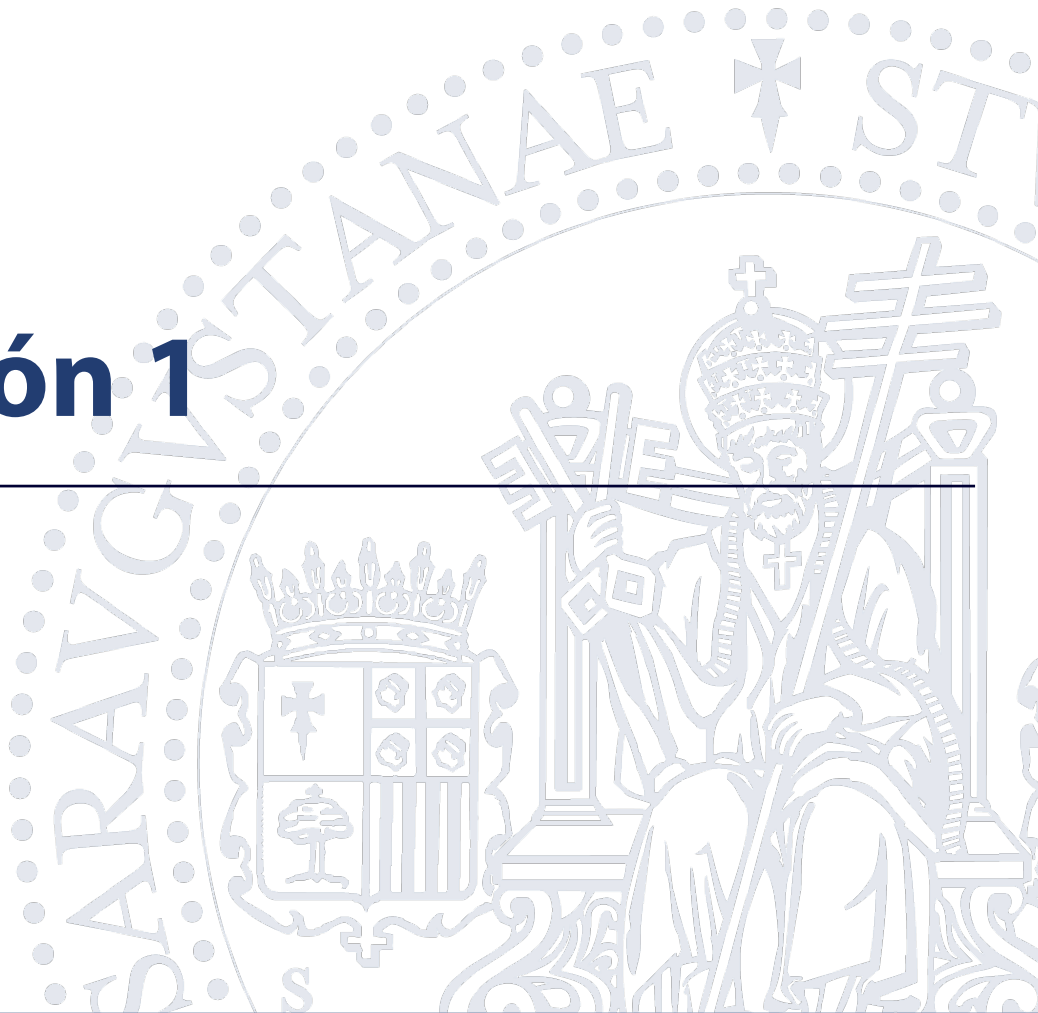


Programación 1

Grado en Ingeniería Informática



Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza



Profesorado de la asignatura

- Miguel Ángel Latre
 - latre@unizar.es
 - Responsable de la asignatura
 - Teoría (411)
 - 2 grupos de problemas (412)
 - Varios grupos de prácticas (411, 412 y 413)
- Departamento de Informática e Ingeniería de Sistemas
 - Área de Lenguajes y Sistemas Informáticos



Profesorado de la asignatura

- Ricardo J. Rodríguez
 - rjrodriguez@unizar.es
 - Teoría (412)
 - Problemas (413)
- Departamento de Informática e Ingeniería de Sistemas
 - Área de Lenguajes y Sistemas Informáticos



Profesorado de la asignatura

- Ignacio Huitzil
 - ihuitzil@unizar.es
 - Teoría (413)
 - Varios grupos de prácticas (411, 412)
- Departamento de Informática e Ingeniería de Sistemas
 - Área de Lenguajes y Sistemas Informáticos



Profesorado de la asignatura

- José Merseguer
 - jmerse@unizar.es
 - 2 grupos de problemas (411)
 - 1 grupo de problemas (412)
 - 1 grupo de prácticas (411)
- Departamento de Informática e Ingeniería de Sistemas
 - Área de Lenguajes y Sistemas Informáticos



Profesorado de la asignatura

- Javier Zarazaga
 - javy@unizar.es
 - 2 grupos de prácticas (412, 413)
- Departamento de Informática e Ingeniería de Sistemas
 - Área de Lenguajes y Sistemas Informáticos



Profesorado de la asignatura

- Raquel Trillo
 - raqueltl@unizar.es
 - 1 grupo de prácticas (411)
- Departamento de Informática e Ingeniería de Sistemas
 - Área de Lenguajes y Sistemas Informáticos



Profesorado de la asignatura

- Ana Cambra
 - acambra@unizar.es
 - 2 grupos de prácticas (413)
- Departamento de Informática e Ingeniería de Sistemas
 - Área de Lenguajes y Sistemas Informáticos



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)
 - **Continuidad** en asignaturas posteriores
 - **Imprescindible** para estudiar prácticamente cualquier otra materia de las que intervienen en los estudios



Planteamiento

- ❑ **Diseño de programas** que resuelvan **problemas de tratamiento de información**
- ❑ Presentación de herramientas para **representar información** y para **tratarla**
- ❑ Planteamiento de **metodologías** para facilitar y sistematizar el **análisis de problemas** y el **diseño de programas** que los resuelvan
- ❑ Más información: *Guía docente curso 2024–2025 de Programación 1*
 - https://estudios.unizar.es/estudio/asignatura?anyo_academico=2024&asignatura_id=30204&estudio_id=20240148¢ro_id=110&plan_id_nk=439



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 **Visual Studio Code**
- ❑ Ejercitarse como **programador**

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 vectores

IV. Diseño de programas que trabajan con ficheros

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

V. Metodología de diseño de programas

Metodología de aprendizaje

- ❑ Clases de teoría
- ❑ Clases de problemas
- ❑ Clases prácticas



Clases de teoría

- Aprendizaje de conceptos y metodologías de diseño de programas
 - 2 horas semanales
 - Asistencia voluntaria



Clases de teoría

- Grupo 411
 - Aula 17 TQ
 - Miércoles y viernes 9:00 a 9:50
- Grupo 412
 - Aula 17 TQ
 - Miércoles y viernes de 16:00 a 16:50
- Grupo 412
 - Aula 17 TQ
 - Miércoles de 18:00 a 18:50
 - Viernes de 14:00 a 14:50



Clases de problemas

- Aplicación de conceptos y metodologías al diseño de algoritmos y programas
 - 1 hora semanal
 - Con grupos desdoblados en 2 o 3 subrupos, en coordinación con IC
 - Asistencia voluntaria



Clases de problemas

Grupo 411

Grupo	Horario	Aula	Profesor	Inicial 1.º apellido
4112	Jueves 8:00 a 8:50	17 TQ	José Merseguer	(por determinar)
4111	Jueves 9:00 a 9:50	24 TQ	José Merseguer	(por determinar)



Clases de problemas

Grupo 412

Grupo	Horario	Aula	Profesor	Inicial 1.º apellido
4123	Miércoles 15:00 a 15:50	25 TQ	José Merseguer	(por determinar)
4122	Jueves 17:00 a 17:50	24 TQ	Miguel Latre	(por determinar)
4121	Jueves 18:00 a 18:50	17 TQ	Miguel Latre	(por determinar)



Clases de problemas

Grupo 413

Grupo	Horario	Aula	Profesor	Inicial 1.º apellido
4131	Miércoles 14:00 a 14:50	25 TQ	Ricardo J. Rodríguez	(por determinar)
4132	Jueves 14:00 a 14:50	25 TQ	Ricardo J. Rodríguez	(por determinar)



Clases prácticas

- ❑ Tecnología necesaria para programar, ejecutar y depurar utilizando
 - Un lenguaje concreto: C++
 - Un entorno concreto: Visual Studio Code
- ❑ 2 horas cada dos semanas
- ❑ **Asistencia voluntaria**
- ❑ Individuales



Grupos de prácticas

Grupo 411

- 5 subgrupos de prácticas
 - Laboratorio L.0.04, edificio Ada Byron

Subgrupos	Horario	Profesores
3;4	Lunes A de 8:00 a 9:50	Raquel Trillo / Miguel Latre
1;2	Martes A de 10:00 a 11:50	José Merseguer / Miguel Latre
5	Lunes B de 8:00 a 9:50	Ignacio Huitzil



Grupos de prácticas

Grupo 412

- 5 subgrupos de prácticas
 - Laboratorio L.0.04, edificio Ada Byron

Subgrupos	Horario	Profesores
21;22	Martes A de 15:00 a 16:50	Ignacio Huitzil / Miguel Latre
24;25	Lunes B de 15:00 a 16:50	Javier Zarazaga / Miguel Latre
23	Martes B de 15:00 a 16:50	Miguel Latre



Grupos de prácticas

Grupo 413

- 4 subgrupos de prácticas

Subgrupos	Horario	Laboratorio	Profesores
31	Martes A de 17:00 a 18:50	L0.06b, edificio Ada Byron	Ana Cambra
32	Martes B de 17:00 a 18:50	L0.06b, edificio Ada Byron	Ana Cambra
33;34	Viernes A de 15:00 a 16:50	L0.01, edificio Ada Byron	Javier Zarazaga / Miguel Latre



Localización

Laboratorio
L.0.04, edificio
Ada Byron

Parada tranvía
«Campus Río Ebro»

Aula 15, edificio
Torres Quevedo

Aula 17, edificio
Torres Quevedo

Fuente: SIGEUZ 3.1. Universidad de Zaragoza (<https://sigeuz.unizar.es/>)



Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza

Clases prácticas

- Prácticas **individuales**
- **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 **asistencia voluntaria**.
- Las prácticas son de **entrega obligatoria**.
 - 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.



Prácticas

- ❑ Algunas de las entregas podrán serán corregidas por pares (por otros estudiantes).
- ❑ Algunas de las entregas o de las correcciones por pares serán revisadas también por los profesores.
- ❑ Un 30 % de la calificación obtenida con los trabajos que se corrijan forma parte de la calificación de la convocatoria de febrero.



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
 - En general, de forma presencial o telemática (preguntad al profesor).
- Horarios y mecanismos de reserva publicados en Moodle



Sobre el uso del correo electrónico

- Medio de comunicación **asíncrono**
 - El emisor manda el mensaje en un determinado momento.
 - El receptor lo lee en un momento posterior.
 - El receptor puede contestar, si lo estima pertinente, todavía más tarde.



Carga de trabajo

- Estimación de **150 horas efectivas** de trabajo:
 - 57 horas de **actividades presenciales activas**
 - clases «teóricas»
 - clases de problemas
 - prácticas
 - 90 horas de **estudio personal efectivo**
 - estudio de apuntes y textos
 - resolución de problemas
 - preparación clases y prácticas
 - trabajo obligatorio
 - desarrollo de programas
 - 3 horas de pruebas de evaluación



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. Examen escrito**
 - Nota mínima: 5,0
 - Ponderación: 70 %
- **Parte práctica**
 - Ponderación: 30 %
 - **P2: Evaluación continua**
 - Solo en convocatoria ordinaria
 - Entrega de prácticas.
 - **P3: Examen global de prácticas**
 - En convocatoria ordinaria o extraordinaria
 - Examen de programación en laboratorio
- **Pruebas voluntarias**
 - Entrega de problemas voluntarios junto con las prácticas
 - 10 % adicional **solo en convocatoria ordinaria** a la calificación final de cada estudiante que **haya aprobado** con las calificaciones P1+P2 o P1+P3.

Actividades de evaluación

- Las calificaciones obtenidas en convocatoria ordinaria en la prueba P2 (parte práctica por evaluación continua) se mantienen en la convocatoria extraordinaria como prueba P3 (examen global de prácticas), salvo que se opte por presentarse a la prueba P3 en la convocatoria extraordinaria, en cuyo caso prevalecerá la nueva calificación.



Actividades de evaluación

- Fecha examen escrito y global de prácticas
 - Convocatoria ordinaria: 8-1-2024, mañana
 - Convocatoria extraordinaria: 23-6-2024, mañana



Bibliografía básica

- **Javier Martínez:** *Curso de Programación 1*. 2017
- **Profesores de la asignatura y Javier Martínez:** *Prácticas de Programación 1*. Moodle. 2024
- **Profesores de la asignatura, Javier Martínez y otros:** *Diversos materiales docentes.*
 - Curso en Moodle:
<https://moodle.unizar.es/add/course/view.php?id=109137>



Bibliografía de consulta

- Páginas web con documentación sobre el lenguaje C++
 - <https://www.cplusplus.com/>
 - <https://www.cprogramming.com/>
 - https://es.wikibooks.org/wiki/Programación_en_C++
- Manual en línea de bibliotecas predefinidas en C++
 - <https://www.cplusplus.com/reference/>
- Entorno de ejecución en línea
 - <https://cpp.sh/>
- Entorno de visualización de la ejecución en línea
 - <https://pythontutor.com/cpp.html>
- Hay muchos textos para apoyar la enseñanza de un primer curso de programación utilizando el lenguaje C++.



Curso en Moodle

- <https://moodle.unizar.es/add/course/view.php?id=109137>



Curso en Moodle

- ❑ Presentación de la asignatura y guía docente
- ❑ Horarios de tutorías
- ❑ Planificación, transparencias del curso y enunciados de problemas
- ❑ Acceso a todo el código fuente
 - Ejemplos de teoría
 - Soluciones a los problemas
 - Código base para las prácticas
- ❑ Material adicional por temas
 - Capítulos de los apuntes del profesor Javier Martínez
 - Enlaces a tutoriales de Cplusplus.com
- ❑ Enunciados de prácticas
- ❑ Material sobre C++ y Visual Studio Code
- ❑ Ejercicios básicos
- ❑ Exámenes resueltos



Para la próxima clase...

- ❑ Buscad y leed atentamente una receta para hacer 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).



Para la primera práctica...

- Podéis instalaros ya Visual Studio Code
 - Instrucciones en Moodle
 - «[Tutorial para la instalación de Visual Studio Code](#)»
(sección Programar en C++)

