

Presentación de la asignatura.

TAP 2017-2018

Germán Bordel
Departamento de Electricidad y Electrónica


http://gtts.ehu.es/German_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	[4]	2	3	4	5	6	[8]			1	2	1	[13]	4	5	6	7	8
	4	5	6	7	8	[5]	9	10	11	12	13	[9]	6	7	8lab	9	10	[14]	11	12	13lab	14	15
[1]	11	12	13	14	15	[6]	16	17	18	19	20	[10]	13lab	14	15	16	17	[15]	18	19	20	21	22
[2]	18	19	20	21	22	[7]	23	24	25	26	27	[11]	20	21	22lab	23	25		25	26	27	28	29
[3]	25	26	27	28	29	[8]	30	31				[12]	27	28	29lab	30							

Horario: Jueves: 10:40-11:30 , Viernes: 9:40-11:20 (aula: 0.14)

Lab: miércoles indicados 15:00-17:30 (aula: 0.22 lab. Multiuso IE)

Tutorías: J 8:00-10:00, V 8:00-9:00, JV 12:00-13:30 (con excepciones. Detalles en GAUR)

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)
- Opcionalmente 30% a detracer del 80% anterior en con el desarrollo de un programa planteado por el alumno, a validar por el profesor (plazo máximo de entrega: una semana antes de cierre de actas).
- Los ejercicios de diario y la participación en clase pueden puntuar hasta un 15% adicional. (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)
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. Continuar con "**Introduction to Programming**" (https://en.wikiversity.org/wiki/Introduction_to_Programming).
3. Seguir con "**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).

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")

PERFILES

GTTS
linkedin
Twitter
Google+

PERFILES BIBLIOGRÁFICOS

GTTS
Google Citation
Microsoft Academic
Semantic Scholar
DBLP
ResearchGate
Academia
ResearcherID
ArnetMiner

Germán Bordel, Profesor Agregado de Universidad
[Departamento de Electricidad y Electrónica](#)
[Facultad de Ciencia y Tecnología](#)
[Universidad del País Vasco](#)

Dirección postal
Germán Bordel
Dpto. Electricidad y Electrónica
Fac. de Ciencia y Tecnología UPV/EHU
48940 Leioa
SPAIN

Teléfonos
+34 946015365 (despacho)
+34 946013071 (fax)

E-mail
german.bordel@ehu.es

IN
VES
TI
GA
CIÓN
@
(GTTS)

DOCENCIA

Asignaturas activas durante este curso.

Grado en Ingeniería Electrónica / Grado en Física / Doble Grado IE+F

[T.A.P.](#) - Técnicas Actuales de Programación
[AC](#) - Arquitectura de Computadoras (sólo laboratorio)

Grado en Ingeniería Química

[IC](#) - Introducción a la Computación (sólo laboratorio)

Máster en Modelización Matemática, Estadística y Computación

[BD&POO](#) - Bases de Datos y Programación Orientada a Objetos (parte POO)

Histórico de asignaturas. [\(Ver\)](#)

Esta sección, iniciada en 2016, recoge algunas de las páginas de apoyo a asignaturas de cursos anteriores. No todo ha sido recuperable puesto que frecuentemente la web de cada curso sobrescribía la anterior.

En material más antiguo se remonta a 1997. No guardo constancia de cuál fue el primer curso con apoyo en la web, pero obviamente no pudo ser antes del 30 de abril de 1993, cuando [el CERN presentó la WWW de forma pública](#). En realidad no sería antes de que estuviesen disponibles los navegadores Netscape (oct.94) o Internet Explorer (ago.95), porque no recuerdo que los alumnos usaran [Mosaic](#). Eso sí, en un principio, con excepción de algún caso muy aislado, los alumnos no disponían de ordenador y modem telefónico en sus domicilios, por lo que accedían únicamente desde el equipamiento disponible en nuestros laboratorios.

Proyectos fin de grado/máster y tesis doctorales. [\(Ver\)](#)

Otras actividades docentes. [\(Ver\)](#)

Cursos diversos, Tutorías UNED, y otros

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

• Principales referencias "raíz":

<http://docs.oracle.com/javase/tutorial/>
https://en.wikipedia.org/wiki/Software_design_pattern
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.)

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	1.- INTRODUCCIÓN. 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	2.- USO DEL ENTORNO DE DESARROLLO. 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	6.- INTERFACES GRÁFICOS. 6.1.- Componentes AWT y SWING. 6.2.- Emplazamiento de componentes. LayoutManagers.
3	3.- ELEMENTOS BÁSICOS DEL LENGUAJE. 3.1.- Introducción. 3.2.- Tipos de datos, identificadores y literales. 3.3.- Operadores. 3.4.- Sentencias.
4	4.- ELEMENTOS RELACIONADOS CON LA ORIENTACIÓN A OBJETO. 4.1.- Objetos y clases. 4.2.- Estructura de la definición de una clase. 4.3.- Herencia. 4.4.- Encapsulamiento (ámbitos de accesibilidad). 4.5.- El bloque static y los atributos static y final. 4.6.- Instanciación, inicialización y eliminación de objetos. 4.7.- Clases y métodos abstractos. 4.8.- Interfaces. 4.9.- Polimorfismo. 4.10.- Arrays*. 4.11.- Enumeraciones.
5	5.- MECANISMO DE TRATAMIENTO DE EXCEPCIONES Y ERRORES. 5.1.- Introducción. 5.2.- Control de excepciones. 5.3.- Generación excepciones. 5.4.- Definición de nuevas excepciones. 5.5.- Aserciones
	6.3.- Comunicación entre componentes. 6.4.- Gráficos. 6.5.- Applets.

Tema	Contenido
7	7.- EL CONTENIDO DE LA BIBLIOTECA DE CLASES. 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	8.- ENTRADA Y SALIDA DE DATOS. 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	9.- HILOS. 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	10.- PROGRAMACION EN RED. 10.1.- Sockets. 10.2.- Una aplicación cliente/servidor.

Tema	Contenido
A1	A1.- CONFECCIÓN DE PÁGINAS PARA LA WEB. 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	A2.- JAVA Y XML. 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	A3.- INGENIERÍA DEL SOFTWARE. A3.1.- Introducción. A3.2.- Unified Modelling Language (UML). A3.3.- Patrones de software y marcos de desarrollo.
A4	A4.- VARIOS. A4.1.- Uso de métodos nativos en Java. A4.2.- Acceso a bases de datos en Java. A4.3.- APIs de interés.