema y los comandos **javac** y **java**, pero tan solo con presionar un botón, de allí que la recomendación sea utilizar el entorno de desarrollo integrado de NetBeans para la ejecución de este proceso de formación.

» **Applet.** Es un programa pequeño que se ejecuta a través de un navegador que sea compatible con Java, se crea utilizando la clase applet; no cuenta con acceso a los archivos del computador y tiene ciertas restricciones, de hecho, ya no es tan utilizado, dado que los navegadores, por seguridad, en múltiples ocasiones, restringen la ejecución de este código.

A continuación, se realiza el ejemplo de la clase **Primera.java** en forma de applet a través de NetBeans para ver su ejecución en modo gráfico. Los pasos son los siguientes:

1. Se crea un nuevo proyecto.
2. Se elimina la clase principal creada dado que al programar applets, esta no se necesita.
3. Se crea un formulario JApplet.
4. Se arrastra una etiqueta para escribir el texto.
5. Se ejecuta el applet.

El código generado y el resultado devuelto puede observarse en la siguiente figura:

A screenshot of a Java Applet window. The title bar reads "Visor de Applet: Applet1/formul...". The main content area displays the text "Me gusta JAVA y estoy listo para comenzar" in a bold, black, sans-serif font. At the bottom of the window, a status bar contains the text "Applet iniciado."

Observando entonces el ejemplo anterior, se puede concluir que los procesos de programación de applets y de aplicaciones son mucho más eficientes a través de la utilización de un entorno de desarrollo integrado como el Netbeans, pues en el ejemplo visto, a pesar de tratarse únicamente de la visualización en pantalla de una cadena de texto, se generaron más de 50 líneas de código cuando la única acción que se llevó a cabo fue arrastrar una etiqueta y asignarle un texto. Es posible imaginarse en el caso de la programación de aplicaciones con requerimientos reales, la cantidad de código que tendría que escribirse, y si es en el bloc de notas, no se trataría de ninguna manera de un proceso óptimo y eficiente.

2 ENTORNO DE TRABAJO

En este apartado del material de formación, se menciona el software requerido para la programación de aplicaciones y applets con el lenguaje orientado a objetos Java; de igual manera, se indica la forma de obtener dicho programa, instalarlo y configurarlo.

2.1 ¿Qué lo compone?

El entorno de trabajo para la escritura y ejecución de programas desarrollados con el lenguaje Java está compuesto básicamente por dos softwares que son: el IDE NetBeans y el JDK (Java Development Kit, traducido al español como Kit de Desarrollo de Java).

ELEMENTOS BÁSICOS DE UN LENGUAJE DE PROGRAMACIÓN ORIENTADO A OBJETOS

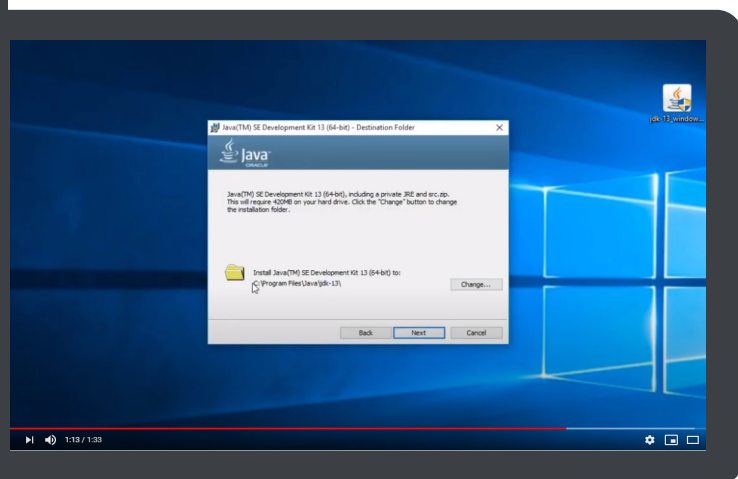


Es importante tener en cuenta que el IDE NetBeans se sugiere como entorno de desarrollo para la escritura de los programas; sin embargo, queda a gusto del aprendiz, el hecho de utilizar un entorno diferente, dado que existen otros, como por ejemplo Eclipse, el cual brinda la misma funcionalidad.

2.2 ¿Cómo se obtiene?

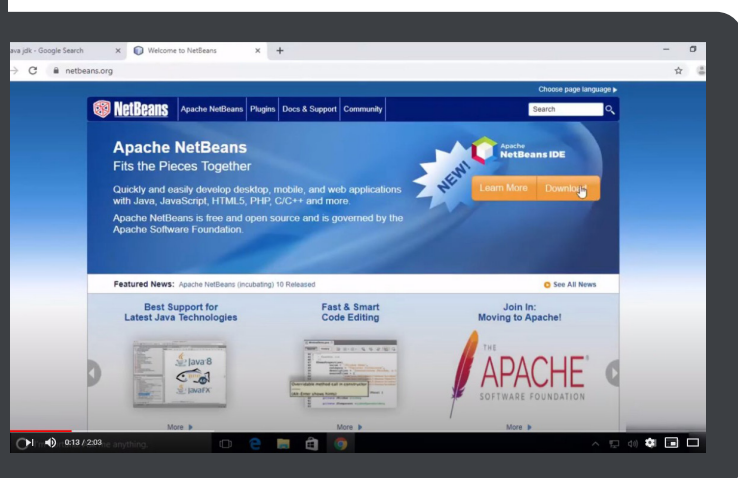
Una de las grandes ventajas en el desarrollo de soluciones con el lenguaje Java es que ambos softwares son gratuitos y de fácil consecución a través de internet; a continuación, se explica con detalle el proceso para descargar ambos programas.

» Instalación JDK (Java Development Kit)



CLIC PARA
VER VIDEO

» Instalación de NetBeans



CLIC PARA
VER VIDEO

ELEMENTOS BÁSICOS DE UN LENGUAJE DE PROGRAMACIÓN ORIENTADO A OBJETOS

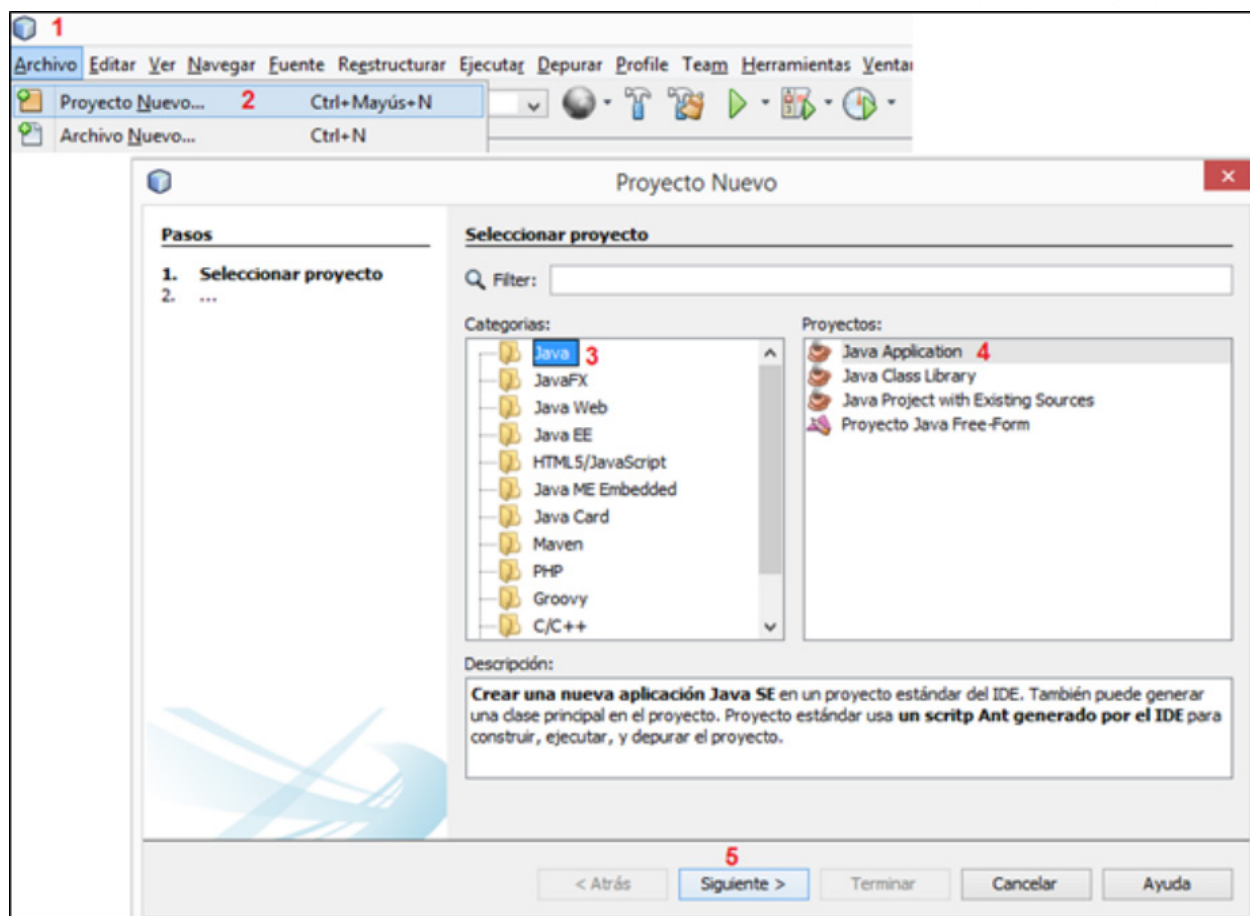
2.3 Ejemplo completo

Teniendo ya listo el entorno de trabajo para el desarrollo de aplicaciones con el lenguaje Java, se realiza a continuación un ejemplo completo que ilustra la forma cómo se integran estos dos softwares.

El ejemplo consiste en programar un applet en modo gráfico que esté compuesto por tres botones y una etiqueta de texto; de acuerdo con el botón que se presione, el color del texto de la etiqueta cambiará.

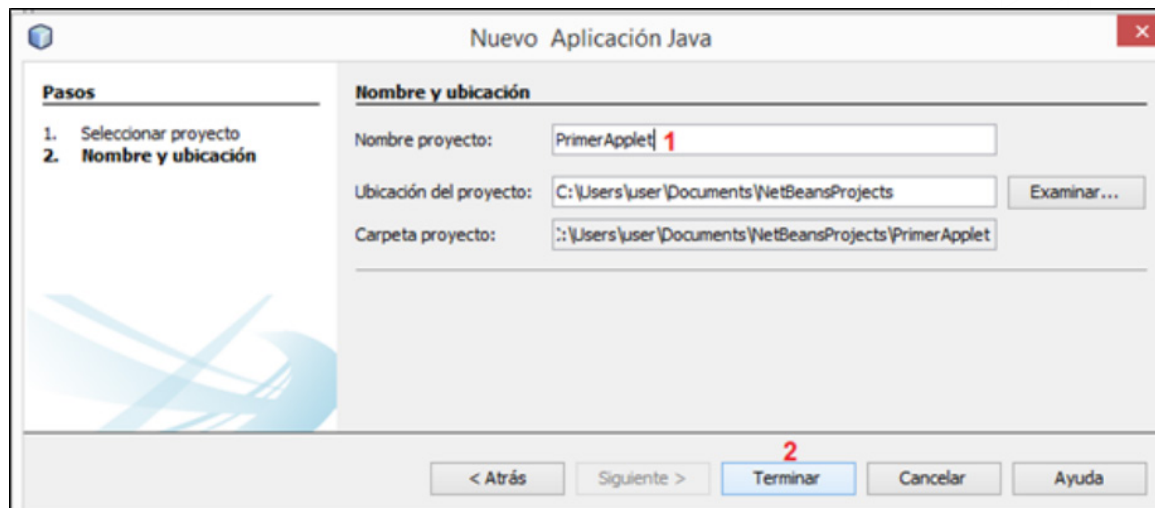
Se observa a continuación el paso a paso para la creación de dicho applet.

Paso 1: crear un nuevo proyecto en NetBeans.

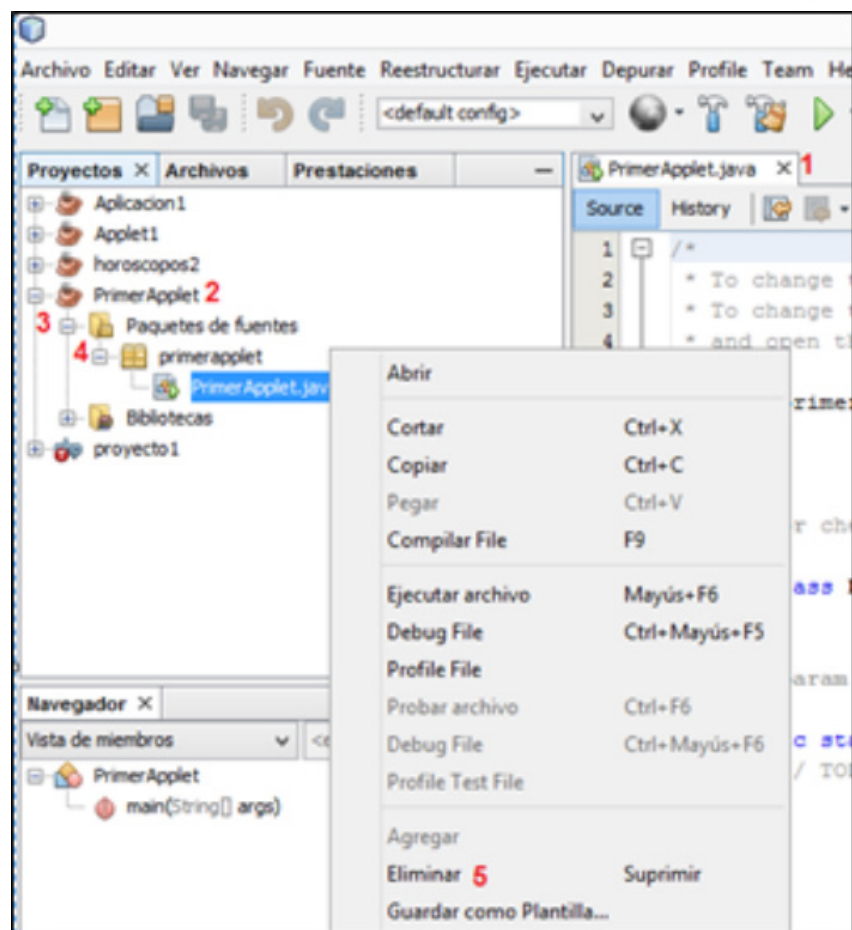


ELEMENTOS BÁSICOS DE UN LENGUAJE DE PROGRAMACIÓN ORIENTADO A OBJETOS

Paso 2: asignar un nombre al proyecto y dar clic en Terminar

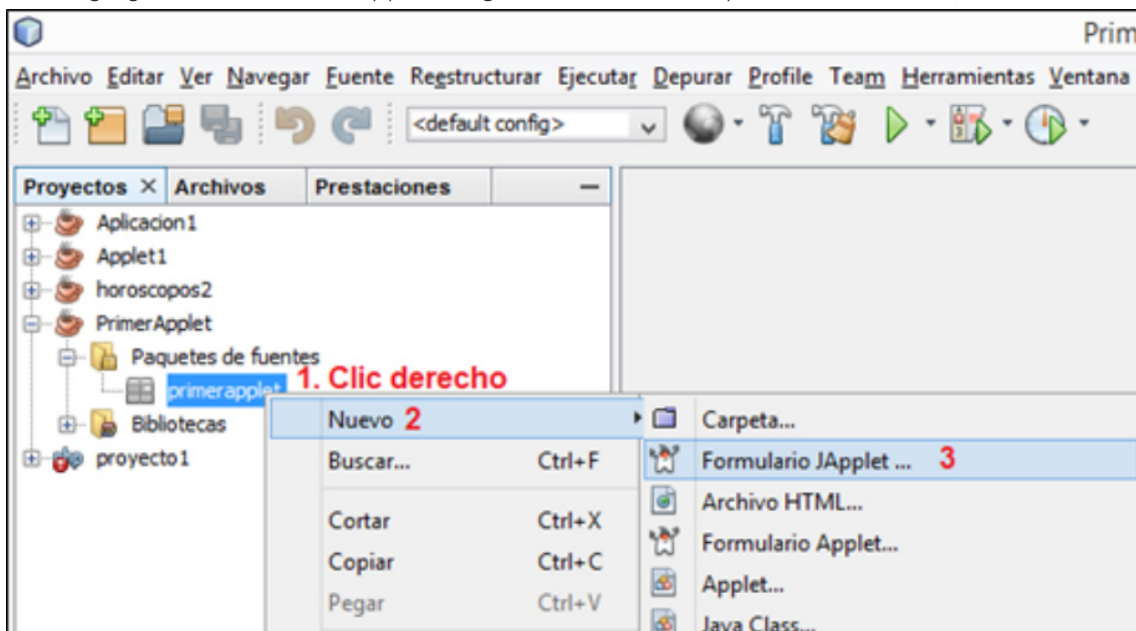


Paso 3: cerrar la clase principal creada y posteriormente eliminarla, ya que en la programación de un applet no es requerida.

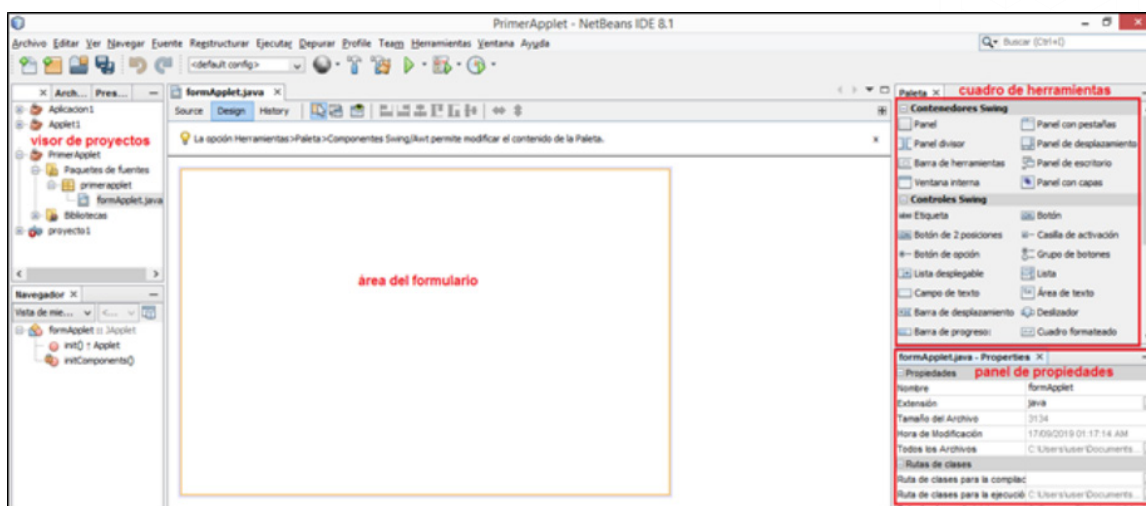


ELEMENTOS BÁSICOS DE UN LENGUAJE DE PROGRAMACIÓN ORIENTADO A OBJETOS

Paso 4: agregar un formulario JApplet, luego darle un nombre y dar clic en Terminar.



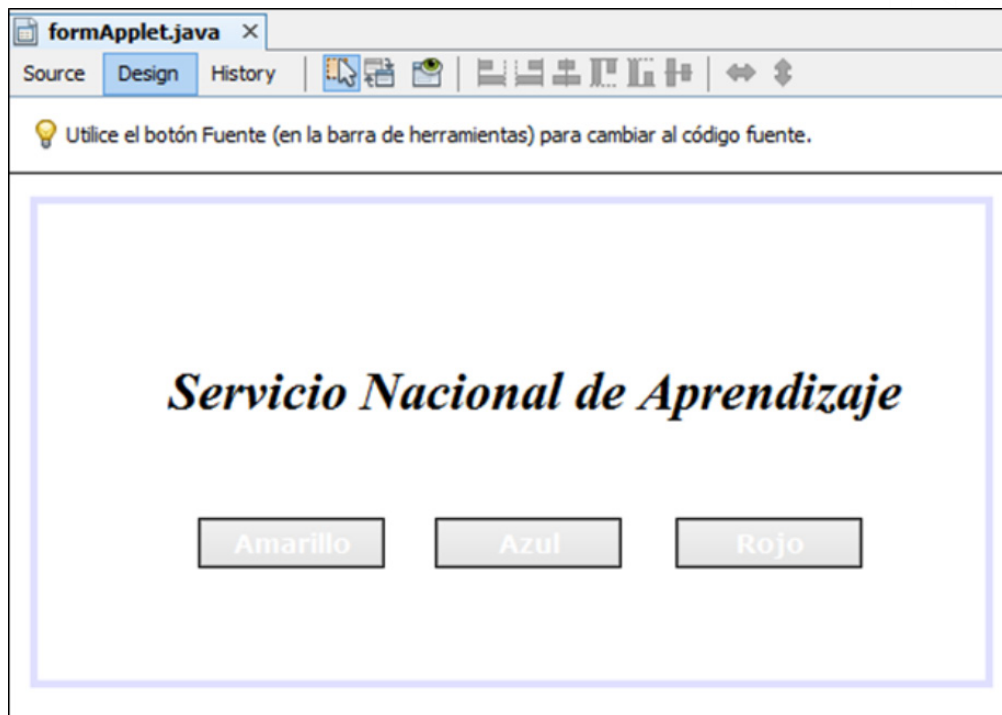
Paso 5: se observa en el entorno de NetBeans, el área del formulario, el panel de propiedades y el cuadro de herramientas con los elementos que se pueden arrastrar para crear la interfaz gráfica del applet.



Paso 6: se inicia la creación de la interfaz gráfica del formulario, arrastrando del cuadro de herramientas la etiqueta y los tres botones requeridos; se ubican en el área del formulario y se da clic derecho a cada uno de estos para cambiarle el nombre (en el ejercicio serán llamados etiqueta, btn1, btn2 y btn3). A través del panel de propiedades se les realiza las configuraciones deseadas (textos, colores, tipo de fuente, bordes, entre otros).

El diseño del formulario debe quedar similar al que se muestra en la siguiente figura:

ELEMENTOS BÁSICOS DE UN LENGUAJE DE PROGRAMACIÓN ORIENTADO A OBJETOS



Paso 7: ahora se hace doble clic en cada uno de los botones para programar la acción que estos deben efectuar cuando sean presionados por el usuario que ejecuta el applet; el código es el siguiente:

```
etiqueta.setForeground(Color.red);
```

Donde **etiqueta** es el nombre que se le dio a la etiqueta que tiene el texto Servicio Nacional de Aprendizaje; **setForeground** se utiliza para modificar la propiedad del color del texto y **Color.red** modifica dicha propiedad al color rojo; naturalmente para los otros dos botones deben usarse los colores yellow y blue, respectivamente. El código de los botones queda como se observa en la siguiente figura.

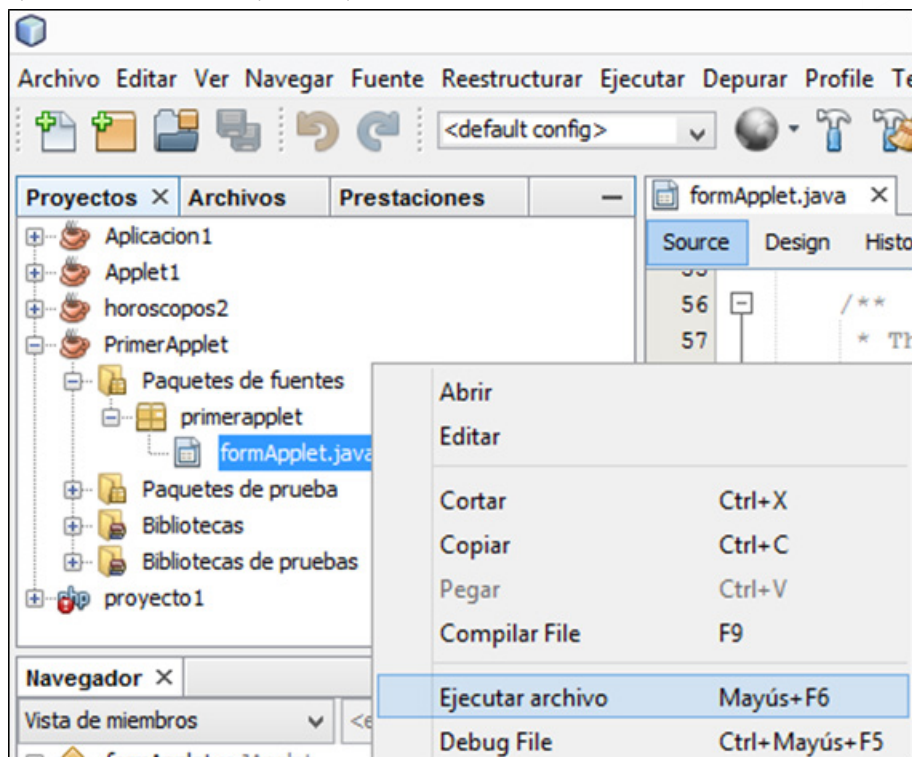
```
private void btn1ActionPerformed(java.awt.event.ActionEvent evt) {
    etiqueta.setForeground(Color.yellow);
}

private void btn2ActionPerformed(java.awt.event.ActionEvent evt) {
    etiqueta.setForeground(Color.blue);
}

private void btn3ActionPerformed(java.awt.event.ActionEvent evt) {
    etiqueta.setForeground(Color.red);
}
```

ELEMENTOS BÁSICOS DE UN LENGUAJE DE PROGRAMACIÓN ORIENTADO A OBJETOS

Paso 8: una vez terminada la programación, se ejecuta el applet dando clic derecho al formulario en el visor de proyectos, y seleccionando la opción Ejecutar archivo.



Paso 9: el applet se carga, inicia y muestra la interfaz del formulario



ELEMENTOS BÁSICOS DE UN LENGUAJE DE PROGRAMACIÓN ORIENTADO A OBJETOS

Paso 10: probar el applet presionando cada botón y observando el resultado generado.



Una vez realizado por completo el ejemplo anterior, se puede verificar que el software ha quedado funcionando de manera satisfactoria y permite la creación de applets a través del lenguaje de programación Java.

Se ha visto el desarrollo de un applet sencillo, pero de este modo se puede programar cualquier tipo de funcionalidad, haciendo uso de las diferentes instrucciones soportadas por el lenguaje, como manejo de declaración de variables y de operaciones.

ELEMENTOS BÁSICOS DE UN LENGUAJE DE PROGRAMACIÓN ORIENTADO A OBJETOS

GLOSARIO

Atributo: Es cada uno de los datos definidos para un objeto.

IDE: Integrated Development Environment, traducido al español como Entorno de Desarrollo Integrado; es un software que a través de una interfaz gráfica, facilita al programador la escritura de código en diferentes lenguajes de programación.

JDK: Java Development Kit, traducido al español como Kit de Desarrollo de Java; es un software requerido para correr programas que son escritos en lenguaje Java.

JRE: Java Runtime Environment, consiste en un conjunto de herramientas de software que posibilita la ejecución de programas escritos en Java.

Método: Es cada una de las funcionalidades definidas para un objeto.

Paradigma: Consiste en la forma en que se escriben los códigos de un programa.

Soluciones a la medida: Programas creados a partir de una lista de requerimientos definida con anterioridad, de acuerdo con las necesidades del cliente.



REFERENCIAS BIBLIOGRÁFICAS

Aburruzaga, G., Medina, I. y Palomo, F. (2006). Fundamentos de C++. Málaga: Agapea.

Java. (2019). Descarga gratuita de Java. <https://www.java.com/es/download/>

NetBeans. (2019). NetBeans IDE 8.2 Download. <https://netbeans.org/downloads/8.2/>

Oracle. (2019). Oracle cloud free tier. <https://www.oracle.com/index.html>



ELEMENTOS BÁSICOS DE UN LENGUAJE DE PROGRAMACIÓN ORIENTADO A OBJETOS

CRÉDITOS

Equipo Contenido Instruccional		
» Gloria Matilde Lee Mejía	Responsable equipo	Centro de Comercio y Servicios - Regional Tolima
» Rafael Nelftali Lizcano Reyes	Asesor pedagógico	Centro Industrial Del Diseño y La Manufactura - Regional Santander
» Lucero Montes Arenas	Gestora de Desarrollo de Programas	Centro para la Formación Cafetera (Caldas)
» Julio Alexander Rodriguez Del Castillo	E-pedagogo instruccional	Centro Atención Sector Agropecuario Regional Risaralda
» Rachman Bustillo Martínez	Evaluador de contenido	Centro Atención Sector Agropecuario Regional Risaralda
» Natalia Andrea Bueno Pizarro	Diseñadora instruccional	Centro de Diseño y Metrología - Regional Distrito Capital
Equipo Diseño y Desarrollo		
» Francisco José Lizcano Reyes	Responsable Equipo	Centro Industrial Del Diseño y La Manufactura - Regional Santander
» Daniel Ricardo Mutis Gómez	Diagramación web	Centro Industrial Del Diseño y La Manufactura - Regional Santander
» Yazmin Rocio Figueroa Pacheco	Construcción documentos digitales	Centro Industrial Del Diseño y La Manufactura - Regional Santander
» Edgar Mauricio Cortes García	Desarrollo front-end	Centro Industrial Del Diseño y La Manufactura - Regional Santander
» Luis Gabriel Urueta Álvarez	Desarrollo de actividades didácticas	Centro Industrial Del Diseño y La Manufactura - Regional Santander
» Leyson Fabian Castaño Pérez	Integración de recursos y pruebas	Centro Industrial Del Diseño y La Manufactura - Regional Santander





ELEMENTOS BÁSICOS DE UN LENGUAJE DE PROGRAMACIÓN ORIENTADO A OBJETOS

Equipo de Gestores de Repositorio

» Kely Alejandra Quiros Duarte

Administrador repositorio de contenidos y gestores de repositorio.

Centro de comercio y servicios - Regional Tolima

Recursos gráficos

Fotografías y vectores tomados de www.shutterstock.com y www.freepik.com

creative
commons



BY NC SA

Este material puede ser distribuido, copiado y exhibido por terceros si se muestra en los créditos. No se puede obtener ningún beneficio comercial y las obras derivadas tienen que estar bajo los mismos términos de la licencia que el trabajo original.

