



Presentación de la asignatura.

# TAP 2018-2019

Germán Bordel  
Departamento de Electricidad y Electrónica

<http://gtts.ehu.es/German>  
[german.bordel@ehu.es](mailto:german.bordel@ehu.es)

**Importante:** modo de comunicación disponible 100%  
siempre acuso recibo.

# Generalidades

## Guión de consideraciones previas “tipo Bolonia”

- Buen clima – comunicación – diálogo – atención – orientación
- 
- Relación con otras asignaturas: es “transversal”.
- Necesidades previas: lo veremos más adelante
- Competencias--- muchas puntuales ⇨
  - competencias básicas + aprendizaje autónomo
  - “La reflexión es importante en el diseño de software”
- Evaluación a lo largo del curso
- 
- El tipo grupo --- trabajo aislado o en grupo.
- 
- Toma de decisiones por los alumnos ⇨ ejercicio final

# Calendario / actividades

Septiembre					Octubre					Noviembre					Diciembre								
L	M	M	J	V	L	M	M	J	V	L	M	M	J	V	L	M	M	J	V				
[1]	10	11	12	13	14	[4]	1	2	3	4	5	[8]			1	2	[13]	3	4	5	6	7	
[2]	17	18	19	20	21	[5]	8	9	10	11	12	[9]	5	6	7	8	9	[14]	10	11	12 <sup>lab</sup>	13	14
[3]	24	25	26	27	28	[6]	15	16	17	18	19	[10]	12	13	14 <sup>lab</sup>	15	16	[15]	17	18	19 <sup>lab</sup>	20	21
						[7]	22	23	24	25	26	[11]	19	20	21 <sup>lab</sup>	22	23						
						[8]	29	30	31			[12]	26	27	28 <sup>lab</sup>	29	30						

N no lectivo, F fiesta, E estudio, X exámenes, C clase, CA clase anulada

Clases: {aula: 0.20S} {horario: jueves 10:40-11:30, viernes 9:40-11:30}

Laboratorios: {aula: 0.22} {horario: 15:00-17:30}

Tutorías: {con las excepciones que se indiquen en GAUR, generalmente, J 8:00-10:00, V 8:00-9:00, JV 12:00-13:30}

\* Si a lo largo del curso se produjera alguna variación puntual, la información actualizada se encontrará siempre en la web de la asignatura.

## Clases teóricas/ Seminarios / prácticas de aula /prácticas de ordenador

Dada la naturaleza de la asignatura, la teoría es inseparable de la práctica. Las clases se desarrollarán intercalando constantemente proyecciones y el sistema de desarrollo integrado Netbeans para ejemplificar cada concepto. El alumno puede reproducir en paralelo con su portátil lo que se va haciendo, y plantear en todo momento las dificultades que se le planteen (aquellas que no sean de interés general quedarán para las tutorías). Respecto a los tres tipos de docencia en aula, dado que no hay separación en subgrupos, podrá haber cambios en la medida de lo razonable respecto a la distribución preasignada (siempre con suficiente preaviso).

## Evaluación

- 20% Test sobre los elementos básicos (tras ver el tema 5)
- 80% ejercicio en ordenador (el día oficial del examen)
- Puede realizarse un proyecto de desarrollo de un programa planteado por el alumno, a validar por el profesor (plazo máximo una semana antes de cierre de actas). En este caso se reduce a 50% en valor del ejercicio/examen final y se aplica el 30% a este trabajo.
- Se obtendrán puntos por resolver en pizarra ejercicios de diario y otras participaciones en clase. (sí, esto pasa de 100%, pero es que en programación los errores son inevitables y por tanto hay que darles un margen)

# Prerrequisito

Es necesario saber programar (mejor “con soltura”) en un lenguaje cualquiera.

Es conveniente hacerlo con “conocimiento de causa”, es decir sabiendo de algorítmica y de estructuras de datos.

Si se parte de cero, la Wikiversity nos puede ayudar:

1. Comenzar con "**Introduction to Computer Science**" ([https://en.wikiversity.org/wiki/Introduction\\_to\\_Computer\\_Science](https://en.wikiversity.org/wiki/Introduction_to_Computer_Science))  
Los 5 primeros capítulos son de "cultura general", pero a partir del sexto es una introducción interesante a la solución de problemas con programación.
- 2.
3. Continuar con "**Introduction to Programming**" ([https://en.wikiversity.org/wiki/Introduction\\_to\\_Programming](https://en.wikiversity.org/wiki/Introduction_to_Programming)).
4. Seguir con "**Introduction to Algorithms**" ([https://en.wikiversity.org/wiki/Introduction\\_to\\_Algorithms](https://en.wikiversity.org/wiki/Introduction_to_Algorithms))

Es conveniente seguir con "estructuras de datos", ya que es fundamental para programar con un conocimiento de los elementos que se utilizan, pero en la Wikiversity se desarrolla en lenguaje C, y no es adecuado a nuestro caso. En el curso veremos unas cuantas estructuras de un modo que podríamos llamar "usuario".

De este modo, se cuenta con unas bases de programación, y si se quiere empezar a ver el lenguaje Java también con la Wikiversity: "**Introduction to Programming in Java**" ([https://en.wikiversity.org/wiki/Introduction\\_to\\_Programming\\_in\\_Java](https://en.wikiversity.org/wiki/Introduction_to_Programming_in_Java)).

# Bibliografía

Todo estará en la web

•No se recomienda ningún libro en particular (quedan "obsoletos" en meses, y es contraproducente aprender a hacer las cosas de modos ya superados. Estamos en Técnicas **Actuales** de Programación, cada curso cambia respecto al anterior en unas cuantas cosas. De todos modos, libros hay, obviamente, y si alguien esta interesado se puede analizar la oferta del momento.)

•Se recomienda usar Internet (quizás para otras temáticas sea una fuente poco fiable; pero para la nuestra es "la navaja suiza definitiva")

•Principales referencias "raíz":  
<http://docs.oracle.com/javase/tutorial/>  
[https://en.wikipedia.org/wiki/Software\\_design\\_pattern](https://en.wikipedia.org/wiki/Software_design_pattern)  
[https://en.wikipedia.org/wiki/Software\\_engineering](https://en.wikipedia.org/wiki/Software_engineering)

•Otros recursos de interés:

Aquellos a los que nos lleven los buscadores, siempre que apliquemos el sentido común en su valoración. Hay multitud de foros específicos de los temas tratados en la asignatura.  
(en la página de la asignatura encontraréis también todos los enlaces a la revista Java Magazine que es siempre interesante.)

The screenshot shows a university profile page for Germán Bordel. On the left, there are navigation menus for 'PERFILES' (GTTS, linkedin, Twitter, Google+) and 'PERFILES BIBLIOGRÁFICOS' (GTTS, Google Citation, Microsoft Academic, Semantic Scholar, DBLP, ResearchGate, Academia, ResearcherID, ArnetMiner). The main content area includes a photo of Germán Bordel, his title 'Profesor Agregado de Universidad', and his affiliation with the 'Departamento de Electricidad y Electrónica' at the 'Facultad de Ciencia y Tecnología' of the 'Universidad del País Vasco'. Contact details for postal address, telephones, and email are provided. Below this, a 'DOCENCIA' section lists 'Asignaturas activas durante este curso' in three columns: 'T.A.P. Técnicas Actuales de Programación' (with sub-item 'AC Arquitectura de Computadoras'), 'IC Introducción a la Computación (sólo laboratorio)', and 'BD&POO Bases de Datos y Programación Orientada a Objetos (parte POO)'. A 'Histórico de asignaturas' section follows, with a note that data starts from 2016 and a link to view the full history. A 'Proyectos fin de grado/máster y tesis doctorales' section is also present. A green callout box points to the 'IC' course box with the text: 'Acceso a la página específica de la asignatura ("reflejo" en Moodle/Egela)'. A green arrow also points from the 'T.A.P.' box to the 'Histórico de asignaturas' section.

Acceso a la página específica de la asignatura ("reflejo" en Moodle/Egela)

# Temario

El temario es, en su mayor parte, el correspondiente a un curso de Java, pero este lenguaje "implementa" y facilita técnicas de programación que serán destacadas y explicadas en cada momento, conforme sea procedente. El curso debe incluir los temas 1..10. Los temas "An" sólo podrán verse si el desarrollo de los anteriores ha podido ir rápido, y representan temas "a elegir".

Tema	Contenido
1	<b>1.- INTRODUCCIÓN.</b> 1.1.- Java, XML e Ingeniería del software. 1.2.- Origen y características de Java. 1.3.- La máquina virtual. 1.4.- Obtención e instalación del software.
2	<b>2.- USO DEL ENTORNO DE DESARROLLO.</b> 2.1.- Ejemplo inicial: el programa "HolaMundo". 2.2.- Bibliotecas de clases: su estructura. 2.3.- Compilación y ejecución. 2.4.- Generación de documentación. 2.5.- Desensamblado y decompilación.

Tema	Contenido
6	<b>6.- INTERFACES GRÁFICOS.</b> 6.1.- Componentes AWT y SWING. 6.2.- Emplazamiento de componentes. LayoutManagers.
3	<b>3.- ELEMENTOS BÁSICOS DEL LENGUAJE.</b> 3.1.- Introducción. 3.2.- Tipos de datos, identificadores y literales. 3.3.- Operadores. 3.4.- Sentencias.
4	<b>4.- ELEMENTOS RELACIONADOS CON LA ORIENTACIÓN A OBJETO.</b> 4.1.- Objetos y clases. 4.2.- Estructura de la definición de una clase. 4.3.- Herencia. 4.4.- Clases y métodos abstractos. 4.5.- Interfaces. 4.6.- Polimorfismo.
5	<b>5.- MECANISMO DE TRATAMIENTO DE EXCEPCIONES Y ERRORES.</b> 5.1.- Introducción. 5.2.- Control de excepciones. 5.3.- Generación excepciones. 5.4.- Definición de nuevas excepciones. 5.5.- Aserciones
6.4.	Gráficos.

Tema	Contenido
7	<b>7.- EL CONTENIDO DE LA BIBLIOTECA DE CLASES.</b> 7.1.- Introducción. 7.2.- El paquete java.lang. 7.3.- Clases de utilidad (java.util) /iterator/. 7.4.- Genéricos. 7.5.- Adendum: 1) uso de bibliotecas externas; 2) patrones diversos;.
8	<b>8.- ENTRADA Y SALIDA DE DATOS.</b> 8.1.- Introducción. 8.2.- Estructura de clases para E/S.[ejercicios] 8.3.- E/S de objetos. Serialización. 8.4.- Creación de nuevas clases de E/S.
9	<b>9.- HILOS.</b> 9.1.- Introducción. 9.2.- Ciclo de vida de un hilo. 9.3.- Distribución de la CPU. Prioridades. El problema de la "Inanición". 9.4.- Mecanismos de sincronización de hilos. El problema del interbloqueo. 9.5.- Agrupamientos de hilos. 9.6.- Estudio de un ejemplo de programación con hilos.
10	<b>10.- PROGRAMACION EN RED.</b> 10.1.- Sockets. 10.2.- Una aplicación cliente/servidor.

Tema	Contenido
A1	<b>A1.- CONFECCIÓN DE PÁGINAS PARA LA WEB.</b> A1.1.- Introducción. A1.2.- Presentación de información en Internet: HTTP, HTML A1.3.- Presentación avanzada: estilos CSS y adaptación a plataformas. A1.4.- Interacción en Internet: programación client-side/server-side, javascript/JSP...
A2	<b>A2.- JAVA Y XML.</b> A2.1.- Lenguajes de marcas. A2.2.- XML en la actualidad.. A2.3.- SAX, DOM, XSD, XSLT, XPATH... A2.4.- Soporte Java para XML
A3	<b>A3.- INGENIERÍA DEL SOFTWARE.</b> A3.1.- Introducción. A3.2.- Unified Modelling Language (UML). A3.3.- Patrones de software y marcos de desarrollo.
A4	<b>A4.- VARIOS.</b> A4.1.- Uso de métodos nativos en Java. A4.2.- Acceso a bases de datos en Java. A4.3.- APIs de interés.