

El curso proporciona la base que necesita cualquier programador de Java, tanto en cuanto a las peculiaridades de la tecnología que engloba Java SE, como del lenguaje de programación en sí. También se estudian las características de dos tipos de aplicaciones que podemos crear con Java SE: applets y aplicaciones autónomas. Al finalizar el curso, el alumno estará preparado para plantearse otros retos profesionales, como el estudio de las tecnologías empresariales de Java o Java EE.

Nº Lecciones	H.Lectivas (teoría)	H.Lectivas (prácticas)	H.Lectivas (total)
20	20	20	40

Este curso dispone de un manual adicional en color (sólo para España).

Este curso sólo está disponible para LMSs compatibles con SCORM o AICC.

## TEMARIO

### Módulo 1: El lenguaje Java

*Introduce la tecnología Java describiendo cuáles son sus características fundamentales. Seguidamente se proporciona una visión bastante completa del lenguaje Java, estudiando desde los conceptos fundamentales de programación, pasando por sus características orientadas a objetos y otros conceptos avanzados, como las excepciones, clases abstractas e interfaces.*

#### Lección 1: Introducción

*Describe qué tecnologías están detrás del nombre Java, describiendo la base de la plataforma tecnológica, es decir, de Java SE. Se describe también cómo descargar e instalar el JDK de Java junto al entorno de desarrollo integrado o IDE NetBeans. Finalmente, se ofrece una visión general del curso.*

Capítulo 1: ¿Qué es Java?

Capítulo 2: ¿Qué es Java SE?

Capítulo 3: Instalar el JDK y NetBeans

Capítulo 4: Presentación del curso

#### Lección 2: NetBeans 5.5

*Presenta el entorno de desarrollo o IDE NetBeans 5.5, describiendo cómo realizar las tareas habituales en este entorno a la hora de desarrollar una aplicación Java. Estudia las distintas partes de la ventana de NetBeans, las herramientas o paneles que habitualmente se utilizan, cómo utilizar sus opciones para configurarlo a nuestra forma de trabajar, etc. La lección finaliza estudiando el sistema de ayuda de NetBeans y la posibilidad de actualizarlo desde Internet.*

Capítulo 1: Iniciar NetBeans

Capítulo 2: El primer proyecto

Capítulo 3: Personalizando el editor

Capítulo 4: Ayuda y actualizaciones

#### Lección 3: Pensar en objetos (I)

*Introduce los conceptos fundamentales de la programación orientada a objetos, ya que Java es un lenguaje que sigue este paradigma de programación.*

Capítulo 1: Orientado a objetos

Capítulo 2: Clases

Capítulo 3: Propiedades

Capítulo 4: Métodos y constructores

#### Lección 4: Pensar en objetos (II)

*Estudia algunos conceptos de la programación orientada a objetos, como la forma de crear objetos y ocultar la*

*estructura interna de las clases. Explica algunas técnicas de refactoring disponibles en NetBeans, que permiten crear código claro, legible y, sobre todo, más fácil de actualizar posteriormente.*

- Capítulo 1: Repaso
- Capítulo 2: Crear objetos
- Capítulo 3: Compilar el código
- Capítulo 4: Ocultar la información
- Capítulo 5: Encapsular propiedades

Lección 5: Herencia

*Explica el concepto de herencia en un lenguaje de programación orientado a objetos y cómo utilizarla en Java. Se destacan los beneficios de utilizar esta característica en un proyecto de programación de cierta envergadura y las herramientas que proporciona NetBeans para ello.*

- Capítulo 1: Presentación
- Capítulo 2: Crear subclases
- Capítulo 3: Crear objetos de las subclases
- Capítulo 4: Sobrescribir métodos
- Capítulo 5: El acceso protected
- Capítulo 6: NetBeans y la herencia

Lección 6: Variables y constantes

*Estudia conceptos básicos de programación, como variable, constante, tipo de datos y cómo utilizarlos en Java.*

- Capítulo 1: Variables en Java
- Capítulo 2: Tipos de datos
- Capítulo 3: Inicializar las variables
- Capítulo 4: Constantes

Lección 7: Expresiones

*Estudia conceptos básicos de programación, como operadores, desbordamiento, conversión implícita y explícita, construir expresiones y cómo hacerlo en Java.*

- Capítulo 1: Operadores para enteros
- Capítulo 2: Desbordamiento (overflow y underflow)
- Capítulo 3: Operadores para reales
- Capítulo 4: Operador de concatenación

Lección 8: Arrays y estructuras de control

*Introduce el concepto de array como estructura de datos y las estructuras de control, que permiten modificar el flujo de ejecución de un programa.*

- Capítulo 1: ¿Qué es un array?
- Capítulo 2: Estructuras de repetición
- Capítulo 3: Estructuras de decisión
- Capítulo 4: Combinar estructuras

Lección 9: Más sobre los arrays

*Describe características avanzadas en el uso de arrays, como la posibilidad de crear arrays cuyos elementos son objetos de una determinada clase; arrays con más de una dimensión; y métodos con un número variable de parámetros.*

- Capítulo 1: Arrays de objetos
- Capítulo 2: Arrays multidimensionales
- Capítulo 3: Métodos con un número variable de parámetros

Lección 10: Cadenas de caracteres o strings

*Describe la clase String del lenguaje Java, que permite trabajar cómodamente con cadenas de caracteres. Para ello, se estudian situaciones típicas en las que intervienen objetos Strings, como las búsquedas o comparación de cadenas. La lección finaliza presentando una clase alternativa a String: StringBuilder, que puede ser más eficaz en ciertas situaciones.*

- Capítulo 1: Cadenas de caracteres en Java
- Capítulo 2: Descripción de la clase String
- Capítulo 3: Comparar strings
- Capítulo 4: Realizar búsquedas
- Capítulo 5: La clase StringBuilder

Lección 11: Excepciones (I)

*Explica el concepto de excepción como método que proporciona Java para detectar y resolver errores durante la ejecución de una aplicación, así como situaciones excepcionales pero que no representan errores. Se estudia la diferencia entre ambos tipos de excepciones: checked y unchecked y cómo manejarlas mediante la estructura try - catch.*

- Capítulo 1: Introducción
- Capítulo 2: Tipos de excepciones
- Capítulo 3: La estructura try - catch
- Capítulo 4: Jerarquía de excepciones

Lección 12: Excepciones (II)

*Estudia el bloque finally para ejecutar código siempre que ocurre una excepción o cuando realmente la ejecución ha transcurrido normalmente. Después estudia la palabra clave throws para indicar que cierto código o método puede lanzar excepciones y, finalmente, estudia cómo crear clases de excepciones propias para tratar situaciones no cubiertas por las clases de Java SE.*

- Capítulo 1: El bloque finally
- Capítulo 2: ¿Dónde manejar las excepciones?
- Capítulo 3: Lanzar excepciones propias

Lección 13: Clases abstractas e interfaces

*Estudia los conceptos de clases abstractas e interfaces, que son habituales encontrar en las clases de la plataforma Java SE. Describe con ejemplos los propósitos con los que pueden ser utilizados.*

- Capítulo 1: Clases abstractas
- Capítulo 2: Clases abstractas de Java
- Capítulo 3: Crear interfaces
- Capítulo 4: Implementar interfaces

Lección 14: Paquetes (packages)

*Estudia el concepto de paquete, como una agrupación de clases afines y otros tipos de datos, que el programador crea para organizar el código. Explica cómo crear y utilizar los paquetes en Java.*

- Capítulo 1: Introducción
- Capítulo 2: Crear un paquete

- Capítulo 3: Utilizar los paquetes
- Capítulo 4: Los paquetes java.\*

Módulo 2: **Aplicaciones y applets**

*Se estudia la creación de applets y aplicaciones autónomas utilizando el lenguaje Java. Se explican temas como el uso de las fuentes de texto, incorporación de imágenes y animación mediante clases AWT o la creación de interfaz de usuario mediante clases Swing.*

Lección 15: Introducción a los applets

*Explica qué es un applet y anticipa la presencia de la clase Applet de Java, que proporciona la funcionalidad básica para crear este tipo de aplicaciones que se ejecutan en las páginas web. También indica cómo incrustar el applet en el código de la página web y las etapas que ocurren durante la ejecución de todo applet.*

- Capítulo 1: ¿Qué es un applet?
- Capítulo 2: La clase Applet
- Capítulo 3: La etiqueta <APPLET>
- Capítulo 4: Métodos fundamentales de los applets

Lección 16: Dibujar y añadir color

*Se estudian algunas clases de Java que ayudan a la hora de dibujar y mejorar el aspecto de un applet. Asimismo, se explica cómo crear color utilizando los colores seguros y el modelo RGB.*

- Capítulo 1: Rectángulos y polígonos
- Capítulo 2: Aplicar colores
- Capítulo 3: Elipses y arcos

Lección 17: Fuentes, imágenes y sonidos

*Se estudian los métodos que proporciona la clase Graphics de Java para dibujar texto, imágenes y utilizar sonido.*

- Capítulo 1: Fuentes de texto
- Capítulo 2: Información sobre las fuentes
- Capítulo 3: Utilizar imágenes
- Capítulo 4: Utilizar sonidos

Lección 18: Crear animación

*Estudia la creación de animación en un applet de Java. Introduce el concepto de hilo de ejecución y cómo manejar excepciones como método para controlar la ocurrencia de errores durante la ejecución.*

- Capítulo 1: Introducción
- Capítulo 2: Preparar la animación
- Capítulo 3: Primer intento
- Capítulo 4: Crear hilos de ejecución
- Capítulo 5: Reducir el parpadeo

Lección 19: Clases Swing (I)

*Presenta las clases Swing como clases más modernas que las AWT y optimizadas. Describe los componentes básicos de este tipo de clases de interfaz de usuario, así como algunos aspectos interesantes, como puede ser la facilidad para cambiar el aspecto y funcionalidad de una aplicación Swing.*

- Capítulo 1: AWT y Swing
- Capítulo 2: La clase JFrame
- Capítulo 3: Componentes básicos

Capítulo 4: Administradores de diseño (layouts)

Capítulo 5: Cambiar el aspecto y funcionalidad

Lección 20: Clases Swing (II)

*En la primera parte de la lección se describe el modelo de eventos de Java, utilizando clases Swing. En la segunda parte se hace referencia a las características de programación visual de NetBeans, aspecto que permite crear interfaz de usuario de una forma mucho más rápida y cómoda.*

Capítulo 1: Introducción

Capítulo 2: Escuchar los eventos

Capítulo 3: Manejar los eventos

Capítulo 4: Programación visual con NetBeans

Capítulo 5: Programar los eventos