

# BLOC B

## Design System com a Skills d'IA

*Track UX/UI*

---

Programa de formació en IA · Àrea d'Enginyeria i Desenvolupament

Corporació Catalana de Mitjans Audiovisuals

Maig 2026

# El diagnòstic

*El DS no és el problema. La traducció ho és.*



***La IA no resol això automàticament. Sense instruccions explícites, repeteix la inconsistència més ràpid.***

# Què és una skill UX/UI

*El salt mental d'aquest bloc.*

## Una skill UX/UI

és un fitxer markdown que descriu un component, patró o regla del Design System amb prou detall perquè un model d'IA generi codi conforme sense haver de mirar Figma.

### NO ÉS

- ✗ Una traducció literal de Figma
- ✗ Codi (no programem)
- ✗ Exhaustiu (cobrim el 80%)

### SÍ ÉS

- ✓ Intencions i restriccions del DS
- ✓ Markdown estructurat
- ✓ Operativa per a la IA

# Les skills són codi viu

*El Design System com a skills és un projecte en constant evolució.*

## **El DS a Figma i les skills al repo han d'anar de la mà.**

*Quan el DS evoluciona a Figma, les skills han d'evolucionar amb ell. Si no, la IA genera codi coherent amb una versió antiga.*

### **Qui manté les skills**

*L'equip UX/UI és el propietari. Els developers proposen canvis via PR, però UX/UI valida i aprova.*

### **Quan actualitzar**

*Quan es crea o modifica un component al DS. No cal esperar: una skill desactualitzada fa més mal que no tenir-ne.*

### **Com garantir coherència**

*Revisió periòdica DS ↔ skills. El GitHub Action de sincronització ajuda, però la decisió és sempre humana.*

# Anatomia d'una skill de component

*Les 6 seccions que ha de tenir cada fitxer.*

1

## Quan usar-lo

*Casos d'ús legítims. **Quan és l'eina correcta.***

3

## Variants i estats

*Primary, secondary, ghost, danger... hover, focus, disabled.*

5

## Regles d'accessibilitat

*WCAG 2.2 AA. **Imprescindible!***

2

## Quan NO usar-lo

*Casos on cal un altre component. **La part més útil.***

4

## Props / API

*Què accepta i què retorna. **El contracte.***

6

## Errors típics

*Els que la IA fa més habitualment. **Llistar-los explícitament.***

***Si la IA es desvia, és perquè una d'aquestes 6 seccions està buida o és ambigua.***

# Tokens, mappings, stacks

*Tres conceptes que connecten la skill amb el codi real.*

## Tokens

*Font única de veritat*

**Colors, espais, mides en JSON.**

`tokens.json` → CSS, Tailwind, SCSS via CI

## Mappings

*Skill → codi del stack*

**Com s'implementa la skill a cada stack.**

`button.react.md` · `button.telerik.md` · `button.html.md`

## Stacks

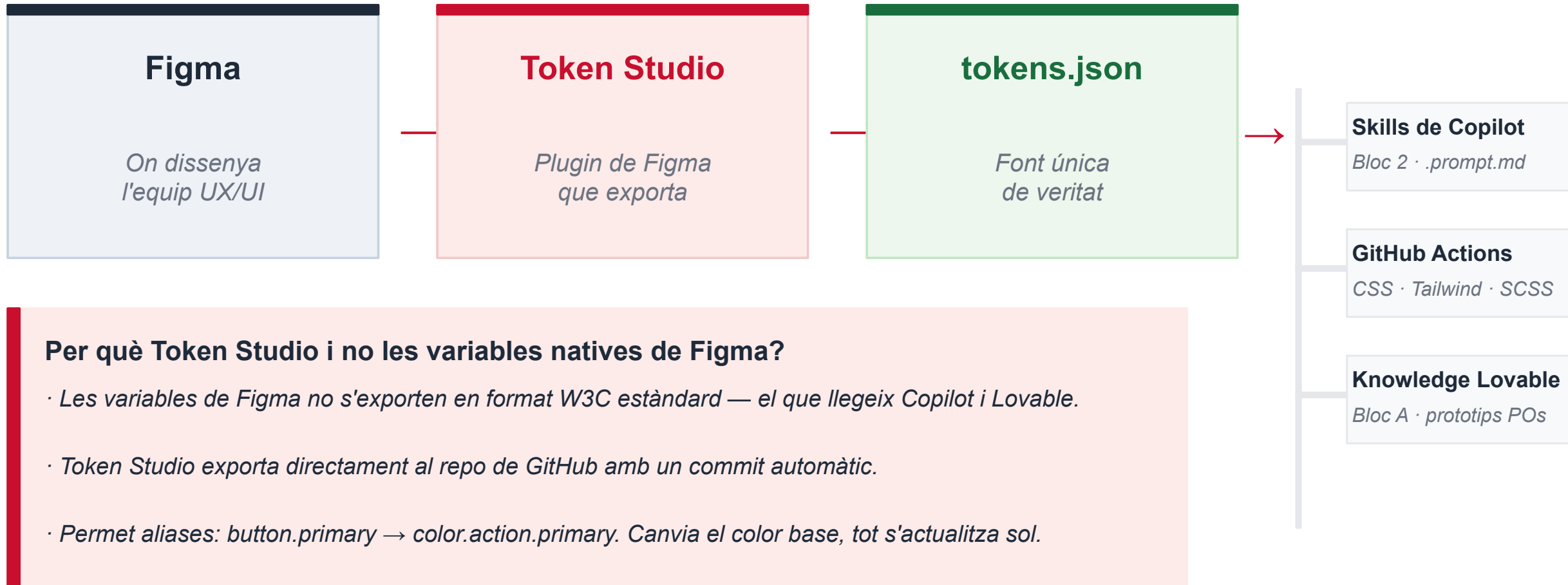
*On viu el codi*

**Els 3 stacks vius a 3Cat.**

`React/TS` · `.NET+Telerik` · `HTML+CSS`

# Token Studio: la font de veritat

*El tokens.json del nostre DS no viu a Figma. Viu al repo. Token Studio és el pont.*



# Exemple del procés

*Convertim el Button del DS en skill, pas a pas.*

1

Obrim Figma · component Button amb les seves variants

2

Obrim el repo 3cat-design-system · veiem l'estructura

3

Generem components/button.stack.md amb l'ajuda del Copilot

4

Commit + push · (OPCIONAL: GitHub Action sincronitza als repos)

5

Anem a 3cat-shows-web · demanem un botó a Copilot · respecta la skill

# L'exercici

*3 iterations*

## Iteració 1

### Explorar la skill Button

Ho veiem al repositori 3cat-design-system

## Iteració 2

### Generar un nou component

ProgramCard · ChannelBadge · LiveIndicator · TimeRangeFilter...

## Iteració 3

### Prova amb Copilot si segueix el design system

Un PO podria generar una UI correcta amb la skill que has creat?

# Abans de començar

*Dos repositoris, accés verificat, branca correcta.*

## 3cat-design-system

*Repositori de skills i tokens*

```
git clone github.com/3cat/3cat-design-system
cd 3cat-design-system
```

### Estructura rellevant:

```
├─ tokens/tokens.json ← font de veritat
├─ components/button/
│   └─ button.skill.md ← Iteració 1
│   └─ button.react.md
└─ _template.md ← plantilla
```

## 3cat-shows-web

*Repositori de validació · branca design-system*

```
git clone github.com/3cat/3cat-shows-web
cd 3cat-shows-web
git checkout design-system
npm install && npm run dev
```

**⚠ Treballeu sempre a la branca design-system**  
**No feu canvis a main**

### Checklist d'inici:

- ☐ Accés al repo 3cat-design-system verificat
- ☐ Branca design-system activa al shows-web
- ☐ Copilot actiu a l'IDE

# Que us emporteu del Bloc B

*Components al repo · una palanca multiplicadora.*

1

## **Un repositori 3cat-design-system viu**

*Amb components definits avui i una plantilla per a la resta.*

2

## **Una palanca, no més feina**

*Decidiu una vegada. Es propaga a 30 developers automàticament.*

3

## **El DS deixa de ser una recomanació**

*Passa a ser una restricció que la IA respecta a cada generació.*

4

## **Un consumidor immediat: el Bloc A**

*Els POs poden fer prototips amb les vostres skills.*

***El multiplicador és real. Cada skill val per cada cop que es generi UI a partir d'avui.***

PROPER PAS

Ens veiem al

# Hackathon!

prepareu les idees que voleu desenvolupar

**i organitzeu els grups de treball**

---