



Universidad de Granada

decsai.ugr.es

Fundamentos de Programación

Grado en Ingeniería Informática y Matemáticas

Francisco José Cortijo Bon



DECSAI

**Departamento de Ciencias de la
Computación e Inteligencia Artificial**



**Departamento de Ciencias de la
Computación e Inteligencia Artificial**
Universidad de Granada

Fundamentos de Programación

**Grado en Ingeniería
Informática y Matemáticas**

Curso 2020-2021

Francisco José Cortijo Bon

Última modificación: 16 de Septiembre de 2020

Horario de clases

- **Teoría:**



- Viernes (17:30 - 19:30) – Aula 1.7

- **Prácticas:**



- Grupo 1 – Jueves (15:30-17:30) – 2.1
 - Grupo 2 – Viernes (15:30-17:30) – 2.8

Profesor



Francisco José Cortijo Bon

- Despacho 29 Dpto. CCIA (4ª planta)
- Email: **cb@decsai.ugr.es**

<http://decsai.ugr.es/index.php?p=miembros>

Tutorías



- Despacho 29 Dpto. CCIA (4ª planta)
- Tutorías :
 - Jueves(8:30-10:30, 17:30-19:30)
 - Viernes (8:30-10:30)

-HORARIO FLEXIBLE-

- Google Meet usando ...@go.ugr.es -



CITA CONCERTADA SIEMPRE CON ANTELACIÓN:
cb@decsai.ugr.es

<http://decsai.ugr.es/index.php?p=miembros>

Fundamentos de ... ¿qué?



Para nosotros un **ordenador** no es más que una máquina que funciona bajo el control de un Sistema Operativo (Windows, Gnu/Linux, IOS,...)

Fundamentos de ... ¿qué?

El ordenador es una “máquina” compuesta por diferentes tipos de dispositivos electrónicos:

- dispositivos de **almacenamiento** permanente de datos,



Fundamentos de ... ¿qué?

- dispositivos de **entrada**,



SoloStocks

LCD Monitor



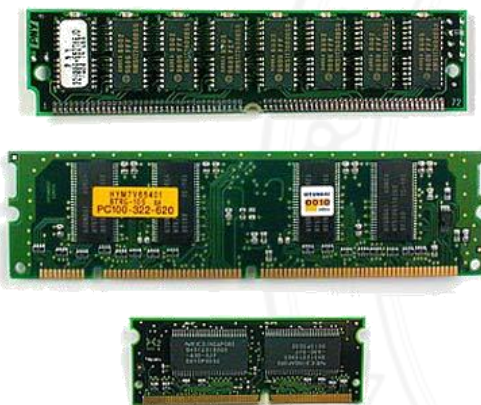
Fundamentos de ... ¿qué?

- dispositivos de **salida**,



Fundamentos de ... ¿qué?

- dispositivos de **cálculo y control** y **memoria (temporal)**



Fundamentos de ... ¿qué?

Los **PROGRAMAS** hacen que el ordenador sea capaz de **leer** datos e interpretar “órdenes” desde los dispositivos de entrada, **almacenarlos** y recuperarlos en/desde los dispositivos de almacenamiento masivo, **realizar cálculos**, **mostrar los resultados** en los dispositivos de salida.

Fundamentos de ... ¿qué?

La **programación** es el proceso de diseñar, codificar, depurar y mantener el código fuente de **programas**. El código fuente es escrito en un *lenguaje de programación*.

El propósito de la programación es crear programas que exhiban un comportamiento deseado.

En serio...

COMPUTER SCIENCE



What my friends think I do



What my mom thinks I do



What society thinks I do



What clients think I do

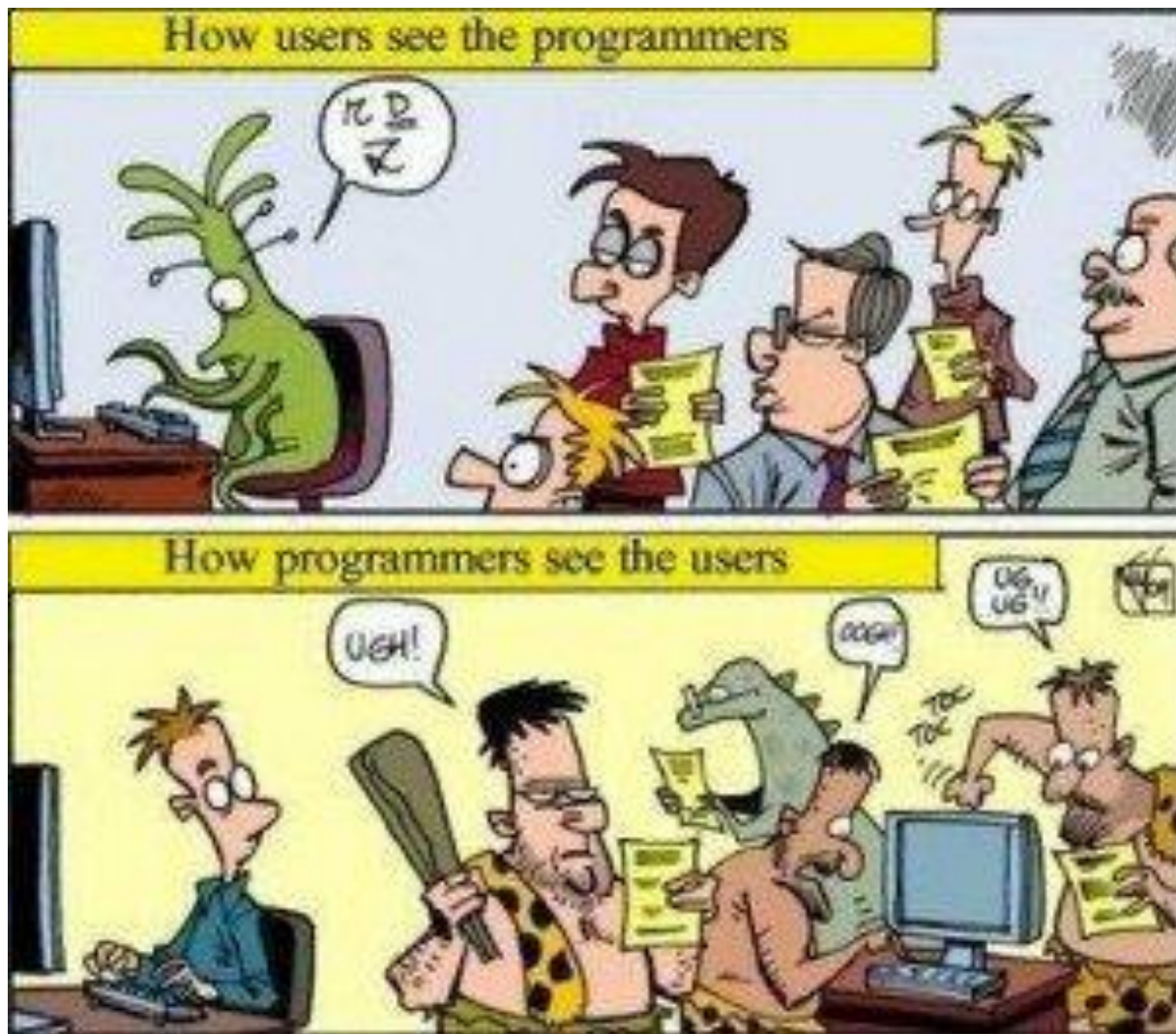


What I think I do



What I really do

En serio...



En serio...

I am programmer


QUOTES

11. "Programming today is a race between software engineers striving to build bigger and better idiot-proof programs, and the Universe trying to produce bigger and better idiots. So far, the Universe is winning."

- Rich Cook

I have no life

En serio...

A black zippered pouch with a white border and a white zipper pull. The text is printed in white on the black fabric.

pro•gram•mer
n. [proh-gram-er]
an organism that
turns caffeine and
pizza into software

En serio...

I am programmer

QUOTES

9. "Computer science education cannot make anybody an expert programmer any more than studying brushes and pigment can make somebody an expert painter."

- Eric S. Raymond (American programmer, open source software advocate, author of "The Cathedral and the Bazaar")

I have no life

En serio...

I am programmer

QUOTES

19. "Programming is like sex. One mistake and you have to support it for the rest of your life."

- Michael Sinz

I have no life

Acceso *identificado* a UGR

- ¿Para qué?

Acceso a los ordenadores, correo electrónico y servicios de la UGR.

- Más información en las páginas del CSIRC:

[Soy estudiante en UGR y... ¿ahora que?](#)



Prácticas

- Grupo de prácticas asignado.
- Grupo común para TODAS las asignaturas.
- Empiezan la semana 2 (28 Sep. – 4 Oct.).
- Nueva matrícula o problemas:
Escribir correo al profesor (cb@decsai.ugr.es)

Prácticas

● Guión de prácticas

- El alumno dispondrá de un **guión de prácticas** en el que se indicará el trabajo a realizar semanalmente.
- Como pauta general, se pedirá la realización de una serie de ejercicios **obligatorios** que se entregarán **semanalmente**, además de una serie de ejercicios **opcionales**.
- La realización de estos ejercicios y su defensa proporcionará al alumno puntuación en el apartado de "**evaluación continua**".

Prácticas

- Entornos de trabajo recomendados:

- **Orwell Dev-C++ 5.11**

- <http://orwelldvcpp.blogspot.com.es/>



- **Code::Blocks 13.12**

- <http://www.codeblocks.org/>



Code::Blocks

Temario

Apuntes	
1	Introducción a la Programación
2	Estructuras de Control
3	Funciones
4	Registros, vectores y matrices
5	Clases (parte I)
6	Clases (parte II)
7	Recursividad

Guia Docente	
1	Programación en C++: Fundamentos
2	Estructuras de Control
3	Funciones y clases
4	Vectores y matrices
5	Recursividad



Temario

- 1. Introducción a la Programación**
- 2. Estructuras de Control**
- 3. Funciones**
- 4. Registros, vectores y matrices**
- 5. Clases (parte I)**
- 6. Clases (parte II)**
- 7. Recursividad**



Temario (1)

- **Tema 1. Introducción a la Programación**

- 1.1 El ordenador, algoritmos y programas
- 1.2 Especificación de programas
- 1.3 Datos y tipos de datos
- 1.4 Operadores y Expresiones
- 1.5 Tipos de datos comunes en C++
- 1.6 El principio de una única vez

- **Tema 2. Estructuras de Control**

- 2.1 Estructura Condicional
- 2.2 Estructuras Repetitivas
- 2.3 Particularidades de C++

Temario (2)

- **Tema 3. Funciones**
 - 3.1 Fundamentos
 - 3.2 El principio de ocultación de información
 - 3.3 Funciones void
 - 3.4 Ámbito de un dato (revisión)
 - 3.5. Parametrización de funciones
 - 3.6 Programando como profesionales

Temario (3)

- **Tema 4. Registros, vectores y matrices**
 - 4.1 Registros
 - 4.2 Vectores
 - 4.3 Recorridos sobre vectores
 - 4.2 Matrices



Temario (4)

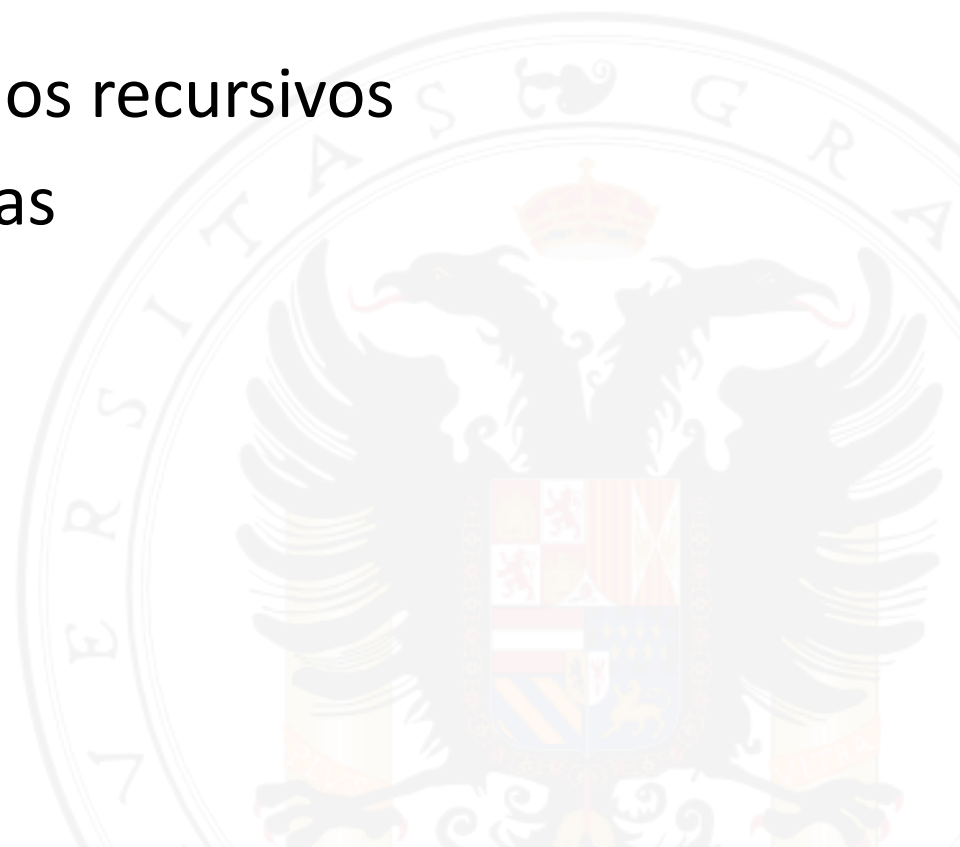
- **Tema 5. Clases (parte I)**
 - 5.1 Motivación: clases y objetos
 - 5.2 Encapsulación
 - 5.3 Ocultación de la información
 - 5.4 Constructores
 - 5.5 Copiando objetos
 - 5.6 Datos miembro constantes
 - 5.7 Programando como profesionales
 - 5.8 Vectores y clases
 - 5.9 La clase **SecuenciaCaracteres**
 - 5.10 La clase **string**

Temario (5)

- **Tema 6. Clases (parte II)**
 - 6.1 Métodos y objetos
 - 6.2 Objetos como datos miembro de otros objetos
 - 6.3 Tablas de datos
 - 6.4 Diseño de una solución (ampliación)
 - 6.5 Tratamiento de errores con excepciones
 - 6.6 Ciclo de vida del software (ampliación)

Temario (6)

- **Tema 7. Recursividad**
 - 7.1 Diseño de algoritmos recursivos
 - 7.2 Funciones recursivas



Bibliografía fundamental

- A. Garrido, A. *Fundamentos de Programación en C++*. Delta Publicaciones, 2005.
- R. Mercer, *Computing Fundamentals with C++. Object-oriented Programming and Design*. 2ª Edición. Palgrave Macmillan. 2000.
- T. Gaddis, J. Walters, G. Muganda. *Starting out with C++: early objects*. Pearson; 8va Ed. 2013
- Walter Savitch. *Resolución de problemas con C++*, Pearson, 2006.

Bibliografía complementaria

- A. Garrido, *Fundamentos de programación con la STL*. Editorial Universidad de Granada, 2016
- A. Garrido *Metodología de la Programación: de bits a objetos*. Editorial Universidad de Granada. 2016
- R. Martin. *Clean Code: A Handbook of Agile Software Craftsmanship*, Prentice Hall, 2008
- S. McConnell. *Code Complete: A Practical Handbook of Software Construction*. Microsoft Press; 2da Ed., 2004
- S. Prata. *C++ Primer Plus*. Addison-Wesley Professional (6 edition) 2011
- B. Stroustrup. *El Lenguaje de Programación C++*. Addison Wesley, 2001.

Método de evaluación (Convocatoria Ordinaria)

- **Teoría (60%)** : Examen escrito multipregunta.
- **Prácticas (40%)** :
 - Se realizarán pruebas de resolución de problemas a lo largo de la asignatura.
(p.e. **exámen(es) práctico(s)** individual(es) **(30%)** realizado(s) durante el cuatrimestre).
 - **evaluación continua (10%)**. Se desarrollará un guión de prácticas y se defenderá durante el desarrollo de las clases prácticas. Asistencia y entrega de los trabajos requeridos.

Método de evaluación (Convocatoria Ordinaria)

- Si la nota de la parte teórica es superior o igual a 3.5 sobre 10, entonces la nota final será:

$$\text{Nota final} = 0,6 * T + 0,4 * P$$

- En otro caso, $\text{Nota final} = T$

Se puede mantener T ó P para la convocatoria extraordinaria.

Método de evaluación

(Convocatoria Ordinaria)

“Evaluación única final”

- **Teoría (60%)** : Examen escrito multipregunta
- **Práctica (40%)** : Un examen práctico (en el ordenador).

Método de evaluación (Convocatoria Ordinaria) “Evaluación única final”

- Si la nota de la parte teórica es superior o igual a 3.5 sobre 10, entonces la nota final será:

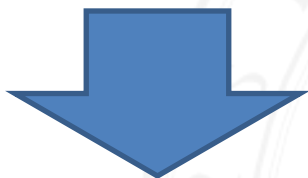
$$\text{Nota final} = 0,6 * T + 0,4 * P$$

- En otro caso, $\text{Nota final} = T$

Se puede mantener T ó P para la convocatoria extraordinaria.

Método de evaluación (Convocatoria Extraordinaria)

- **Teoría (60%)** : Examen escrito multipregunta
- **Práctica (40%)** : Un examen práctico (en el ordenador).



El alumno tiene la opción de conservar la nota que obtuvo (de cada parte) en la convocatoria ordinaria, o volver a evaluarse.

Método de evaluación (Convocatoria Extraordinaria)

- Si la nota de la parte teórica es superior o igual a 3.5 sobre 10, entonces la nota final será:

$$\text{Nota final} = 0,6 * T + 0,4 * P$$

- En otro caso, $\text{Nota final} = T$

Enlaces de interés

- C Plus Plus (en inglés)
<http://www.cplusplus.com>
- C++ Reference (en inglés)
<http://www.cppreference.com>
- C++ con clase <http://c.conclase.net>
- Zator (libro programación)
<http://www.zator.com/Cpp/>

Información adicional (acceso a la Guía Docente)

- Página web del Departamento:

<http://decsai.ugr.es>

Sección pública → Docencia | Asignaturas

- o directamente:

<http://decsai.ugr.es/index.php?p=asignaturas&accion=info&id=2971112>

Las cuentas CLARAS

1 ECTS = 25..30h Según doc. de la CRUE (Conferencia de Rectores de Universidades Españolas)

FP : 6 ECTS equivale a 150..180 horas totales

Clases presenciales por semana (4h)

- 2 horas de teoría
 - Lecciones magistrales en grupo grande
- 2 horas prácticas
 - Seminario, tutoría grupal ... en grupo pequeño

15 (-+1) semanas de clase

$15 \times 4 = 60$ h horas presenciales

Las cuentas CLARAS

6 ECTs equivalen a 150 horas

$15\text{Sem} \times 4\text{h/sem} = 60\text{h}$

$150\text{h} - 60\text{h presenciales} = 90$ horas de estudio personal

Estudio personal por semana

$90\text{h} / 15 \text{ semanas} = 6$ horas mínimo.

Y Si partimos de 1ECTS = 30h???

GRACIAS