

## Fundamentos de Informática

Código: 103806  
Créditos ECTS: 9

**2024/2025**

Titulación	Tipo	Curso
2502441 Ingeniería Informática	FB	1

## Contacto

Nombre: Francesc Xavier Roca Marva

Correo electrónico: [xavier.roca@uab.cat](mailto:xavier.roca@uab.cat)

## Equipo docente

Francesc Xavier Roca Marva

Miquel Àngel Senar Rosell

Sandra Adriana Mendez

Daniel Munzon Gil

Diego Alejandro Velazquez Dorta

Roberto Benavente Vidal

## Idiomas de los grupos

Puede consultar esta información al [final](#) del documento.

## Prerrequisitos

Dado que la asignatura es de introducción asumirá que no se posee ningún tipo de conocimiento previo sobre la materia. Es responsabilidad de la propia asignatura proporcionar a los alumnos un medio para adquirir los conocimientos descritos en el apartado de contenidos de la asignatura (apartado 6 de esta guía).

A pesar de ello es recomendable:

- haber cursado el bachillerato tecnológico,
- tener conocimientos a nivel de usuario de algún tipo de plataforma (windows, mac o linux)
- tener acceso a un ordenador, si es portátil mejor

## Objetivos y contextualización

Esta asignatura tiene un carácter general e introductorio a la informática. Se divide en dos grandes ejes: el primero comprende el estudio de los aspectos metodológicos de la programación y el aprendizaje de un

lenguaje de alto nivel y el segundo consiste en la introducción de los conceptos básicos de la estructura de los computadores y de los principios de diseño. Por tanto, los objetivos generales que se proponen para la asignatura son los siguientes:

- Proporcionar una visión general de la informática, introduciendo sus antecedentes históricos y explicando los conceptos básicos: hardware, software, sistema operativo, estructura de un ordenador, algoritmo, programa y lenguajes de programación, compilador, intérprete, etc.
- Familiarizar al alumno / a con el ordenador.
- Entender el ciclo de vida del software: analizar el problema (entender lo que se nos pide), diseño (proponer una solución al problema), implementación (codificación en un lenguaje de programación de la solución elegida), prueba (realización de un test de manera sistemática para asegurar la corrección de la solución implementada).
- Dotar al alumno / a de la capacidad de diseño de algoritmos para la resolución de problemas, introduciendo de manera progresiva y sistemática una metodología rigurosa y estructurada de programación, basada fundamentalmente en la técnica del diseño descendente de algoritmos.
- Introducir al alumno / a en un lenguaje de programación real. Se pretende que el alumno perciba la diferencia entre la flexibilidad de la notación pseudo-algorítmica que se emplea en los primeros temas y la estricta sintaxis de un lenguaje de programación real, tanto en sus aspectos léxicos (palabras válidas del lenguaje ), sintácticos (reglas para combinarlas) y semánticos (significado de las mismas).
- Habituar al alumno / a desarrollar programas siguiendo unas normas de estilo tendentes a conseguir programas de calidad. Dentro de estas normas de estilo engloban aquellas que facilitan la compresión del código, como pueden ser el uso de comentarios, la indentación del código, la utilización de nombres adecuados para los tipos de datos, etc.
- Proporcionar una visión de las unidades funcionales del ordenador y su interconexión.
- Ofrecer al alumno / a un conocimiento amplio del funcionamiento del ordenador a bajo nivel.
- Introducir el código máquina y el lenguaje ensamblador y mostrar la traducción de las estructuras básicas de alto nivel a bajo nivel.
- Mostrar los diferentes niveles de hardware y software necesarios para el funcionamiento de los ordenadores y su interconexión.

## Competencias

- Adquirir hábitos de pensamiento.
- Capacidad para comprender y dominar los conceptos básicos de matemática discreta, lógica, algorítmica y complejidad computacional, y su aplicación para la resolución de problemas propios de la ingeniería.
- Conocimiento de la estructura, organización, funcionamiento e interconexión de los sistemas informáticos, los fundamentos de su programación, y su aplicación para la resolución de problemas propios de la ingeniería.
- Conocimiento de las materias básicas y tecnologías, que capaciten para el aprendizaje y desarrollo de nuevos métodos y tecnologías, así como las que les doten de una gran versatilidad para adaptarse a nuevas situaciones.
- Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.

## Resultados de aprendizaje

1. Aplicar los principios básicos de la programación de computadores a la resolución de problemas de ingeniería.
2. Comprender y aplicar los principios básicos de la lógica de los computadores.
3. Conocer los conceptos básicos de la estructura y programación de los computadores.
4. Conocer y aplicar los principios de la interconexión de sistemas.
5. Conocer y saber utilizar los sistemas operativos, bases de datos y programas de uso común en la ingeniería.
6. Desarrollar la capacidad de análisis, síntesis y prospectiva.

7. Reconocer e identificar los métodos, sistemas y tecnologías propios de la ingeniería informática.

## Contenido

### BLOQUE 1: INTRODUCCIÓN A LA INFORMÁTICA

Tema 1: Introducción a la informática

Historia, Estructura funcional del ordenador, Programas / instrucciones, Niveles conceptuales del ordenador.

### BLOQUE 2: INTRODUCCIÓN A LA PROGRAMACIÓN

Tema 2: Resolución de problemas: introducción a la algorítmica y la programación.

Introducción a la resolución de problemas. Concepto de algoritmo. Fases en el desarrollo de algoritmos. La programación como disciplina de ingeniería. Ciclo de vida del software. Elementos básicos de un algoritmo. Herramientas para la representación de algoritmos. Lenguajes de programación. Clasificación. Traductores de lenguajes: Compiladores e intérpretes.

Tema 3: Conceptos básicos y estructuras de control

Estructura Secuencial. Estructuras de Selección. Estructuras iterativas o repetitivas.

Tema 4: Subprogramas. Procedimientos y funciones

El concepto de subprograma como abstracción de operaciones. Localidad, neamente, ámbito y visibilidad. Definición de funciones. Llamadas a funciones. Declaración o prototipo de funciones. Tipo de argumentos: de entrada, de salida, de entrada / salida. Paso de argumentos a funciones: por valor y por referencia.

Tema 5: Tipo de datos

Definición de variables y constantes. Tipo de datos fundamentales. El tipo Array. Arrays unidimensionales. Arrays multidimensionales. Las cadenas de caracteres. Funciones relacionadas. El tipo registro. Arrays de registros.

### BLOQUE 3: INTRODUCCIÓN AL COMPUTADOR

Tema 6: Representación de la información en el ordenador

Sistemas de numeración: decimal-binario, operatividad, sistemas octal y hexadecimal. Representación de datos numéricos (C1 / C2). Números reales. Representación de texto.

Tema 7: Introducción a la arquitectura del ordenador

Unidades funcionales del computador. Funcionamiento interno de un computador. El procesador. Flujo de ejecución de la instrucción. El sistema de entrada / salida: Entrada / Salida Programada. Concepto de interrupción. El sistema de memoria. Estructura jerárquica de la memoria. Visión de un PC.

Tema 8: Lenguaje máquina y ensamblador

Direccionamiento. Computador didáctico elemental. Lenguaje ensamblador.

Tema 9: Periféricos

Tema 10: Sistema operativo

Evolución. Gestión: procesador, memoria, E / S, archivos

## Tema 11: Conceptos sobre redes de computadores

### Transmisión de datos. Redes de computadores. Internet

## Actividades formativas y Metodología

Título	Horas	ECTS	Resultados de aprendizaje
Tipo: Dirigidas			
Clases Presenciales / Teoría	42	1,68	2, 3, 4, 5, 7
Explicación y resolución de problemas	7,5	0,3	1, 2, 5, 7
Prácticas de laboratorio	14	0,56	1, 3, 7
Tipo: Supervisadas			
Resolución de problemas adicionales	15	0,6	1, 2, 7
Seguimiento y refuerzo en la resolución de los casos prácticos propuestos	10	0,4	1, 2, 3, 7
Tipo: Autónomas			
Estudio individual	62	2,48	2, 3, 4, 5, 7
Resolución de casos prácticos	28	1,12	1, 2, 7
Resolución de problemas de forma individual y grupal	22	0,88	1, 2, 3, 7

La asignatura tiene dos partes diferenciadas Programación y Computación. Cada parte tendrá una metodología docente adaptada a los contenidos docentes.

### Programación

La gestión de la docencia de la asignatura se hará a través del gestor documental Caronte (<http://caronte.uab.cat/>), que servirá para poder ver los materiales, gestionar los grupos de prácticas, hacer las entregas correspondientes, ver las notas, comunicarse con los profesores, etc. Para poder utilizarlo es necesario hacer los siguientes pasos:

1. Darse de alta como usuario dando el nombre, NIU, y una foto carnet en formato JPG. Si ya se ha dado de alta por alguna otra asignatura, no es necesario volver a hacer, puede ir al siguiente paso.
2. Incribirse en el tipo de docencia "Docencia de Fundamentos de Informática", dando como código de asignatura el proporcionado el primer día de clase.

En el desarrollo de la parte de programación seguirá la metodología de clase inversa. Esto significa que los / las estudiantes deberán trabajar algunos conocimientos antes de cada sesión para poder dejar para las sesiones presenciales aquellas cuestiones que sólo pueden suceder cara a cara. El punto de partida son las experiencias y los conocimientos de los propios participantes, que generan el marco conceptual de forma interactiva.

La actividad educativa se configura como un sistema de doble vía, en que el profesorado y los estudiantes construyen conjuntamente y comparten responsabilidades sobre el proceso de aprendizaje donde el objetivo es poder ver el máximo de casos posibles para entender la complejidad de lo que implica plantear una solución a un problema de programación.

- Antes de Clase. Los conceptos básicos se trabajan antes de las sesiones presenciales. Prepare la clase con los contenidos que se encuentra con una semana de antelación a la plataforma [Caronte](#). El material disponible será documentación escrita, contenido multimedia y actividades de aprendizaje. Paralelamente al material habrá un foro de discusión a [Caronte](#), moderado por el profesorado, para resolver dudas y las sesiones de tutoría. Cada estudiante puede marcar su ritmo revisando los materiales tantas veces como haga falta para adquirir estos conceptos. A modo indicativo cada estudiante debería dedicar unas 2 ó 3 horas para la preparación. Habrá un cuestionario para evaluar el logro de los conceptos explicados
- En Clase. El objetivo de las sesiones presenciales se pasó de la información al conocimiento. Pondremos el énfasis en crear conocimientos aplicables al análisis de problemas concretos, en que el profesorado pone a prueba los / las estudiantes, los cuales consolidan sus aprendizajes a partir de problemas. La estructura de las clases será:
  - Revisión de los conceptos básicos más importantes (15 min). El objetivo es acabar de resolver las dudas que pueda haber sobre aspectos teóricos no entendidos y hacer seguimiento de la adquisición de los conceptos básicos.
  - Propuesta de problemas (resto del tiempo de clase) El objetivo es experimentar haciendo una tarea práctica que permita aplicar los conceptos aprendidos para resolver retos y así compartir y crear conocimiento.
- Despues de Clase. El objetivo es consolidar el conocimiento. En la plataforma [Caronte](#) se podrán encontrar un conjunto de problemas evaluables de autocorrección que deben permitir a los/ las estudiantes profundizar en la comprensión y personalizar el conocimiento. El hecho de que sea auto-evaluable permite adecuar el ritmo de consolidación. La resolución debe permitir a los estudiantes reflexionar sobre el propio aprendizaje.

## Computación

La gestión de la docencia de la asignatura se hará a través del gestor documental Campus Virtual (<http://cv.uab.cat>), que servirá para poder ver los materiales, gestionar los grupos de prácticas, hacer las entregas correspondientes, ver las notas, comunicarse con los profesores, etc.

En el desarrollo de la asignatura se podrán diferenciar tres tipos de actividades docentes:

- Clases teóricas. Exposición en la pizarra de la parte teórica de cada tema del programa. La estructura típica de una clase magistral de este tipo será la siguiente: en primer lugar se hará una introducción donde se presentarán brevemente los objetivos de la exposición y los contenidos a tratar. Con el fin de proporcionar el contexto adecuado, la presentación se hará referencia al material expuesto en clases precedentes, por lo que se clarifique la posición de estos contenidos dentro del marco general de la asignatura. A continuación se desgranarán los contenidos objeto de estudio, incluyendo exposiciones narrativas, desarrollos formales que proporcionen los fundamentos teóricos, e intercalando ejemplos, que ilustren la aplicación de los contenidos expuestos. Se resaltarán los elementos importantes de manera que se sea capaz de distinguir lo relevante de los aspectos periféricos. Finalmente, se resumirán los conceptos introducidos y se elaborarán las conclusiones, incluyendo una valoración de en qué medida se han alcanzado los objetivos propuestos al principio de la lección.
- Clases prácticas. Todos los temas irán acompañados de una relación de problemas que el alumno debe intentar resolver. En este sentido, ya medida que el alumno vaya progresando en la profundidad de sus conocimientos, estos problemas serán poco a poco más complejos, permitiendo de esta manera apreciar claramente las ventajas de utilizar las herramientas metodológicas impartidas durante el curso. Aquellos que el profesor considere de mayor interés o en los que los alumnos encuentren mayor dificultad serán corregidos en la pizarra. Antes de cada sesión de problemas el profesor podrá proponer una lista de ejercicios que los alumnos deberán resolver y entregar antes de clase.
- Clases en laboratorio. La parte práctica de cada tema quedará completada con al menos una sesión en el laboratorio, donde el alumno deberá intentar resolver un problema concreto. Algunos de estos ejercicios deberán entregarse después de la clase. La resolución de estos ejercicios debe servir como

base de aprendizaje para poder realizar un caso práctico que se propondrá como trabajo de todo el curso.

### Competencias transversales

La competencia transversal que se trabajará y evaluará a lo largo del curso es la T01.02 - Desarrollar la capacidad de análisis, síntesis y prospectiva. Se trabajará a lo largo de las sesiones de los diferentes tipos de actividades previstas y se evaluará principalmente en las pruebas de evaluación y en la entrega de las prácticas.

Nota: se reservarán 15 minutos de una clase dentro del calendario establecido por el centro o por la titulación para que el alumnado rellene las encuestas de evaluación de la actuación del profesorado y de evaluación de la asignatura o módulo.

## **Evaluación**

### **Actividades de evaluación continuada**

Título	Peso	Horas	ECTS	Resultados de aprendizaje
Bloque A: Entrega de la práctica de programación	18% de la asignatura	6	0,24	1, 3, 6, 7
Bloque A: Evaluación continuada de los problemas propuestos	12% de la asignatura	4,5	0,18	1, 3, 6, 7
Bloque A: Examen Teórico Individual (parcial)	30% de la asignatura	2	0,08	1, 3, 6
Bloque B: Entrega de la práctica correspondiente a nivel máquina	12% de la asignatura	3	0,12	1, 3, 6, 7
Bloque B: Evaluación continuada de los problemas propuestos	4% de la asignatura	3	0,12	1, 2, 5, 7
Bloque B: Examen Teórico Individual (parcial)	24% de la asignatura	2	0,08	1, 3, 6, 7
Reevaluación Teoría Parte A y / o B: Prueba individual de Teoría (sólo para los alumnos que hayan suspendido la parte A y / o B)	30% correspondiente a parte A y 24% correspondiente a la parte B	4	0,16	1, 2, 3, 4, 5, 7

**IMPORTANTE:** Esta asignatura no contempla el sistema de evaluación única.

### Programación y recuperación de las actividades de evaluación programadas

La asignatura está dividida en dos bloques:

1. Las primeras nueve (9) semanas se dedicarán al estudio de la programación y contará un 60% del valor de la nota final (Bloque A).
2. Las siguientes seis (6) semanas se dedicarán al estudio del computador y contará un 40% del valor de la nota final (Bloque B).

Los dos bloques se evaluarán de manera independiente. Para obtener la nota final se deberán aprobar las dos partes por separado (Nota $>$  = 5).

La nota final de la asignatura será: 60% bloque A + 40% bloque B

### Programación (BLOQUE A)

Se realizan las siguientes actividades evaluables:

Actividad	Fecha	Recuperación	Porcentaje Bloque	Nota mínima	Porcentaje Asignatura
Examen Teórico (T) Individual	Consultar Planificación	Consultar Planificación	50 %	T $>$ = 5	30%
Cuestionario autoevaluación de logros conceptos	Semanalmente y Hasta examen de recuperación teoría	No	5%	No.	3%
Resolución Problemas (P) Individual	Hasta examen de recuperación teoría  SOLO se podrá acceder si previamente se ha realizado el cuestionario de consecución  (algunos tendrán fecha límite)	No	15%	No	9%
Práctica de Laboratorio (PL) Grupal	Consultar Planificación  Se compone de un test (PLT) y la corrección de la entrega final (PLF)	Consultar Planificación	30%	PL = 30% PLT + 70% PLF  PLT y PLF $>$ = 5	18%

### Computación (BLOQUE B)

Se realizan las siguientes actividades evaluables:

Actividad	Fecha	Recuperación	porcentaje Bloque	Nota mínima	porcentaje Asignatura
Examen Teórico (T) Individual	Consultar Planificación	Consultar Planificación	60%	T $>$ = 5	24%

Resolución Problemas (P) Individual	Consultar Planificación	No	10%	No	4%
Práctica de Laboratorio (PL) Grupa el	Consultar Planificación Tiene tres entregas, una por práctica P1, P2 y P3	Consultar Planificación	30%	PL= $0.33*P1+0.33*P2+0.33*P3$ P1,P2,P3 > 3.5 i PL>=5	12%

### Global para los dos bloques

Deberá superarse con una nota de 5 o superior, tanto la parte de teoría como la de prácticas, independientemente. Las prácticas son obligatorias, así como la asistencia a todas las sesiones. Para superar las prácticas es necesario hacerlo de una en una. Sólo en casos extraordinarios se podrá acceder a la reevaluación de las prácticas, siempre a criterio del profesorado de la asignatura.

La nota de cada bloque se obtiene de la suma ponderada de las actividades de evaluación con su correspondiente porcentaje. El resultado de la suma ponderada debe ser  $\geq 5$  por considerar aprobado cada bloque.

La parte de Teoría (T) se evaluará con un examen escrito individual. Existe un primer examen individual correspondiente al primer bloque (bloque A) y otro al segundo (bloque B). En caso de no superar alguna de estas pruebas habrá una recuperación el día que tenemos asignado en la semana de recuperaciones (en enero/febrero) donde se pueden recuperar las pruebas que no hayan sido superadas en los exámenes parciales. La nota mínima para superar cada prueba es 5.

Las fechas de evaluación y entrega de trabajos se publicarán en el gestor documental escogido por cada bloque (Caronte (bloque A) o Campus virtual UAB (bloque B)) y pueden estar sujetos a cambios de programación por motivos de adaptación a posibles incidencias. Se informará de dichos cambios mediante el gestor documental dado que se entiende que ésta es la plataforma habitual de intercambio de información entre profesorado y estudiantes.

### Procedimiento de revisión de las calificaciones

Los estudiantes tendrán derecho a la revisión de los exámenes teóricos (T). El lugar, fecha y hora de la revisión se publicará el día en que sean publicadas las notas. Si el estudiante no se presenta a esta revisión, no se revisará posteriormente. Sólo en casos justificados se podrá realizar una revisión a posteriori de la fecha fijada y siempre hasta un máximo de 7 días naturales después.

De las notas de las prácticas de laboratorio (PL) no se realizará revisión porque la evaluación se realiza delante de los estudiantes.

### Calificaciones

Nota Final: Ambos bloques se evaluarán de forma independiente. Para obtener la nota del expediente se tendrán que aprobar los dos bloques por separado (Nota  $\geq 5$ ).

La nota de cada bloque aparece de la suma ponderada según los criterios expuestos en el apartado de actividades de evaluación. Si la teoría (T) o las prácticas (PL) tienen una nota inferior a 5 la nota saldrá de las siguientes fórmulas: Mínimo (aplicar suma ponderada y 4,5).

La nota final del expediente académico se obtiene según el cuadro de calificación final:

		Nota > 5	Nota < 5	NA
Computación (Bloque B)	Nota > = 5	60% bloque A + 40% bloque B	Min(60% bloque A + 40% bloque B, 4,5)	Min(40% bloque B, 3,4)
	Nota < 5	Min(60% bloque A + 40% bloque B, 4,5)	Min(60% bloque A + 40% bloque B, 4,5)	Min(40% bloque B, 3,4)
	NA	min(60% bloque A, 3,4)	min(60% bloque A, 3,4)	NA

No Evaluable (NA): Cualquier estudiante que entregue una práctica o una actividad programada tendrá nota. Sólo se considerará no evaluable, en el global, en caso de no entregar actividad evaluable alguna. En caso de que el estudiante entregue una práctica o una actividad programada en uno de los bloques y no en el otro se considerará que del bloque que no haya presentado nada tendrá un 0.

Matrículas de honor:

Otorgar una calificación de matrícula de honor es decisión del profesorado responsable de la asignatura. La normativa de la UAB indica que las MH sólo podrán concederse a estudiantes que hayan obtenido una calificación final igual o superior a 9.00. y su número no puede superar el 5% del total de estudiantes matriculados.

En caso de que el número de estudiantes con nota superior o igual a 9 sean más del 5% del total de matriculados, los criterios a aplicar en el orden enumerado a continuación serán los siguientes:

1. Se priorizan los que no han ido a ninguna recuperación.
2. Se priorizan los que tienen nota >9 en los dos bloques.
3. Se priorizan los que tienen nota >9 en todas las partes de los 2 bloques.
4. Se priorizan por orden de nota media.

Irregularidades por parte de los/las estudiantes, copia y plagio

Sin perjuicio de otras medidas disciplinarias que se estimen oportunas, y de acuerdo con la normativa académica vigente, las irregularidades cometidas por una estudiante que puedan conducir a una variación de la calificación se calificarán con un cero (0). Por ejemplo, plagiar, copiar, dejar copiar... una actividad de evaluación implicará suspender esta actividad de evaluación con un cero (0). Las actividades de evaluación calificadas de esta forma y por esta razón no serán recuperables. Si es necesario superar cualquiera de estas actividades de evaluación para aprobar la asignatura, esta asignatura quedará suspendida directamente, sin oportunidad de recuperarla en el mismo curso.

Evaluación de los estudiantes repetidores

Quien repita la asignatura y tenga uno de los dos bloques aprobados de forma completa (Teoría (T) y Practiques (PL) del bloque, con notas >=5) no será necesario que vuelva a evaluarse de este bloque el curso siguiente.

Para el bloque A, si el curso anterior al actual se superó la asignatura "Metodología de la programación" y se aprobaron los ejercicios de repaso propuestos en aquella asignatura, se podrá realizar una evaluación de síntesis del bloque, sin tener que realizar todas las actividades de evaluación del bloque. La nota del bloque en este caso será de 5.

Para poder optar a esta evaluación diferenciada, la persona interesada debe pedirlo expresamente al profesorado mediante correo electrónico a: para el bloque de Programación superado (xavier.roca@uab.cat o robert.benavente@uab.cat), para el bloque de Computación superado (miquelangel.senar@uab.cat) a más tardar 15 días después del inicio de las clases de cada Bloque.

Es responsabilidad del estudiante comprobar que el profesorado responsable publica correctamente la nota obtenida el curso finalizado. En ningún caso se guardan notas de alguna de las actividades de evaluación de un bloque que haya quedado suspendido en un curso anterior, como podría ser el caso haber suspendido la teoría y haber superado las prácticas de un bloque.

Nota: La evaluación propuesta puede experimentar alguna modificación en función de posibles restricciones a la presencialidad que impongan las autoridades sanitarias.

## Bibliografía

- *J. Pujol: Algorismes i Programes.* Servei de publicacions de la UAB, 1996.
- *E. Valveny, R. Benavente, A. Lapedriza, M. Ferrer, J. García: Programació en Llenguatge C.* Amb 56 problemes resolts i comentats. Servei de publicacions UAB, 2009.
- *L. Joyanes Aguilar : Fundamentos de Programación: Algoritmos, Estructuras de Datos y Objetos.* 5<sup>a</sup> Edición, Mc. Graw-Hill, 2008.
- *L. Joyanes Aguilar, L. Rodriguez Baena, M. Fernández Azuela: Fundamentos de Programación. Libro de Problemas.* 2<sup>a</sup> Edición, Mc Graw-Hill, 2003.
- *B.W. Kernighan, D.M. Ritchie: El lenguaje de programación C.* 2<sup>a</sup> Edición, Prentice Hall, 1986.
- *L. Joyanes, A. Castillo, L. Sánchez, I. Zahonero: Programación en C: libro de problemas,* Mc Graw-Hill, 2005.
- *L. Joyanes, I. Zahonero: Programación en C: metodología, estructura de datos y objetos,* Mc Graw-Hill, 2001.
- *B.W. Kernighan, R. Pike: La Práctica de la Programación.* Pearson Educación, 2000.
- *A. Prieto, A. Lloris, J.C. Torres: Introducción a la Informática.* Mc Graw-Hill, 4<sup>a</sup> Edición, 2006.
- *A. Prieto, B. Prieto: Conceptos de Informática.* Mc Graw-Hill, Schaum, 2006.

## Software

### Programación (BLOQUE A) & Computación (BLOCK B)

Microsoft Visual Studio <https://azureforeducation.microsoft.com/devtools>

## Lista de idiomas

Nombre	Grupo	Idioma	Semestre	Turno
(PAUL) Prácticas de aula	411	Catalán	primer cuatrimestre	mañana-mixto
(PAUL) Prácticas de aula	412	Catalán	primer cuatrimestre	mañana-mixto
(PAUL) Prácticas de aula	431	Catalán	primer cuatrimestre	mañana-mixto
(PAUL) Prácticas de aula	432	Catalán	primer cuatrimestre	mañana-mixto

(PAUL) Prácticas de aula	451	Catalán	primer cuatrimestre	tarde
(PAUL) Prácticas de aula	452	Catalán	primer cuatrimestre	tarde
(PAUL) Prácticas de aula	471	Catalán	primer cuatrimestre	tarde
(PLAB) Prácticas de laboratorio	411	Catalán/Español	primer cuatrimestre	mañana-mixto
(PLAB) Prácticas de laboratorio	412	Catalán/Español	primer cuatrimestre	mañana-mixto
(PLAB) Prácticas de laboratorio	413	Catalán/Español	primer cuatrimestre	mañana-mixto
(PLAB) Prácticas de laboratorio	414	Catalán/Español	primer cuatrimestre	mañana-mixto
(PLAB) Prácticas de laboratorio	415	Catalán/Español	primer cuatrimestre	tarde
(PLAB) Prácticas de laboratorio	416	Catalán/Español	primer cuatrimestre	tarde
(PLAB) Prácticas de laboratorio	417	Catalán/Español	primer cuatrimestre	tarde
(PLAB) Prácticas de laboratorio	418	Catalán/Español	primer cuatrimestre	tarde
(PLAB) Prácticas de laboratorio	419	Catalán/Español	primer cuatrimestre	mañana-mixto
(PLAB) Prácticas de laboratorio	420	Catalán/Español	primer cuatrimestre	mañana-mixto
(PLAB) Prácticas de laboratorio	421	Catalán/Español	primer cuatrimestre	mañana-mixto
(PLAB) Prácticas de laboratorio	422	Catalán/Español	primer cuatrimestre	mañana-mixto
(PLAB) Prácticas de laboratorio	423	Catalán/Español	primer cuatrimestre	tarde
(TE) Teoría	41	Catalán	primer cuatrimestre	mañana-mixto
(TE) Teoría	43	Catalán	primer cuatrimestre	mañana-mixto
(TE) Teoría	45	Catalán/Español	primer cuatrimestre	tarde