

# Programación 1

---

Grado en Ingeniería Informática — Curso 2019-20

<https://webdiis.unizar.es/asignaturas/PROG1/>

<https://moodle.unizar.es/add/course/view.php?id=29833>



Escuela de  
Ingeniería y Arquitectura  
**Universidad Zaragoza**

# Profesores de la asignatura

---

- ❑ **Miguel Ángel Latre** [latre@unizar.es](mailto:latre@unizar.es)
  - Profesor responsable de la asignatura
- ❑ **Simona Bernardi** [simonab@unizar.es](mailto:simonab@unizar.es)
- ❑ **José Neira** [jneira@unizar.es](mailto:jneira@unizar.es)
- ❑ **Ricardo J. Rodríguez** [rjrodriguez@unizar.es](mailto:rjrodriguez@unizar.es)
- ❑ **Prof. por contratar**
  
- ❑ Área de Lenguajes y Sistemas Informáticos
- ❑ Departamento de Informática e Ingeniería de Sistemas



# Programación 1

---

- Asignatura de formación básica (obligatoria)
  - 6,0 créditos ECTS
  - Primer paso en el aprendizaje de la programación de computadores (se parte de cero)
  - Tiene **continuidad** en asignaturas posteriores de la materia común de Computación y Programación
  - **Imprescindible** para estudiar prácticamente cualquier otra materia de las que intervienen en los estudios



# Objetivos

---

- ❑ Conocer los **conceptos básicos** ligados a la programación
- ❑ Saber **analizar problemas** concretos, **plantear soluciones** y **desarrollar programas** que permitan resolverlos en un computador
- ❑ Aprender a utilizar y familiarizarse con un **entorno tecnológico** específico:
  - Lenguaje **C++** y entorno integrado **CodeLite**
- ❑ Ejercitarse como **programador**

# Resultados de aprendizaje

---

Esta asignatura acredita que quien la apruebe:

1. Conoce **conceptos básicos** ligados a la programación y la informática.
2. Comprende, analiza y resuelve **problemas de tratamiento de información de complejidad baja o media** y construye algoritmos que los resuelven.
3. Define las **estructuras de datos** más adecuadas para representar la información asociada a cada problema.
4. Diseña de forma **descendente** y documenta las **acciones** algorítmicas que resuelven cada problema de forma eficaz y eficiente.
5. Conoce algoritmos para resolver los **problemas más frecuentes** que se presentan al trabajar con **estructuras de datos secuenciales e indexadas**.

# Resultados de aprendizaje

---

Esta asignatura acredita que quien la apruebe (cont.):

6. Conoce y comprende la **sintaxis** y la **semántica** de las construcciones básicas de un **lenguaje de programación: C++**
7. Escribe programas con **buen estilo**, con una **documentación** adecuada, con los comentarios precisos y con las **especificaciones** necesarias.
8. Sabe utilizar **herramientas de edición, compilación, depuración y ejecución** para desarrollar programas, así como sistemas operativos y otros programas con aplicación en ingeniería.
9. Utiliza **estrategias para corregir los programas** cuando no funcionan bien.
10. Implementa y ejecuta en un computador programas escritos en un lenguaje de programación determinado: **C++**

# Programa

---

## **I. Conceptos y elementos básicos de Programación**

- Problemas de tratamiento de información, algoritmos y programas
- Lenguajes de programación y ejecución de un programa
- Información, datos, operaciones y expresiones

## **II. Diseño de los primeros programas**

- Diseño de algunos programas elementales
- Instrucciones simples y estructuradas
- Problemas de cálculos con enteros
- Desarrollo modular y descendente de programas
- Problemas de cálculo con números reales

# Programa

---

## **III. Diseño de programas que trabajan con estructuras de datos**

- Vectores
- Cadenas de caracteres
- Registros
- Algoritmos básicos de trabajo con estructuras de datos indexadas

## **IV. Diseño de programas que trabajan con ficheros**

- Entrada y salida de datos
- Ficheros de texto
- Ficheros binarios
- Ficheros: otras posibilidades



# Metodología de aprendizaje

---

- Clases magistrales
- Clases de problemas
- Prácticas en laboratorio
- Trabajo obligatorio
- Seguimiento de trabajos y prácticas
  - Práctica TP6
  - Tutorías



# Grupos de prácticas y problemas (grupo de mañanas)

---

- Nuevos estudiantes:
  - Grupo 1: Apellido entre [**AAA-CAT**]
  - Grupo 2: Apellido entre [**CAU-ESTER**]
  - Grupo 3: Apellido entre [**ESTES-GIMENE**]
  - Grupo 4: Apellido entre [**GIMENF-ZZZ**]
- Resto de estudiantes:
  - Confirmar con el profesor la adscripción a un grupo



# Grupos de prácticas y problemas (grupo de tardes)

---

- Nuevos estudiantes grado Informática:
  - Grupo 1: Apellido entre [**AAA-LOQ**]
  - Grupo 2: Apellido entre [**LOR-MTZ**]
  - Grupo 3: Apellido entre [**MUA-RÑZ**]
  - Grupo 4: Apellido entre [**ROA-ZZZ**]
- Estudiantes doble grado:
  - Grupo 1: Apellido entre [**AAA-PAE**]
  - Grupo 2: Apellido entre [**PAF -ZZZ**]
- Resto de estudiantes:
  - Confirmar con el profesor la adscripción a un grupo



# Clases magistrales

---

- Aprendizaje de conceptos y metodologías de diseño de programas
  - 2 horas semanales
  - Asistencia voluntaria
- Aula 18 del edificio Torres Quevedo
  - Grupo de mañanas
    - Lunes y viernes 12:10 a 13:00



# Clases magistrales

---

- Aprendizaje de conceptos y metodologías de diseño de programas
  - 2 horas semanales
  - Asistencia voluntaria
- Aula 18 del edificio Torres Quevedo
  - Grupo de tardes
    - Martes de 17:10 a 18:00
    - Viernes de 16:00 a 16:50



# Clases de problemas

---

- Aplicación de conceptos y metodologías al diseño de algoritmos y programas
  - 1 hora semanal, los jueves
  - Asistencia voluntaria
- Grupo de mañanas  
Profesora: Simona Bernardi
  - Grupos 3 y 4: 12:10 a 13:00 (aula 19)
  - Grupos 1 y 2: 13:10 a 14:00 (aula 18)



# Clases de problemas

---

- Aplicación de conceptos y metodologías al diseño de algoritmos y programas
  - 1 hora semanal, los jueves
  - Asistencia voluntaria
- Grupo de tardes  
Profesor: Miguel Ángel Latre
  - Grupos 3 y 4: 18:10 a 19:00 (aula 14)
  - Grupos 1 y 2: 19:10 a 20:00 (aula 18)



# Clases prácticas de laboratorio

---

- ❑ Tecnología necesaria para programar, ejecutar y depurar utilizando C++ y en un entorno concreto
- ❑ 2 horas cada dos semanas
- ❑ Asistencia voluntaria





# Clases prácticas de laboratorio

---

## □ Grupo de mañanas

Horario: 10:00–12:00

Laboratorio L.0.04 del edificio Ada Byron

■ Grupo 1: lunes A

Profesor: Ricardo J. Rodríguez

■ Grupo 2: lunes B

Profesor: José Neira

■ Grupo 3: martes A

Profesor: Miguel Ángel Latre

■ Grupo 4: martes B

Profesor: Miguel Ángel Latre



# Clases prácticas de laboratorio

---

## □ Grupo de tardes

Horario: 15:00–17:00

Laboratorio L.0.04 del edificio Ada Byron

■ Grupo 1: martes A

Profesor: José Lloret

■ Grupo 2: martes B

Profesor: José Lloret

■ Grupo 3: lunes A

Profesor: Ricardo J. Rodríguez

■ Grupo 4: lunes B

Profesor: Miguel Ángel Latre



# Clases prácticas de laboratorio

---

- **Puestos de trabajo del laboratorio L.0.04**
  - Equipos con sistema operativo Linux (CENT OS), **Windows**, ...
  - **Clúster** (hendrix) hace funciones de **servidor de ficheros**
  - **Cuenta** para el acceso a los equipos y al clúster
    - Se os informará de su nombre de usuario y contraseña en la primera sesión de prácticas
- **¿Dónde desarrollar programas C++?**
  - Trabajando **desde cualquier puesto del L.0.04** (o de otro laboratorio del DIIS) con cualquier entorno de desarrollo (editor de texto + compilador, **CodeLite**, Code::Blocks, ...)
  - Trabajando **desde tu computador personal** con cualquier entorno de desarrollo (editor de texto + compilador, **CodeLite**, Code::Blocks, ...)



# Clases prácticas de laboratorio

---

- Prácticas **individuales**
  - Es positivo intercambiar experiencias y resolver dudas con compañeros
- **Preparación** de las prácticas (antes de la sesión):
  - Lectura completa de los guiones de prácticas
  - Resolución de los problemas en la medida de lo posible
  - Identificación de dudas
- **Durante las sesiones** de prácticas:
  - Consulta de dudas al profesor
  - Supervisión del trabajo por el profesor
  - Finalización del trabajo de prácticas



# Prácticas

---

- Las prácticas son de entrega voluntaria
  - En el enunciado de cada práctica se indicará qué hay que entregar en concreto
- Plazo de entrega genérico: sábados de las semanas B antes de las 18:00, a través de Moodle
  - En el enunciado de cada práctica se indicará la fecha en concreto
- Algunas de las entregas serán corregidas por los profesores
- Un 10% de la calificación obtenida con los trabajos corregidos se sumará a la calificación obtenida en la convocatoria de febrero en el caso de haber aprobado la asignatura.
  - No se aplica a quienes no aprueben la asignatura en febrero y no se conserva para la convocatoria de septiembre.



# Trabajo obligatorio y seguimiento de trabajos y prácticas

---

- Trabajo de programación
  - Se publicará en la primera quincena de diciembre
  - Entrega en enero, al finalizar el periodo de clases (en torno al 15 de enero)
- Seguimiento de trabajos y prácticas
  - Prácticas TP6
  - En horarios de tutorías específicos de los profesores José Neira y Miguel Ángel Latre



# Tutorías

---

- Tutorías académicas
  - Supervisión del trabajo de los alumnos, orientación, resolución de dudas, recomendación de bibliografía, revisión de trabajos y pruebas, etc., dentro del ámbito de la asignatura



# Tutorías

---

- ❑ **Miguel Ángel Latre**
- ❑ Lugar
  - Despacho D.2.22, edificio Ada Byron
- ❑ Horario
  - Martes: 12:00 a 13:00
  - Jueves: 11:30 a 13:30 y 15:00 a 18:00
  - <http://webdiis.unizar.es/~latre/tutorias>
  - Las tutorías deben reservarse con antelación a través del siguiente calendario Google: <https://goo.gl/w94D5J>





# Tutorías

---

- ❑ **Simona Bernardi**
- ❑ Lugar
  - Despacho D.2.12, edificio Ada Byron
- ❑ Horario
  - Martes            12:00 a 14:00
  - Miércoles        12:00 a 14:00
  - Viernes           12:00 a 14:00



# Tutorías

---

- **José Neira**
- Lugar
  - Despacho D.1.19, edificio Ada Byron
- Horario
  - Martes 11:00 a 13:00
  - Jueves 16:00 a 18:00
  - Viernes 9:00 a 11:00



# Tutorías

---

- ❑ **Ricardo J. Rodríguez**
- ❑ Lugar
  - Despacho D.0.08, edificio Ada Byron
- ❑ Horario
  - Martes y Jueves, de 15.00 a 17.00
  - Miércoles, de 11.00 a 13.00



# Localización

Despacho  
D.2.22, edificio  
Ada Byron

Laboratorio  
L.0.04, edificio  
Ada Byron

Aula 19, edificio  
Torres Quevedo

Aula 18, edificio  
Torres Quevedo

Parada tranvía  
«Campus Río Ebro»

Aula 14, edificio  
Torres Quevedo

Mapa: 675888.72, 4617018.2 (UTM30N, EDE0) - Escala 1:2625

Datos cartográficos base suministrados por el Servicio de Información Geográfica del Ayuntamiento de Zaragoza. Otras fuentes de datos puntuales: [OpenStreetMap](#), [CC-BY-SA](#)

# Carga de trabajo

---

- Estimación de **150 horas efectivas** de trabajo:
  - 56 horas de **actividades presenciales activas**
    - clases «teóricas»
    - clases de problemas
    - prácticas en laboratorio
  - 91 horas de **estudio personal efectivo**
    - estudio de apuntes y textos
    - resolución de problemas
    - preparación clases y prácticas
    - desarrollo de programas
  - 3 horas de **examen final escrito**



# Proceso de aprendizaje

---

- El aprendizaje de la programación exige un **trabajo continuado desde el primer día de clase:**
  - **comprensión** de conceptos,
  - **análisis y la resolución de problemas** utilizando lápiz y papel,
  - **puesta a punto en computador** de un buen número de programas.
- *“A programar se aprende programando”*





# ¿Qué hacer para aprender?

---

- Asistir a clase
  - Habiendo realizado el trabajo previo que se haya encargado
  - Atendiendo y participando activamente en la clase
  - Repasando después y comprendiendo cada lección
- Resolver los problemas de programación propuestos en las clases de problemas y en las prácticas:
  - Lápiz y papel
  - Programación en computador
  - Validación del código desarrollado
- Colaborar con otros compañeros y consultarles
- Consultar dudas a los profesores en sus horarios de tutorías



# Actividades de evaluación

---

- **P1 Prueba escrita**
  - Nota mínima: 4,0
  - Ponderación: 70%
- **P2 Trabajos y prueba de programación en laboratorio.**
  - Ponderación: 30%
  - Febrero
    - Trabajo obligatorio. Ponderación 15%
    - Prueba de programación en laboratorio. Ponderación 15%
  - Septiembre
    - Prueba de programación en laboratorio. Ponderación 30%
- **Prácticas**
  - Entrega voluntaria
  - Permiten subir la nota, hasta 1 punto, a quienes aprueben en febrero





# Actividades de evaluación

---

- Las calificaciones obtenidas en febrero en las pruebas P1 y P2 se mantienen en septiembre, salvo que se opte por presentarse a la prueba correspondiente en septiembre, en cuyo caso prevalecerá la nueva calificación



# Bibliografía básica

---

- **Javier Martínez:** *Curso de Programación 1*. 2017
- **Miguel Ángel Latre y Javier Martínez:** *Prácticas de Programación 1*. Moodle. 2019
- **Miguel Ángel Latre y Javier Martínez:** *Diversos materiales docentes*.
  - Web de la asignatura:  
<http://webdiis.unizar.es/asignaturas/PROG1>
  - Curso en Moodle:  
<https://moodle.unizar.es/add/course/view.php?id=29833>



# Bibliografía de consulta

---

- Páginas web con documentación sobre el lenguaje C++
  - <http://www.cplusplus.com/>
  - <http://www.cprogramming.com/>
  - [http://es.wikibooks.org/wiki/Programación en C++](http://es.wikibooks.org/wiki/Programaci3n_en_C%2B%2B)
- Manual en línea de bibliotecas predefinidas en C++
  - <http://www.cplusplus.com/reference/>
- Entorno de ejecución en línea
  - <http://cpp.sh/>
- Hay muchos textos para apoyar la enseñanza de un primer curso de programación utilizando el lenguaje C++



# Web de la asignatura

<https://webdiis.unizar.es/asignaturas/PROG1/>

## Información

[La asignatura](#)

[Programa](#)

[Horarios y organización de clases y sesiones prácticas](#)

[Profesores](#)

[Evaluación de la asignatura](#)

[Sistema de evaluación](#)

[Convocatorias de examen](#)

[Calificaciones](#)

[Estadísticas](#)

[Material docente](#)

[Código C++ y datos](#)

[Transparencias](#)

[Programar en C++](#)

[Documentación C++](#)

[Restricciones en el uso de recursos del lenguaje C++](#)

[Entorno de programación en línea](#)

<https://moodle.unizar.es/add/course/view.php?id=29833>



Escuela de  
Ingeniería y Arquitectura  
Universidad Zaragoza

# Para la clase del jueves...

---

- Notación Backus-Naur o BNF
  - Lectura de la sección 2.1.2 de los apuntes del Profesor Martínez
  - Disponible en Moodle



# Para la clase del martes/viernes...

---

- Problemas de **tratamiento de información**
  - Objetivo: resolución **automática** del problema
  - ¿Quién? Un computador
  - Necesidad de **programarlo**



# Algoritmo

---

- Conjunto de operaciones

- ordenado,
- finito,
- carente de ambigüedades,

que permite hallar la solución de un problema de tratamiento de información



# Para la clase del martes/viernes...

---

- ❑ Resolución de un problema ~~de tratamiento de información~~
- ❑ Hacer una tortilla de patata



File:Tortilla de Patatas (Corte transversal).jpg. (9 de marzo de 2017).  
*Wikimedia Commons, the free media repository.* Accedido el 18 de septiembre de 2017.  
[https://commons.wikimedia.org/w/index.php?title=File:Tortilla\\_de\\_Patatas\\_\(Corte\\_transversal\).jpg&oldid=236535592](https://commons.wikimedia.org/w/index.php?title=File:Tortilla_de_Patatas_(Corte_transversal).jpg&oldid=236535592).



Escuela de  
Ingeniería y Arquitectura  
**Universidad Zaragoza**