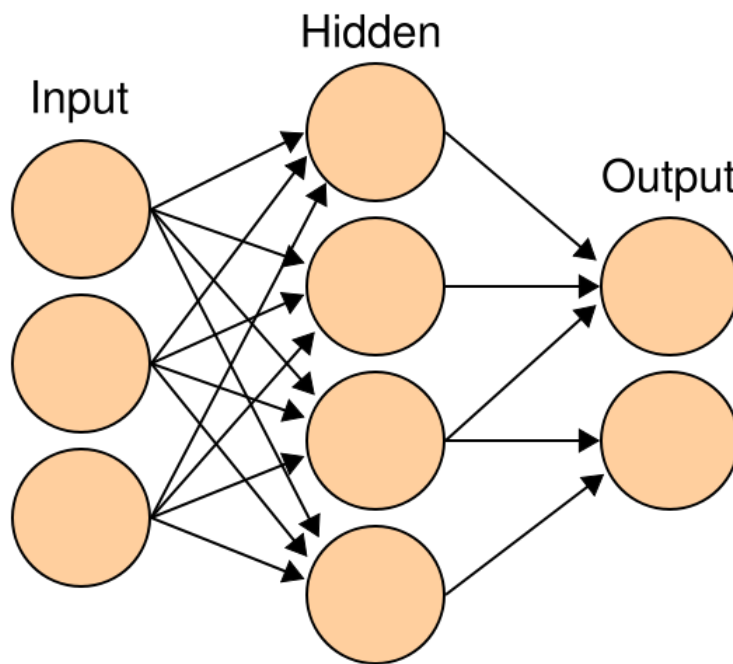


# Máster en ciencia de la computación



**Linux – Java – Android – PHP Web Services**

*Juan Antonio Breña Moral*  
*Versión 0.1*

# Índice de contenido

Introducción al Máster.....	3
Contenidos.....	3
Objetivos del Master.....	3
Bibliografía.....	3
Referencias.....	3
Sistemas Operativos.....	3
Lenguajes de programación.....	3
Ingeniería del software.....	3
Capítulo 1: Infraestructura.....	4
Ejercicios previos.....	4
Plazo de entrega.....	4
Tutoría 1: Resultados del primer capítulo.....	5
Resultados esperados:.....	5
Dudas esperadas.....	5
Dudas no esperadas.....	5
Resultados finales.....	5
Ayuda para usar Ubuntu.....	6
Capítulo 2: Fundamentos de Java.....	7
Nociones de Orientación a objetos con Java.....	7
Nociones del lenguaje de programación Java.....	7
Herramientas.....	8
Prácticas.....	8
Referencias.....	8
Tipos de datos básicos.....	8
Matemáticas.....	9
Bifurcaciones.....	9
Iteraciones.....	9

# Introducción al Máster

## **Contenidos**

Los contenidos del Máster son:

1. Linux
2. Java
3. Android
4. LAMP

Requerimientos:

1. Notebook

## **Objetivos del Master**

El objetivo del máster, es el aprendizaje del alumno en tecnologías usadas en la industria.

## **Bibliografía**

Ubuntu Unleashed

<http://book.opensourceproject.org.cn/distrib/ubuntu/unleashed/>

Piensa en Java, Tercera Edición

[http://rs115.rapidshare.com/files/34587579/Piensa\\_en\\_Java.rar](http://rs115.rapidshare.com/files/34587579/Piensa_en_Java.rar)

## **Referencias**

### **Sistemas Operativos**

<http://www.ubuntu.com/>

### **Lenguajes de programación**

<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

<http://www.java.com/es/>

<http://java.sun.com/>

<http://www.eclipse.org/>

<http://www.android.com/>

<http://www.apache.org/>

<http://php.net/>

<http://www.mysql.com/>

# Ingeniería del software

<http://code.google.com/intl/es/>

# Capitulo 1: Infraestructura

## ***Ejercicios previos***

El alumno debe enviar un email con las respuestas a las siguientes preguntas.

1. ¿Ubuntu es Linux?
  1. En caso afirmativo, en que te basas
2. ¿Que versión de Linux esta disponible? ¿Que licencia de mantenimiento tiene la ultima versión? ¿Cuanto tiempo dura? ¿Que es LTD? ¿Por que es interesante para un proyecto de desarrollo?
3. ¿Que diferencias observas entre Windows y Linux?
4. ¿Tu portátil puede arrancar S.O. Desde Pendrives?
5. ¿Que proyectos Open Source permiten cargar sistemas operativos en ISO sobre pendrive?
6. [PRACTICA] Instala la ultima versión de ubuntu en un pendrive
7. [PRACTICA] Instala ubuntu sobre tu portátil usando un pendrive
8. ¿Que es OOP?
9. Diferencias entre C++ y Java
10. [PRACTICA] Bajate el libro de Java y léete el capitulo 1 entero.
  1. ¿Que es la programación extrema?
  2. ¿Que es un applet?
  3. ¿Que es el multihilo?
11. ¿Cuantas versión de Java existen?
12. ¿Android es Java?
13. [PRACTICA] Descarga el ultimo JDK de java y el ultimo JRE de java para ubuntu.
  1. Emplea la consola de ubuntu usando el comando APT
14. ¿Como se comprueba que tienes instalado un JDK y no un JRE?
  1. ¿Cuales son las diferencias?
15. [PRACTICA] Instala Eclipse en tu maquina
16. ¿Que es SVN?
  1. ¿Para que se usa?
17. [PRACTICA] Crea un proyecto en Google code llamado “master” con sistema SVN
18. ¿Que plugins de eclipse recomiendas para poder sincronizar tus proyectos con tu proyecto en Google Code?
19. [PRACTICA] Crea un proyecto en Java y subelo a SVN

## ***Plazo de entrega***

Tienes una semana para terminar todo.

## **Tutoría 1: Resultados del primer capítulo**

### ***Resultados esperados:***

Se espera tener resuelto:

1. Instalación de Linux
  1. Aprender a instalar Linux desde USB
2. Instalación de Java y JRE.
  1. Distinguir diferencias
3. Instalación de IDE de Java
  1. Instalación de Eclipse
4. Instalación de otros elementos de desarrollo de software
  1. Herramienta SVN
  2. Creación de cuenta SVN sobre servicio online
5. Ejercicios de Desarrollo
  1. Hola Mundo

### ***Dudas esperadas***

Se espera tener dudas sobre:

1. Java
  1. Estructura del programa mas sencillo en Java
2. Software Engineering
  1. SVN

### ***Dudas no esperadas***

Las dudas que no se planificaron son:

1. Linux
  1. Como lanzar programas de Linux desde linea de comandos
  2. Como moverse por el sistema de ficheros

## Resultados finales

1. Entorno de desarrollo terminado
2. Herramientas colaborativas funcionando
3. Generación de documentación de código fuente

## ***Ayuda para usar Ubuntu***

Es necesario familiarizarse con el entorno de desarrollo de ubuntu. Para ello es necesario conocer:

1. Sistema de ficheros
  1. `cd PATH`  
para acceder a una carpeta
  2. `cd ..`  
para salir de una carpeta
  3. `ls`  
para listar el contenido de una carpeta
2. Java
  1. Ciertos programas como por ejemplo BlueJ es necesario ejecutarlos así:  
`./bluej`
  2. Para ejecutar un programa en `.jar`, es necesario ejecutar:  
`java -jar bluej.jar`

## **Capítulo 2: Fundamentos de Java**

### ***Nociones de Orientación a objetos con Java***

En este capítulo, se aprenderán los conceptos de orientación a objetos

1. Orientación a objetos
  1. Concepto de clase
    1. Propiedades
    2. Métodos
    3. Eventos
    4. Constructores.
  2. Mecanismos de orientación a objeto
    1. Herencias
    2. Polimorfismo.
    3. Interfases y Clases Abstractas.
  3. Diseño de objetos
    1. UML

### ***Nociones del lenguaje de programación Java***

1. Elementos básicos
  1. Variables
  2. Tipos de datos
  3. Operadores Matemáticos, Relacionales y Lógicos.
  4. Comentarios
2. Estructuras de control
  1. Condicionales if/else.
  2. Uso de los ciclos for y while.
  3. Uso de switch.
3. Matemáticas con Java
4. Gestión de errores
  1. Definición de Errores y Bloques try/catch/finally.
5. IO
  1. Datos de Entrada



6. Estructuras de datos
  1. Arrays
  2. Vectores
  3. Hashtables
  4. Otras estructuras de Datos.

## **Herramientas**

Para aprender a usar estos elementos se emplearan las siguientes elementos:

1. BlueJ
  1. <http://www.bluej.org/>
2. Jeliot
  1. <http://cs.joensuu.fi/jeliot/>
3. UML
  1. <http://argouml.tigris.org/>

## **Practicas**

1. Desarrollar ejemplos de:
  1. OOP
    1. Crear un proyecto llamado **TolkienRaces** donde se cree una clase padre donde se definan los atributos básicos de las razas del mundo del Señor de los anillos. Sobre dicha clase, se crearan 3 clases que hereden. Dichas clases serán Humanos, Orcos, Hobbits y Elfos. Define una serie de métodos base para todas las razas y define una serie de métodos propios por raza. Crea un programa principal que instancie objetos de todas las razas. Este proyecto crealo desde BlueJ para familiarizarte en el diseño de soluciones UML.
  2. Elementos básicos del lenguaje
    1. Debido a que el alumno, tiene los conceptos de Clase, Programa Principal, Tipos de datos y Matemáticas elabora un proyecto que solucione:
      1. Teorema de pitágoras:
        1. Clase que soluciona el problema de pitagoras. Usar una clase que soluciona la problemática y otra que lo usa.
        2. <http://www.juntadeandalucia.es/averroes/iesarrojo/matematicas/pitagoras.htm>
      2. Trigonometría:

1. Clase que soluciona un problema de triángulos. Usar una clase que soluciona la problemática y otra que lo usa.
  2. <http://centros5.pntic.mec.es/~marque12/matem/funciones/seno7.htm>
  3. Interpolación:
    1. Clase que soluciona un problema de interpolación. Usar una clase que soluciona la problemática y otra que lo usa.
    2. <http://centros5.pntic.mec.es/~marque12/matem/funciones/funciones3.htm>
  4. Funciones:
    1. <http://centros5.pntic.mec.es/~marque12/matem/funciones/funciones6.htm>
  5. Fractales:
    1. <http://www.sc.ehu.es/sbweb/fisica/cursoJava/numerico/recurso/recurso.htm>
  6. Sacar los números primos de los 100 primeros números
  7. Sacar los 10 primeros números de Fibonacci
  8. Dada una nota de un examen, saber si esta aprobado o suspenso. Dado un valor, graduar su nota en suspenso, bien, notable, sobresaliente o matricula de honor.
  9. Dados 100 números, sacar su raíz cuadrada de cada uno de ellos
2. Buscar en internet ejemplos con BlueJ y probarlos
  3. Probar los ejemplos básicos del lenguaje a través de Jeliot

## **Referencias**

### **Tipos de datos básicos**

<http://java.sun.com/docs/books/tutorial/java/nutsandbolts/datatypes.html>

<http://java.sun.com/javase/7/docs/api/java/lang/String.html>

<http://java.sun.com/javase/7/docs/api/java/lang/Integer.html>

<http://java.sun.com/javase/7/docs/api/java/lang/Float.html>

<http://java.sun.com/javase/7/docs/api/java/lang/Double.html>

### **Matemáticas**

<http://java.sun.com/javase/7/docs/api/java/lang/Math.html>

### **Bifurcaciones**

<http://java.sun.com/docs/books/tutorial/java/nutsandbolts/if.html>

### **Iteraciones**

<http://java.sun.com/docs/books/tutorial/java/nutsandbolts/for.html>

<http://java.sun.com/docs/books/tutorial/java/nutsandbolts/switch.html>

<http://java.sun.com/docs/books/tutorial/java/nutsandbolts/while.html>

## Tutoría 2: Resultados del primer capítulo

### **Resultados esperados:**

En la tutoría, se estuvo repasando los conceptos de OOP sobre BlueJ. Se propuso hacer un programa de Rally que sabe que coche llega a la meta o no dependiendo de la autonomía de cada coche medido en Km. Y la distancia del viaje.

#### 1. Ejercicios de coches

##### 1. Coche

1. Ruedas
2. Frenos
3. NivelFreno
4. Aceleracion
5. NivelAceleracion
6. Puertas
7. DisponeMaletero
8. CapacidadMaletero
9. Autonomia

##### 2. Hijos

###### 1. Renault

1. Si Maletero
2. Capacidad: 500
3. Autonomia: 300

###### 2. Ford

1. No
2. CapacidadMaletero: 0
3. Autonomia: 400

###### 3. Opel

1. Si
2. CapacidadMaletero: 700

3. Autonomia: 500
3. Viaje
  1. Propiedades
    1. Distancia
  2. Metodo
    1. getWinner
      1. Define que coche tiene mas autonomia para llegar a la meta
4. Juego
  1. 3 Objetos de tipo Ford, Renault y Opel
  2. Definimos la distancia de un objeto Viaje
  - 3.

Ejemplo de parametros:

1. Metodo sin parametros
  1. Ejemplo de la puerta.
    1. Estas? Si/No
    2. Cuantos hay en la habitacion? 5/2/Ninguno
2. Metodo con parametros
  1. Ejemplo de la puerta
    1. Abrete
      1. Sesamo

¿Como se define una funcion?

Las funciones usan return

Siempre poner ; en cada sentencia

A través del ejemplo, se sabria manejar con la estructura de control IF;

SI ( (PRGUNTA 1) O (PREGUNTA 2) O (PREGUNTA 3))

PREGUNTA: 1 o 0

2 preguntas

0 0 = 0

$$0 \ 1 = 1$$

$$1 \ 0 = 1$$

$$1 \ 1 = 1$$

Y

$$0 \ 0 = 0$$

$$0 \ 1 = 0$$

$$1 \ 0 = 0$$

$$1 \ 1 = 1$$

Ejercicios de las notas.