



Abans de començar ...

Programació - 1er DAW





**Quina experiència o
relació has tingut amb la
programació?**





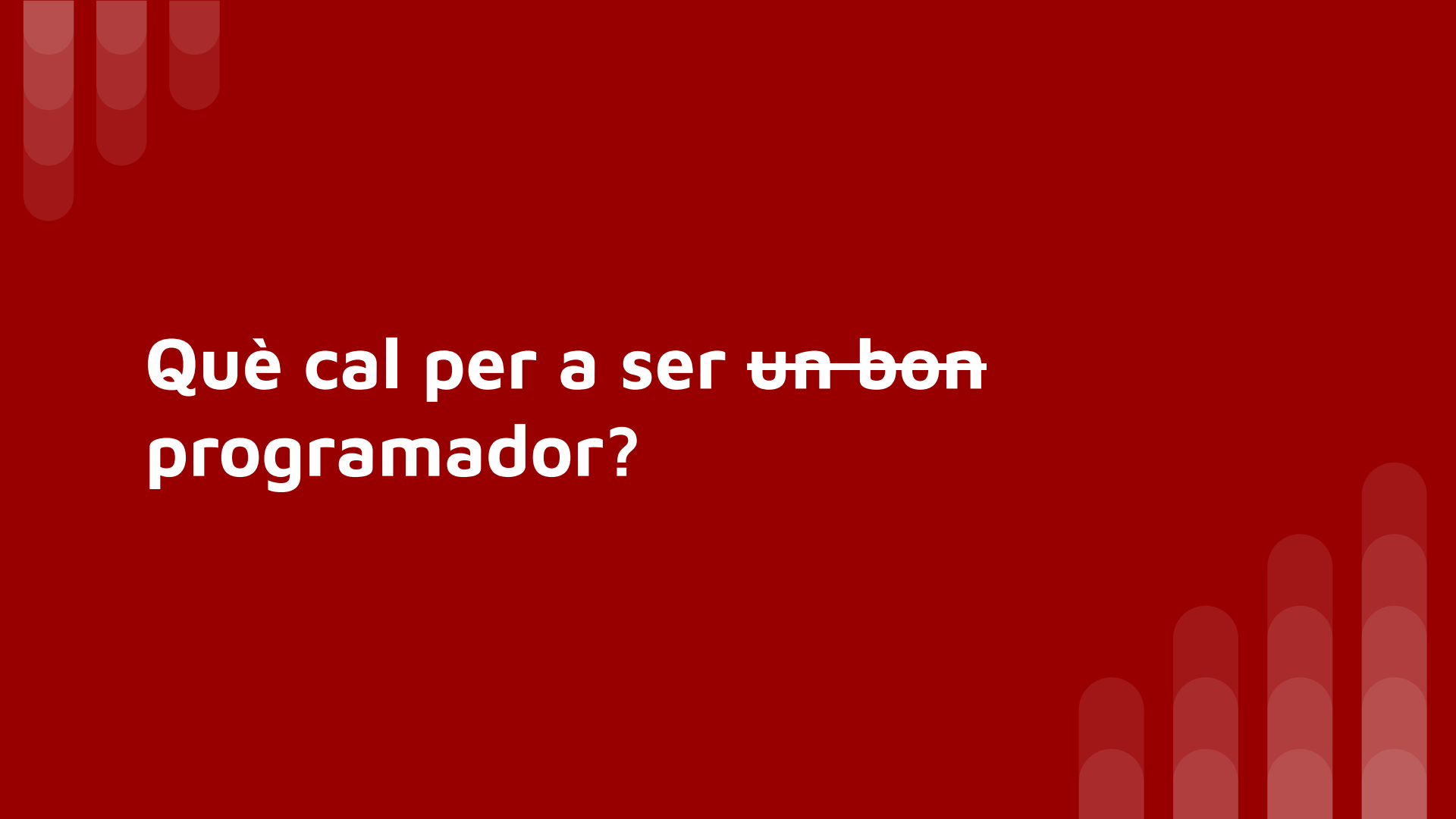
**Opinions.
Què és la programació?**




- **Donar instruccions...**
- **Resoldre problemes...**
- **Manipular dades...**

Un poc d'història:

https://www.youtube.com/watch?v=0yL_loiMbFI

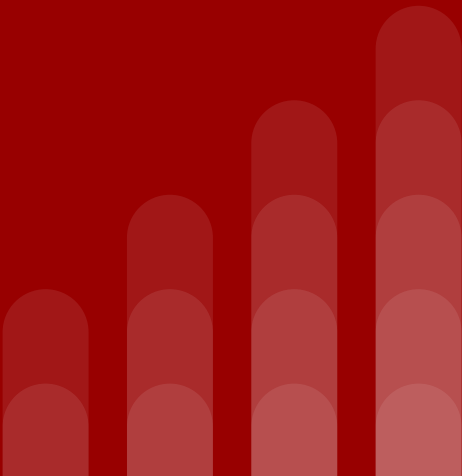
The background is a solid dark red color. In the top-left corner, there are three vertical bars of varying heights, each composed of three overlapping circles. In the bottom-right corner, there are four vertical bars of increasing height from left to right, each also composed of three overlapping circles.

**Què cal per a ser ~~un bon~~
programador?**



**Un programador ha de
tindre capacitat d'abstracció
i capacitat resolutiva.**

**Ha de ser analític i lògic en
l'enfocament per a la solució
de problemes**



Branques de la programació:

**Web, mòbil, d'escriptori, videojocs,
realitat virtual i augmentada, IA,
seguretat informàtica, sistemes
operatius ...**



A tindre en compte

- ★ Començarem des de zero.
- ★ Hi ha gent amb facilitat per a la programació.
- ★ I gent que li cal més esforç.
- ★ Ser conscients i **dedicar-li el temps** que siga necessari a casa per no perdre el fil. (no copieu)
- ★ Els continguts són acumulatius.
- ★ Hard skills vs. Soft skills.





Per a començar a aprendre programació

- ~~★ Cal saber molt de matemàtiques.~~
- ~~★ Has de ser un geni.~~
- ~~★ Només per a hòmens.~~
- ~~★ No cal esforçar-se, no té futur.~~
- ★ Paciència.
- ★ Honestat.



Què heu d'aprendre?





Què heu d'aprendre?

~~Ser experts en un llenguatge de programació.~~

1. Resolució de problemes i correcció d'errors
2. Conèixer diferents formes o estils (paradigmes) de programació.



Què heu d'aprendre?

1. Resolució de problemes i correcció d'errors

- Pensament computacional
- Aplicar la lògica per donar solució als problemes.
- ~~○ Escriure codi directament en un llenguatge de programació~~



Què heu d'aprendre?

1. Resolució de problemes i correcció d'errors

- Fases desenvolupament del programari:
 - Anàlisi, disseny, implementació, proves, desplegament i manteniment.
- Detecció i reconeixement dels errors. Aprendre a solucionar-los per experiència.



Què heu d'aprendre?

2. Conèixer diferents formes o estils (paradigmes) de programació.

- DAI → Programació estructurada
- Final del s.XX → Programació Orientada a Objectes (començarem este paradigma en el segon trimestre)
- Currículum educatiu queda antiquat massa ràpid (a final de curs → programació funcional)

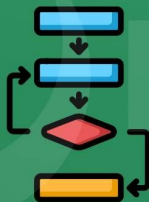
¿QUÉ SON LOS PARADIGMAS DE PROGRAMACIÓN?

Los paradigmas son los diferentes estilos de usar la programación para resolver un problema.



PROGRAMACIÓN ESTRUCTURADA

Programación secuencial con la que todos aprendemos a programar. Usa ciclos y condicionales.



PROGRAMACIÓN ORIENTADA A OBJETOS

Divide los componentes del programa en objetos que tienen datos y comportamiento y se comunican entre sí.



PROGRAMACIÓN REACTIVA

Observa flujos de datos asíncronos y reacciona frente a sus cambios.



PROGRAMACIÓN FUNCIONAL

Divide el programa en tareas pequeñas que son ejecutadas por funciones.





Què heu d'aprendre?

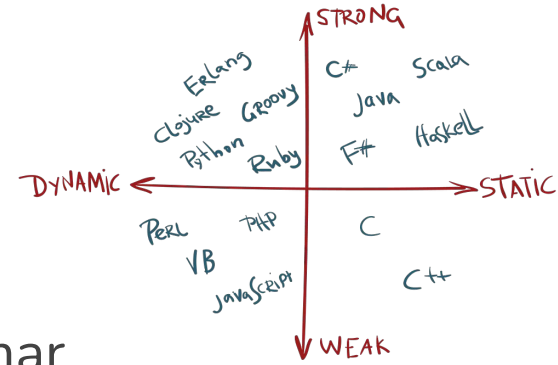
En resum, l'objectiu no és que domineu JAVA, que és un **llenguatge mastodòntic**. Veurem una xicoteta part de la seua API i funcionalitats que us servisquen per a que sigueu capaços en un futur d'**extrapolar-ho a qualsevol altre llenguatge o ampliar el que ja coneixeu** (fils, sockets, webservices, programació reactiva, frameworks de tot tipus...)



I per què JAVA i no un altre?



I per què JAVA i no un altre?



- ★ Realment importa aprendre a programar.
- ★ Altres centres treballen amb C#, Python o Kotlin.
- ★ *Senzill per aprendre programació vs Útil laboralment*
- ★ Creix amb força la demanda de JavaScript/TypeScript i Python però Java i C estan consolidats.
- ★ C++ o Java son bons llenguatges per a iniciar-se en la programació i és senzill saltar a altres després.
- ★ Tipat estàtic → Sempre és més senzill passar d'un llenguatge estricte a un menys estricte.



I per què JAVA i no un altre?

C++

- ★ És un llenguatge potent i molt estès però té un ús més específic i presenta dificultat en la gestió de punters i la recollida de basura.
- ★ Podríem dir que està a cavall entre els llenguatges de baix nivell i els d'alt nivell



I per què JAVA i no un altre?

JAVA

★ Avantatges:

- Basat en C però multiplataforma
- Més senzill de tractar els punters (referències) i la recoll·lecció de basura (que és automàtica)

★ Inconvenients:

- O.O. pur → Difícil per a explicar els procediments o funcions. Massa codi per a la prog. estructurada.
- Verbós → massa codi per a tasques simples.



Alternativa de futur

Encara que JAVA i C siguin usant-se, en el futur poden ser bones alternatives per aprendre programació les següents:

- **Python:** Per a programació general. Amb les últimes versions es pot forçar el tipat estàtic. Aporta més llibertat → cal més experiència.
- **TypeScript:** Per a la programació web. Genera codi JavaScript. Llenguatge (superconjunt de JavaScript) de tipat estàtic utilitzat en entorn client i servidor. (Recomanable una ullada abans de 2n curs)
- **Kotlin:** Es pot considerar l'evolució natural de Java. Més simple i llegible. Millor gestió de valors nuls... De moment, limitat laboralment.