

Presentación de la asignatura

Programación 2

Curso 2017-18

Información en:

<http://webdiis.unizar.es/asignaturas/PROG2/>

Índice:

1. ¿Qué vamos a aprender en esta asignatura?
2. Algunas preguntas
3. ¿Qué temas vamos a estudiar?
4. ¿Cómo van a ser las clases y las prácticas?
5. ¿Cómo estudiar y trabajar la asignatura?
6. El equipo de profesores
7. ¿Cómo va a ser la evaluación de la asignatura?
8. ¿De qué materiales vamos a disponer?

1. ¿Qué vamos a aprender en esta asignatura?

- En **PROGRAMACIÓN 1** hemos aprendido a **resolver problemas de tratamiento** de información **analizándolos** y **desarrollando programas** aplicando una determinada metodología de diseño
- En **PROGRAMACIÓN 2** vamos a aprender a ...

1. ¿Qué vamos a aprender en esta asignatura?

- En **PROGRAMACIÓN 1** hemos aprendido a **resolver problemas de tratamiento** de información **analizándolos** y **desarrollando programas** aplicando una determinada metodología de diseño
- En **PROGRAMACIÓN 2** vamos a aprender a **programar mejor**, sabiendo:
 - **qué queremos** exactamente programar,
 - **por qué el código** que estamos escribiendo **es correcto**
 - y **cuál va a ser el coste**, en tiempo y en memoria, necesario para ejecutarlo

Objetivo general	Objetivo desglosado	Qué debemos aprender	Dónde aprenderlo	Prácticas y problemas
Aprender a programar mejor	Saber lo que queremos programar	Especificar algoritmos formalmente	Tema I	Problemas
	Saber cómo programar correctamente (con/sin bucles)	Un poco más de tecnología C++	Prácticas	Práctica 1ª
		Diseñar algoritmos correctos con bucles (iterativos) y recursivos	Tema II y Tema V	Prácticas 2ª, 3ª y 5ª Problemas
	Comprender lo que estamos programando	Analizar formalmente la corrección de algoritmos	Tema IV	Problemas
	Saber cuánto cuesta ejecutar (en tiempo y en memoria) lo que hemos programado	Analizar costes de algoritmos (en tiempo y en memoria)	Tema III	Práctica 4ª Problemas

2. Algunas preguntas:

¿Qué pasa si aún no he aprobado PROG 1?

¿Es PROG 2 más fácil o más difícil que PROG 1?

¿Qué relación hay entre PROG 1 y PROG 2?

¿Hay más asignaturas de Programación?

Algunas preguntas:

¿Qué pasa si aún no has aprobado PROG 1?

- **Nada insuperable. Pero hay que trabajar mucho más**

¿Es PROG 2 más fácil o más difícil que PROG 1?

¿Qué relación hay entre PROG 1 y PROG 2?

¿Hay más asignaturas de Programación?

Algunas preguntas:

¿Qué pasa si aún no has aprobado PROG 1?

- Nada insuperable. Pero hay que trabajar mucho más

¿Es PROG 2 más fácil o más difícil que PROG 1?

- **Suele resultar MUCHO más DIFÍCIL. Y si vas flojo en matemáticas, debes repasar las de bachiller y Matemáticas 1 y estudiar a fondo Matemática Discreta**

¿Qué relación hay entre PROG 1 y PROG 2?

¿Hay más asignaturas de Programación?

Algunas preguntas:

¿Qué pasa si aún no has aprobado PROG 1?

- Nada insuperable. Pero hay que trabajar mucho más

¿Es PROG 2 más fácil o más difícil que PROG 1?

- Suele resultar MUCHO más DIFÍCIL. Y si vas flojo en matemáticas, debes repasar las de bachiller y Matemáticas 1 y estudiar a fondo Matemática Discreta

¿Qué relación hay entre Prog 1 y Prog 2?

- **PROG1 (comprensión de una tecnología y una metodología de programación) y PROG2 (profundización en ellas aplicando conocimiento científico – matemáticas – para programar mejor)**

¿Hay más asignaturas de Programación?

Algunas preguntas:

¿Qué pasa si aún no has aprobado Prog 1?

- Nada insuperable. Pero hay que trabajar mucho más

¿Es Prog 2 más fácil o más difícil que Prog 1?

- Suele resultar MUCHO más DIFÍCIL. Y si vas flojo en matemáticas, debes repasar las de bachiller y Mat. 1 y estudiar a fondo Matemática Discreta

¿Qué relación hay entre Prog 1 y Prog 2?

- Prog1 (comprensión de una tecnología y una metodología de programación) y Prog2 (profundización en ellas aplicando conocimiento científico –matemáticas – para programar mejor)

¿Hay más asignaturas de Programación?

- **PROG1 (c1º) → PROG2 (c2º) → EDA (c3º) → PSCD (c3º)**
→ TecPROG (c4º) → IngSOFT (c5º) → ProySOFT (c6º) → ...

3. ¿Qué temas vamos a estudiar?

1. Especificación formal de algoritmos
 - **Escritura de predicados lógicos**
 2. Diseño recursivo de algoritmos
 - **Escritura de predicados lógicos. Metodología de diseño sin bucles**
 3. Análisis del coste de algoritmos
 - **Funciones matemáticas. Suma de costes. Suma de series. Resolución de recurrencias**
 4. Análisis de la corrección
 - **Escritura de predicados lógicos. Cálculo de predicados lógicos. Demostraciones matemáticas elementales**
 5. Diseño de algoritmos correctos
 - **Escritura de predicados lógicos. Cálculo de predicados lógicos. Demostraciones matemáticas elementales. Metodología de diseño con bucles**
- Además en las prácticas: algunas novedades sobre tecnología C++ y sobre herramientas de desarrollo de programas

4. ¿Cómo van a ser las clases y las prácticas?

Las clases de los jueves (18:10-20:00)

- Antes de acudir a estas clases conviene **haber leído** la lección que corresponda para ese día.
- En ellas trataré de facilitar la **comprensión de las ideas esenciales** de cada tema y que se **aprenda el modo de aplicarlas**, así como de **aclarar las dudas** que se os hayan planteado en su lectura previa o en la propia clase.
- No me voy a dedicar a “**contar**” los apuntes ya que todos sabemos leer.
- Es esencial **comprender** lo que se explica en clase. Para ello hay que esforzarse, poner los medios y preguntar cuando sea preciso.
- A clase **acudimos todos a trabajar**. Es necesaria una actitud **activa**. Conviene evitar actitudes **pasivas, oyentes o durmientes**.
- Máxima **interacción**: habrá preguntas y respuestas en ambos sentidos.

Las clases de los viernes (16:00-16:50 ó 18:10-19:00)

- Su objetivo es que aprendamos a **resolver problemas**. ¿Cómo? Muy sencillo: **resolviendo problemas**
- Normalmente, dispondréis de sus **enunciados** con suficiente antelación, en la web de PROG2
- Normalmente, los problemas **los resolveréis vosotros** y luego **discutiremos vuestras soluciones** en clase
- Al **estudiar** la asignatura por vuestra cuenta debéis **resolver muchos problemas**, desde la primera semana. Para ello disponéis de los **enunciados de los *problemas* de cada tema** y de los **problemas resueltos** propuestos en los exámenes escritos de cursos pasados.
- En esta asignatura es muy importante **asistir a todas sus clases**; también a las de los viernes

Las prácticas

- **Primera:** aprenderemos a trabajar en una máquina con **sistema operativo unix** o **linux** y aprenderemos a usar algunas herramientas nuevas de C++
- **Segunda y tercera** (doble): aprenderemos a diseñar **programas sin bucles**, aplicando la metodología de **diseño recursivo** aprendida en el **Tema II**
- **Cuarta:** **analizaremos el coste**, en tiempo, de algunos algoritmos
- **Quinta:** aplicaremos algunos **algoritmos de ordenación** de vectores a la ordenación de matrices y lo haremos mediante un **diseño sin bucles**

Muy importante:

- Las prácticas están pensadas como un **trabajo a realizar a lo largo de dos semanas** (excepto la tercera, que será de **seis semanas**). Este trabajo debe comenzar mucho antes de acudir al laboratorio a la sesión de prácticas
- En las sesiones de prácticas hay un **profesor** que te va a ayudar a **resolver las dudas** que tengas y va a supervisar tu trabajo. Las sesiones de prácticas deben servir para **finalizar** los trabajos asociados a cada práctica y no para empezarlos
- El trabajo pedido en cada práctica **lo debes hacer y conservar** ya que te será **útil y**, quizás, **necesario** más adelante.

Guía de estilo para programar en C++:

- Todos los programas C++ o fragmentos de programas que se escriban en esta asignatura (en prácticas, en trabajos y en los exámenes escrito y práctico) deberán respetar las normas recogidas en la **Guía de estilo para programar en C++** que puede consultarse y descargarse desde la web de la asignatura. Estas normas constituyen un buen resumen de las buenas prácticas que han debido ser aprendidas en ***Programación 1***
- Esta misma guía de estilo se va a seguir en varias de las asignaturas de los estudios: ***Programación 1, Programación 2, Estructuras de Datos y Algoritmos***, etc.

5. ¿Cómo estudiar y trabajar la asignatura?

Qué hacer en las clases impartidas en el aula:

- **Comprender** las ideas que cada lección presenta
- **Preguntar** lo que no entiendas
- **Comprender** los ejercicios y problemas que se explican
- Esforzarte por **resolver** los problemas que se proponen en clase

Qué hacer después de las clases:

- Verificar si los conceptos se comprenden y se domina su aplicación:
 - **Repasar los apuntes** de la asignatura y, en su caso, **consultar bibliografía** complementaria
 - Volver a **resolver** los problemas de los apuntes y los resueltos en clase
 - **Resolver problemas** propuestos e **inventar nuevos problemas y resolverlos**
 - **Consultar a los profesores** en sus tutorías

Sobre las prácticas con computador:

- **En qué máquinas** programar: en los computadores de la Universidad (*PCs* de los laboratorios y clúster *hendrix-ssh*) o en los computadores que disponga cada cual (conexión remota a *hendrix-ssh* o con entornos de desarrollo integrados, *IDE*, tales como *CodeLite* o similares)
- Algunas de las tareas se deben realizar necesariamente en el **clúster *hendrix-ssh*** o en los ***PCs de los laboratorios***
- **Desde dónde** conectarse con el clúster *hendrix-ssh*:
 - desde los terminales dispuestos en las salas informáticas de la Universidad
 - desde los computadores personales a los que cada uno tenga acceso (ver información en la web de la asignatura sobre cómo establecer conexión)

6. El equipo de profesores

- Profesores del grupo de mañana:
 - **Joaquín Ezpeleta** (clases de “teoría” y “problemas” en aula)
 - **Simona Bernardi** (supervisión de prácticas)
- Profesores del grupo de tarde:
 - **Javier Martínez** (clases de “teoría” y “problemas” en aula)
 - **Carlos Tellería** (supervisión de prácticas)
- Trabajamos **en equipo**, de forma **coordinada**, y evaluamos la asignatura **en común**

7. ¿Cómo va a ser la evaluación de la asignatura?

Pruebas de evaluación en junio:

- Trabajo obligatorio (**Trab**): 15%
- Examen individual en laboratorio (**ExLab**): 15%
- Examen individual escrito (**ExEsc**): 70% [mínimo 4.0 puntos]

Pruebas voluntarias durante el cuatrimestre (**PVol**):

- Permiten subir la nota de junio (no la de septiembre) hasta 1.0 puntos si has hecho todas las pruebas y has aprobado la asignatura en junio

Y tu nota en la asignatura en junio será:

- **ExEsc** $\geq$ 4.0 $\rightarrow$ Nota = $0.15 * \text{Trab} + 0.15 * \text{ExLab} + 0.70 * \text{ExEsc} [+ 0.10 * \text{PVol}]$
- **ExEsc** $<$ 4.0 $\rightarrow$ Nota = **ExEsc**

Y si te queda para septiembre:

- Se guardan las notas de junio: [Trab+ExLab] y [ExEsc]
- Nuevo examen individual en laboratorio [**ExLab**]: 30%
 - Si se hace este examen su nota sustituye a [Trab+ExLab] de junio
 - Si no se hace este examen [**ExLab**] = [Trab+ExLab] de junio
- Nuevo examen individual escrito [**ExEsc**]: 70%
 - Si se hace este examen su nota sustituye a [ExEsc] de junio
 - Si no se hace este examen [**ExEsc**] = [ExEsc] de junio

Y tu nota en la asignatura en Septiembre será:

- **ExEsc** ≥ 4.0 $\rightarrow$ Nota = $0.30 * \text{ExLab} + 0.70 * \text{ExEsc}$
- **ExEsc** < 4.0 $\rightarrow$ Nota = **ExEsc**

8. ¿De qué materiales vamos a disponer?

Material docente específico:

- J. MARTÍNEZ. *Curso de Programación 2: Diseño de algoritmos correctos y eficientes*. Servicio de reprografía de la EINA. Enero 2018
- J. MARTÍNEZ y J. EZPELETA. *Curso de Programación 2: Manual de prácticas*. Servicio de reprografía de la EINA y en la web de la asignatura. Enero 2018
- *Guía de estilo para programar en C++*. En la web de la asignatura

Libros de consulta:

- J.L. BALCÁZAR. *Programación metódica*. McGraw-Hill. 1993
- R. PEÑA. *Diseño de programas: Formalismo y abstracción*. Tercera edición. Pearson/Prentice Hall. 2005
- Bibliografía complementaria en la web de la asignatura

Web de la asignatura:

- Descripción y detalles sobre la organización de la asignatura
- Noticias, avisos e instrucciones
- Materiales docentes: copia de los materiales que se presenten en clase, enunciados de prácticas, exámenes de cursos anteriores, etc.
- Código de los programas que se citan en el curso
- Código y, en su caso, datos para facilitar el trabajo en prácticas
- Convocatorias de exámenes y calificaciones

<http://webdiis.unizar.es/asignaturas/PROG2/>

