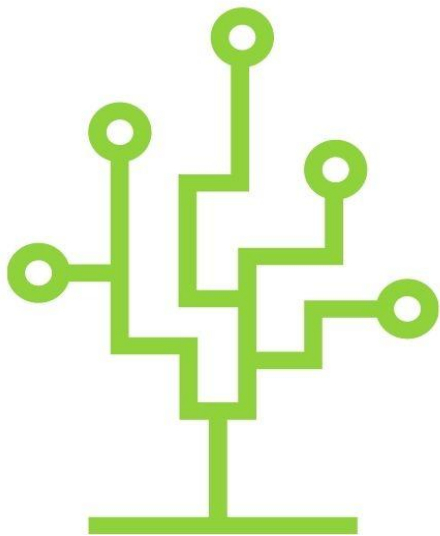


A grayscale background image showing a laptop screen with lines of code and a dark mug with a crown logo and the text 'KEEP CALM' next to it.

# GRUP D'EXPERTS

Bloc A



1r ASIX  
MP03 UF01

Daniel García Andrés  
Aleix Montero Sabaté  
Ismael Semmar Galvez  
Betlem Espuny Cuadrad

## **ÍNDEX**

<b>1- Projectes de desenvolupament d'aplicacions. Entorn integrats de desenvolupament.</b>	<b>3</b>
<b>2- Comentaris al codi.</b>	<b>3</b>
<b>3- Fonaments de programació.</b>	<b>4</b>
<b>4- Disseny d'algorismes.</b>	<b>4</b>
<b>5- Prova de programes. Depuració d'errors.</b>	<b>5</b>
<b>6. Documentació dels programes</b>	<b>5</b>
<b>7- Entorns de desenvolupament de programes.</b>	<b>6</b>

## 1- Projectes de desenvolupament d'aplicacions. Entorn integrats de desenvolupament.

Quan parlem d'entorns integrats de desenvolupament, o *Integrated Development Enviroment* (IDE), ens referim a l'escenari digital utilitzat a la programació per desenvolupar aplicacions, videojocs, etc... Fan que la feina del programador sigui més senzilla, gràcies a les eines que té incorporades com ara:

- Editor de codi: Es tracta d'un editor de text creat exclusivament per treballar amb el codi font de programes informàtics.
- Compiladors: Programa informàtic que tradueix un programa escrit en un llenguatge de programació a un altre, normalment l'objectiu és traduir-lo a codi màquina, encara que també pot ser traduït a un codi intermig o text pla.
- Depuradores: Programes informàtics que faciliten als desenvolupadors la feina de buscar errors al software.
- Biblioteques: A ciències de la computació, una biblioteca es un conjunt de subprogrames utilitzat per desenvolupar software.

Les eines poden variar depenent del IDE que s'estigui utilitzant, però en general tots tenen la mateixa finalitat que es augmentar la productivitat del programador.

## 2- Comentaris al codi.

Els comentaris de codi ens permeten als programadors deixar anotat en el codi text. Aquest no serà reconegut per el compilador i per tant no afectara al codi final del programa, és a dir, que no serà executat.

Molts cops pot ser que programem algo i ho fem de una forma tan enrevessada que en cas de voler editar-ho en un futur, tindrem dificultats per recordar la lògica del codi. Gràcies als comentaris podem evitar aquests casos.

Els comentaris també poden ser útils per si es dona el cas de que en un futur un altre programador es fiqui a editar el codi pugui entendre aquest sense cap problema.

En el cas de Java, la forma de afegir comentaris en el codi es la mateixa que a altres llenguatges de programació com bé podria ser per exemple Javascript o PHP.

Per fer-ho hi han dos formes:

Si fem // es comenta el text que vagi després fins al final de la línia.

**Exemple:**

//TODO: Acabar d'afegir variables (Exemple d'un comentari real)

**“Codi del programari”**

Podem comentar seccions utilitzant al inici /\* i utilitzant al final \*/.

**Exemple:**

```
/*
Programa comptabilitat del IES Ebre
Versió: 1.0
*/
```

**“Codi del programari”**

### 3- Fonaments de programació.

Si volem buscar una definició per a programar, la més encertada seria l'acció de crear una seqüència d'instruccions per a un ordinador, amb el fi de que pugui executar-les completant una feina determinada.

El llenguatge de les computadores es basa en la utilització d'algorismes, mitjançant la utilització d'aquests és possible indicar que realitzin accions determinades.

Segons quina sigui la estructura o característiques del nostre programa, podem diferenciar dos models:

- Programació Estructurada
- Programació Orientada a Objectes

La programació estructurada permet l'escriptura de programes fàcils de llegir i modificar, orientada a millorar la claredat, qualitat i temps de desenvolupament utilitzant únicament subrutines o funcions dins dels nostres programes.

El teorema del programa estructurat postula que, simplement amb la combinació de tres estructures bàsiques, és suficient per a expressar qualsevol funció computable:

- Seqüència: L'estructura seqüencial es la que es dona de forma natural en el llenguatge, perquè les sentències s'executen en l'ordre en el que apareixen al programa, una darrera l'altra.
- Selecció o condicional: L'estructura condicional es basa en que una sentència s'executa segons el valor que se li atribueix a una variable booleana, per tant aquesta estructura es pot executar de dos formes diferents, depenent del valor que tingui la variable.

- Iteració: L'estructura de repetició executa una o un conjunt de sentències sempre que una variable booleana sigui verdadera.

## 4- Disseny d'algorismes.

Un algorisme és un conjunt d'accions que especifiquen la seqüència d'operacions a realitzar, en ordre, per resoldre un problema.

Els algorismes són independents tant del llenguatge de programació com de l'ordinador que els executa.

Ha de complir les següents característiques:

- Un algorisme ha de ser precís i indicar l'ordre de realització de cada pas.
- Un algorisme ha d'estar definit. Si es segueix un algorisme dos vegades, s'ha d'obtenir el mateix resultat cada vegada.
- Un algorisme ha de ser finit. Si es segueix un algorisme, és deu acabar a algun moment, o sigui, ha de tenir un número finit de passos.

## 5- Prova de programes. Depuració d'errors.

Quan s'ha desenvolupat el pseudocodi de un algorisme o programa i l'introduïm a l'ordinador en forma de codi font, s'intenta la primera execució, el més normal és que sorgeixin errors i d'aquí obtindrem informació respecte l'estat del programa en quant a errors.

A partir d'un procés de prova i error, anirem transformant el nostre programa embrió en el programa desitjat.

En aquest món és probable que durant el desenvolupament del programa apareguin errors per molt menors que siguin. Per detectarlos la millor forma és anar provant el programa constantment i revisant que tot el que haguem anat escrivint al codi es vagi executant sense cap anomalia.

Normalment les grans companyies tenen els seus testers (famosos beta testers) que se encarreguen de realitzar la tasca de provar els programes. En el nostre cas no és així i ho haurem de realitzar nosaltres.

Una forma molt utilitzada per fer debugging es anar ficant missatges de consola per anar comprovant que el programa es va executant com volem.

## 6. Documentació dels programes

La documentació d'un programa pot ser interna o externa. La documentació interna es l'afegida mitjançant comentaris al codi. La documentació externa inclou anàlisi, diagrames de flux i pseudocodis, manuals d'usuari amb instruccions per executar el programa o per interpretar els resultats. La documentació d'un programa comença a la vegada que la construcció del mateix i finalitza justament abans de l'entrega del programa o aplicació al client.

Tot el procés de desenvolupament de Software hauria de quedar ben documentat. Això permet passar de una fase a l'altra de forma clara i definida.

En la documentació en el desenvolupament de Software veiem tres seccions:

### A) Guia tècnica

- a. Informació que conté:
  - El disseny de l'aplicació.
  - La codificació del programa.
  - Les proves realitzades
- b. Dirigit a:
  - Personal tècnic en informàtica (analistes y programadors)
- c. Objectiu:
  - Facilitar un correcte desenvolupament, realitzar correccions en els programes i permetre un manteniment futur.

### B) Guia d'ús

- a. Información que conté:
  - Descripció de la funcionalitat de la aplicació.
  - Forma de començar a executar l'aplicació.
  - Exemples d'ús del programa.
  - Requeriments software de l'aplicació.
  - Solucions dels possibles problemes que es poden presentar.
- b. Dirigit a:
  - Usuarios que utilizarán la aplicación (clients).
- c. Objectiu:
  - Donar als usuaris finals tota la informació necessària per utilitzar la aplicació.

### C) Guia de instal·lació

- a. Informació que conté:
  - Tota la informació necessària per posar en marxa la aplicació.
- b. Dirigit a:
  - Personal informàtic responsable de la instal·lació, en col·laboració amb els usuaris que van a fer servir l'aplicació (clients).
- c. Objectiu:
  - Donar tota la informació necessària per garantir que la implantació de l'aplicació es realitza de forma segura, fiable i precisa.

## 7- Entorns de desenvolupament de programes.

Un entorn de desenvolupament integrat o entorn de desenvolupament interactiu, en anglès Integrated Development Environment (IDE), és una aplicació informàtica que proporciona serveis integrals per facilitar-li al desenvolupador o programador el desenvolupament de programari.

Normalment, un IDE consisteix en un editor de codi font, eines de construcció automàtiques i un depurador. La majoria dels IDE tenen auto-completat intel·ligent codi (IntelliSense). Alguns IDE contenen un compilador, un intèrpret, o tots dos, com NetBeans i Eclipse; altres no, com SharpDevelop i Lazarus.

Els IDE estan dissenyats per maximitzar la productivitat del programador proporcionant components molt units amb interfícies d'usuari similars. Els IDE presenten un únic programa en què es porta a terme tot el desenvolupament. Generalment, aquest programa sol oferir moltes característiques per a la creació, modificació, compilació, implementació i depuració de programari. Això contrasta amb el desenvolupament de programari utilitzant eines no relacionades, com Vi, GNU Compiler Collection (GCC) o Make.

Els entorns de desenvolupament de programes més coneguts en Java són Eclipse, NetBeans, IntelliJ Idea, BlueJ, Codenvy, Xcode i jGRASP.

### Conoce estos 6 entornos de programación (IDE) para programar en varios lenguajes

- NetBeans. Netbeans también es un **entorno de programación** muy utilizado por los programadores. ...
- Visual Studio. ...
- JetBrains. ...
- QtCreator. ...
- CodeLite.