



Curso	JAVA
-------	------

Objetivos

Nº	Objetivo de aprendizaje	Contenidos Generales	Actividades generales	Horas
				P

Objetivos

Los fundamentos de programación son la base para empezar a programar, antes incluso de elegir un lenguaje. Este etapa facilita el aprendizaje a todos aquellos que quieren aprender a programar y no saben por dónde empezar, o a aquellos programadores autoformados que carecen de fundamentos de algoritmia.

Destinatarios

Cualquier persona que quiera aprender a programar y tenga un nivel de matemáticas igual o superior al de bachillerato. Especialmente dirigido a personas que no han programado nunca, pero también útil para programadores con carencias en algoritmia.

1A	Introducción a al mundo de resolución de problemas por medio de un computador		✓ Teórico	3
2A	Conocer el problema a resolver	<ul style="list-style-type: none"> ✓ Sobre el objetivo ✓ Sobre los condicionantes ✓ Sobre el método o esquema de resolución ✓ Problema con resolución directa ✓ Problema con resolución documentada ✓ Problema con iteración para búsqueda de soluciones ✓ Problema a plantear con una tabla de decisión ✓ Problema con resolución intuitiva pero método "paso a paso" a determinar ✓ Sobre los resultados a obtener ✓ La anécdota en torno a conocer el problema 	<ul style="list-style-type: none"> ✓ Teórico ✓ Practico 	9
3A	Algoritmos, Pseudocódigo y diagramas de flujo	<ul style="list-style-type: none"> ✓ Introducción los conceptos de algoritmos, pseudocódigo y diagramas de flujos 	<ul style="list-style-type: none"> ✓ Teórico ✓ Practico 	3
4A	Concepto de Variable	<ul style="list-style-type: none"> ✓ Nombres de Variables ✓ Contenido y asignación de contenido a variables ✓ Variables booleanas ✓ Variables con índice o localizador. Arreglos ✓ Las variables como base de cálculo 	<ul style="list-style-type: none"> ✓ Teórico ✓ Practico 	9
4A	Estructura de Pseudocódigo	<ul style="list-style-type: none"> ✓ Introducción 	<ul style="list-style-type: none"> ✓ Teórico ✓ Practico 	6



		<ul style="list-style-type: none"> ✓ Inserción de comentarios en el pseudocódigo ✓ Herramientas lógicas y matemáticas. Escritura de fórmulas y operadores matemáticos. Prelación ✓ Prioridad entre operadores ✓ Funciones matemáticas ✓ Concatenación de órdenes y ordenación simbólica del pseudocódigo ✓ Instrucciones básicas 		
5A	Representación gráfico-esquemática	<ul style="list-style-type: none"> ✓ Símbolos para diagramas de flujo ✓ Normas para la creación de diagramas de flujo 	<ul style="list-style-type: none"> ✓ Teórico ✓ Practico 	3
6A	Estructuras de decisión	<ul style="list-style-type: none"> ✓ Instrucción Si ... Entonces y Si ... Entonces - SiNo ✓ Paradoja del Si burlado por un intervalo ✓ Instrucción Según (Caso) Hacer ✓ Transformación de tablas de decisión en pseudocódigo y diagramas de flujo 	<ul style="list-style-type: none"> ✓ Teórico ✓ Practico 	6
7A	Estructuras de repetición (bucles)	<ul style="list-style-type: none"> ✓ Instrucción Desde ... Siguiente ✓ Modificación del valor incremental en una instrucción Desde mediante Paso ✓ Valor del contador al acabar un bucle Desde ... Siguiente ✓ Uso y mal uso de la instrucción Desde ... Siguiente ✓ Instrucción Mientras ... Hacer ✓ Instrucción Hacer ... Repetir Mientras 	<ul style="list-style-type: none"> ✓ Teórico ✓ Practico 	6
8A		<ul style="list-style-type: none"> ✓ Contadores ✓ Acumuladores ✓ Interruptores ✓ Centinelas ✓ Centinelas para control de bucles ✓ Centinelas para toma de decisiones ✓ Entradas controladas por el usuario ✓ Final de archivo 	<ul style="list-style-type: none"> ✓ Teórico ✓ Practico 	9
9A	Modificación directa del flujo de programas	<ul style="list-style-type: none"> ✓ Introducción ✓ Instrucción Finalizar ✓ Instrucción SalirDesde ✓ Instrucción SalirMientras ✓ Instrucción SalirHacer e instrucción Hacer ... Repetir ✓ Instrucción IrA ✓ Reflexiones finales en torno al control directo del flujo de programas 	<ul style="list-style-type: none"> ✓ Teórico ✓ Practico 	6
1B	Que los estudiantes sean capaces de comprender y aplicar Clases y Objetos	<ul style="list-style-type: none"> ✓ Clases y objetos ✓ Variables y métodos de instancia y de clase 	<ul style="list-style-type: none"> ✓ Teórico ✓ Practico 	9



		<ul style="list-style-type: none"> ✓ Encapsulamiento ✓ Sobrecarga de métodos ✓ Constructores 		
2B	Que los estudiantes sean capaces de comprender y aplicar Herencia y Polimorfismo	<ul style="list-style-type: none"> ✓ Herencia ✓ Sobreescritura (overriding) de métodos ✓ Polimorfismo ✓ Clases abstractas ✓ Interfaces ✓ Clases internas ✓ Clases anónimas 	<ul style="list-style-type: none"> ✓ Teórico ✓ Practico 	9
3B	Que los estudiantes sean capaces de identificar y utilizar herramientas de paquetes	<ul style="list-style-type: none"> ✓ Estructuración de un sistema en paquetes ✓ Clases públicas y no públicas ✓ Estructura de directorios ✓ Acceso y uso de paquetes ✓ Archivos .jar 	<ul style="list-style-type: none"> ✓ Teórico ✓ Practico 	3
4B	Que los estudiantes sean capaces de comprender diseños UML dinámicos y estáticos	<ul style="list-style-type: none"> ✓ Introducción ✓ Que es un patrón de diseño ✓ Estructura de un patrón ✓ Clasificación de patrones de diseño 	<ul style="list-style-type: none"> ✓ Teórico ✓ Practico 	6
5B	Que los estudiantes sean capaces de comprender y escribir modelos de objetos simple	<ul style="list-style-type: none"> ✓ Patrones de diseño ✓ State ✓ Adapter ✓ Builder ✓ Singleton ✓ Strategy ✓ Facade 	<ul style="list-style-type: none"> ✓ Teórico ✓ Practico 	9
6B	Que los estudiantes tengan la capacidad de utilizar Patrones de diseño en modelos de mediana complejidad	<ul style="list-style-type: none"> ✓ Patrones de diseño ✓ Composite ✓ Decorator ✓ Factory method ✓ Composite ✓ Observer 	<ul style="list-style-type: none"> ✓ Teórico ✓ Practico 	9
7B	Que los estudiantes tengan la capacidad de utilizar Patrones de diseño en modelos reactivos	<ul style="list-style-type: none"> ✓ Patrones de diseño ✓ Command ✓ Memento ✓ Visitor ✓ Mediator ✓ Chain of Responsibility ✓ Flyweight 	<ul style="list-style-type: none"> ✓ Teórico ✓ Practico 	9
8B	Que los estudiantes tengan noción de patrones de Arquitectura	<ul style="list-style-type: none"> ✓ Diseño por capas ✓ Bus ✓ MVC/X ✓ PAC ✓ Reflexion 	<ul style="list-style-type: none"> ✓ Teórico ✓ Practico 	6
1C	Que los estudiantes sean capaces de comprender y aplicar Excepciones	<ul style="list-style-type: none"> ✓ Manejo de errores mediante excepciones ✓ Propagación de excepciones ✓ La jerarquía de excepciones 	<ul style="list-style-type: none"> ✓ Teórico ✓ Practico 	3



		<ul style="list-style-type: none"> ✓ Excepciones "checked" y "unchecked" ✓ Lanzamiento de excepciones ✓ Creación de clases de excepciones 		
2C	Que los estudiantes sean capaces de comprender y aplicar Threads	<ul style="list-style-type: none"> ✓ Procesos y threads ✓ Creación de threads ✓ La clase Thread ✓ La interfaz Runnable ✓ Sincronización ✓ Los métodos wait() y notify() ✓ Estados de un thread ✓ Bloqueo de threads ✓ Locks explícitos ✓ Locks de lectura/escritura 	<ul style="list-style-type: none"> ✓ Teórico ✓ Practico 	9
3C	Que los estudiantes sean capaces de comprender y aplicar Entrada / Salida	<ul style="list-style-type: none"> ✓ Streams ✓ El paquete java.io ✓ Streams de bytes y de caracteres ✓ Streams de bajo y alto nivel ✓ Leyendo y escribiendo ✓ Manejo de archivos ✓ Manejo de sockets 	<ul style="list-style-type: none"> ✓ Teórico ✓ Practico 	9
4C	Que los estudiantes sean capaces de comprender y aplicar JDBC	<ul style="list-style-type: none"> ✓ Drivers JDBC ✓ DriverManager ✓ Conexiones a bases de datos ✓ Transacciones ✓ Ejecución de sentencias SQL ✓ Recuperación de resultados ✓ Invocación de procedimientos almacenados ✓ Extensiones 	<ul style="list-style-type: none"> ✓ Teórico ✓ Practico 	9
5C	Que los estudiantes sean capaces de comprender y aplicar Colecciones	<ul style="list-style-type: none"> ✓ El framework de colecciones ✓ Las interfaces Collection, Set y List ✓ Implementaciones de Set y List ✓ La interfaz Map ✓ Implementaciones de Map ✓ Iteradores ✓ Wrappers ✓ Algoritmos 	<ul style="list-style-type: none"> ✓ Teórico ✓ Practico 	9
6C	Que los estudiantes puedan utilizar estructuras complejas en ambientes concurrentes	<ul style="list-style-type: none"> ✓ Uso de Queue ✓ Uso de Topic ✓ Uso de Colecciones Sincronizadas 	<ul style="list-style-type: none"> ✓ Teórico ✓ Practico 	6
1D	Que los estudiantes tengan la capacidad de entender y utilizar conceptos de Integración Continua	<ul style="list-style-type: none"> ✓ Introducción ✓ Ambientes Desarrollo, QA, Producción ✓ Hardware vs Virtualización ✓ Jenkins ✓ Git ✓ Sonar 	<ul style="list-style-type: none"> ✓ Teórico ✓ Practico 	6
2D	Que los estudiantes puedan comprender	<ul style="list-style-type: none"> ✓ Arquetipos ✓ Definición de DevOps 	<ul style="list-style-type: none"> ✓ Teórico 	3



	<i>todas los actores y definiciones del desarrollo de aplicaciones web</i>	<ul style="list-style-type: none"> ✓ Definición de Frontend ✓ Definición de Backend ✓ Definición de API ✓ Introducción a UX 		
3D	<i>Que los estudiantes puedan utilizar toda la potencia de HTML 5 en interfaces graficas</i>	<ul style="list-style-type: none"> ✓ Maquetar una página web usando HTML5 ✓ Aplicar diseños usando CSS3 ✓ Diferencias entre Float y Display ✓ Flexbox y Arquitectura de Información ✓ Metodología BEM ✓ Pseudo-Selectores ✓ Fuentes y Tipografías para Web ✓ Responsive Media Queries ✓ Preprocesadores de CSS 	<ul style="list-style-type: none"> ✓ Teórico ✓ Practico 	9
4D	<i>Que los estudiantes tengan la capacidad de desarrollar aplicaciones de diseño adaptativo y MVC</i>	<ul style="list-style-type: none"> ✓ Introducción a Responsive Design ✓ Los 5 patrones del Responsive ✓ "Best practices" de Responsive ✓ Crea una Single Page Application ✓ Aprende sobre las directivas de AngularJS ✓ Crea una aplicación basada en MVC ✓ Maneja de manera eficiente los formularios ✓ Conoce las ventajas del Data Binding ✓ Valida datos usando filtros ✓ Aprovecha Local Storage para persistencia de Datos ✓ Consume otros servicios Web desde Angular 	<ul style="list-style-type: none"> ✓ Teórico ✓ Practico 	11
5D	<i>Que los estudiantes tengan la capacidad de construir servicios REST, gestionar y administrar estos servicios</i>	<ul style="list-style-type: none"> ✓ Introducción a uso de API y Gateway API ✓ Uso de verbos HTML ✓ Concepto de parámetros ✓ Uso de JSON , JSONP y MSGPack ✓ Implementación de servicios Rest 	<ul style="list-style-type: none"> ✓ Teórico ✓ Practico 	14
6D	<i>Que los estudiantes tengan la noción de integración entre el modelo y la interface</i>	<ul style="list-style-type: none"> ✓ Implementación de una aplicación con AngularJS e integración con el Backend sobre servicios Rest utilizando JsonP y MSGPack. ✓ Conceptos de base de datos noSQL y bases de datos en memoria 	<ul style="list-style-type: none"> ✓ Teórico ✓ Practico 	20
7D	<i>Que los estudiantes tengan la capacidad de implementar una aplicación full stack con el uso de base de datos</i>	<ul style="list-style-type: none"> ✓ Introducción a ORMs ✓ Crea una arquitectura de información orientada a Documentos ✓ Autenticación con MongoDB ✓ Uso de Shell de MongoDB ✓ Integracion con aplicación web 	<ul style="list-style-type: none"> ✓ Teórico ✓ Practico 	12
Total				240