Grau en Enginyeria Informàtica

LABORATORI INTEGRAT DE COMPUTACIÓ

L'Enginyeria dels sistemes intel · ligents:
robustesa, eficiència i ètica en MLOps i IA Generativa

UAB
Universitat Autònoma de Barcelona

Jordi Gonzalez Sabaté
Departament de Ciències de la Computació

Juny 2026


BY NC ND

A la memòria del meu pare.

A Jorge Bernal, per haver fet evolucionar la meva manera d'entendre la docència: de Magikarp a Gyarados.

Noa, mai no et cansis de caminar...

Índex

Capítol 1

Fonaments de l'Aprenentatge Computacional: Del Símbol a la Dada

Aquest capítol obre el projecte transversal del Laboratori Integrat de Computació (LIC) amb una decisió explícita: el primer resultat del curs no serà un conjunt de gràfics, sinó un repositori executable. Abans d'entrenar models complexos, cal construir una base professional: estructura de projecte, configuració externa, partició correcta de dades, baseline, mètriques inicials, proves mínimes i traçabilitat experimental.

La teoria del capítol s'introdueix només quan ajuda a prendre una decisió de construcció: quan usar regles o models estadístics, com detectar memorització, com escollir mètriques, com separar dades i com justificar que un model generalitza. Les visualitzacions continuen sent útils, però passen a ser sortides secundàries d'un pipeline. El producte avaluable és el sistema D0–D1: codi encapsulat, executable i auditable.

Repte d'enginyeria

Repte del capítol. Construir el primer baseline reproduïble del projecte LIC.
Entregables associats: D0 i D1.
Artefactes mínims:

- repositori amb estructura `src/`, `configs/`, `tests/` i `reports/`;
- script executable amb configuració externa;
- partició `train/validation/test` amb `random_seed`;
- baseline de classificació amb mètriques inicials;
- primer registre experimental amb Weights & Biases o fitxer JSON equivalent;
- tests mínims de càrrega de dades i càlcul de mètriques.

Error típic

En aquest llibre, un notebook pot servir per explorar, però no és un lliurable suficient. Tot experiment que compti per avaluació ha de tenir una entrada executable, una configuració externa, una sortida registrada i una prova mínima.

Artefacte lliurable

Contracte tècnic D1. En acabar el capítol, el professorat ha de poder executar:

- `python -m lic_project.train --config configs/d1_baseline.yaml`
- `python -m lic_project.report_d1 --metrics-dir reports/metrics`
- `pytest`

Si alguna d'aquestes comandes falla, el capítol no s'ha completat com a laboratori d'enginyeria.

```
lic_project/
|-- configs/
|   |-- d1_baseline.yaml
|   `-- d1_experiments.yaml
```

Artefacte lliurable *(continuació)*

```
|-- artifacts/
|   `-- models/
|-- reports/
|   |-- figures/
|   |-- metrics/
|   `-- d1_report.md
|-- src/
|   `-- lic_project/
|       |-- __init__.py
|       |-- data.py
|       |-- models.py
|       |-- evaluate.py
|       |-- train.py
|       |-- report_d1.py
|       `-- experiments/
|           |-- compare_paradigms.py
|           |-- overfitting.py
|           |-- representation.py
|           |-- dim_reduction.py
|           |-- metrics_threshold.py
|           `-- anomaly_detection.py
|-- tests/
|   |-- test_data.py
|   |-- test_metrics.py
|   `-- test_training_contract.py
|-- requirements.txt
`-- README.md
```

Listing 1.1: Estructura mínima del repositori D0–D1

1.1 Paradigmes de Computació: Aprenentatge Simbòlic vs. Connexionista

La història de la Intel · ligència Artificial es pot narrar com la decisió de disseny entre dues hipòtesis fonamentals sobre la naturalesa del coneixement: la hipòtesi del sistema de símbols físics (Newell i Simon, 1976) i la hipòtesi del processament distribuït paral · lel (Rumelhart et al., 1986). En enginyeria moderna, comprendre aquesta dualitat no és un exercici històric, sinó una necessitat arquitectònica crítica: saber quan aplicar regles rígides i quan aplicar inferència estadística és el que distingeix una persona que programa i prova models d'una persona que construeix sistemes fiables. Cal tenir en compte que la dicotomia Simbòlic-Connexionista no és excloent. El futur de l'enginyeria de computació passa pels sistemes híbrids que integren la robustesa de les dades amb la verificabilitat de la lògica.

El Paradigma Simbòlic: Lògica i Cerca

L'enfocament simbòlic, sovint anomenat GOFAI (*Good Old-Fashioned AI*), postula que la intel · ligència emergeix de la manipulació sintàctica de símbols que representen conceptes del món real. Aquest paradigma és **declaratiu**: l'equip d'enginyeria descriu el *què* (el coneixement), i un motor d'inferència resol el *com*. Des d'una perspectiva formal, definim un sistema simbòlic $\mathcal{S}$ com una tupla $\langle \mathcal{B}, \mathcal{R}, \mathcal{I} \rangle$:

- $\mathcal{B}$ és la **Base de Coneixement** inicial (fets i axiomes).
- $\mathcal{R}$ és el conjunt de **Regles de Producció** (predicats lògics de primer ordre).
- $\mathcal{I}$ és el **Motor d'Inferència** que aplica regles com *Modus Ponens* o *Resolució*.

L'inferència es formalitza com una derivació lògica on es garanteix la veritat semàntica:

$$\mathcal{K} \vdash \phi \quad \iff \quad \forall M (M \models \mathcal{K} \implies M \models \phi) \tag{1.1}$$

On $\mathcal{K}$ és el coneixement i ϕ la conclusió. El principal repte matemàtic d'aquest enfocament no és la correcció, sinó la tractabilitat. La resolució de problemes simbòlics es redueix sovint a una cerca en un espai d'estats $\mathcal{S}_{state}$. Si b és el factor de ramificació (nombre de regles aplicables) i d la profunditat de la solució, la complexitat temporal tendeix a $O(b^d)$. Aquesta **explosió combinatòria** fa que els sistemes simbòlics siguin excel · lents per a dominis tancats (escacs, verificació de codi), però fràgils i computacionalment inaccessibles en entorns oberts i sorollosos (visió per computador, llenguatge natural).

El Paradigma Connexionista: Aproximació Universal

El connexionisme, la base del *Deep Learning* actual, rebutja la representació explícita i localitzada. El coneixement no resideix en un node específic, sinó que està **distribuït** en els pesos sinàptics (θ) de tota la xarxa. Matemàticament, modelem l'aprenentatge connexionista com un problema d'optimització no convexa en un espai de funcions. Sigui un dataset $\mathcal{D} = \{(X_i, y_i)\}_{i=1}^N$, busquem una funció f_θ parametritzada per θ que minimitzi el risc empíric:

$$\theta^* = \underset{\theta \in \mathbb{R}^d}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(f(X_i; \theta), y_i) + \lambda \Omega(\theta) \tag{1.2}$$

Per què confiem en aquestes xarxes? Fonamentalment pel Teorema d'Aproximació Universal (Cybenko, 1989), que estableix que una xarxa *feed-forward* amb una sola capa oculta i una funció d'activació no lineal $\sigma(\cdot)$ (com ReLU o Sigmoid) pot aproximar qualsevol funció contínua $f : [0,1]^n \to \mathbb{R}$ amb precisió arbitrària $\epsilon > 0$, donada una amplada suficient de la capa.

$$|F(X) - f(X)| < \epsilon, \quad \forall X \in [0,1]^n \tag{1.3}$$

Això converteix la tasca de "dissenyar regles" en "cercar paràmetres". **La Hipòtesi del Manifold**: Un concepte clau per a aquest curs és la **Manifold Hypothesis**. Les dades reals (com imatges o àudio) resideixen en un espai d'alta dimensió ($\mathbb{R}^{High}$), però la seva estructura rellevant es troba en un subespai topològic (manifold) de dimensió molt menor ($\mathcal{M}^{Low}$). L'aprenentatge profund funciona perquè les xarxes neuronals aprenen a "desplegar" aquest manifold per fer que les dades siguin linealment separables. Això contrasta amb el paradigma simbòlic, que intenta tallar l'espai original amb hiperplans rígids.

Intuïció operativa: El Llibre de Lleis vs. L'Artesà

Imagina que has d'identificar bitllets falsos en un entorn real (pressa, soroll, casos límit, falsificadors que evolucionen):

- **Simbòlic (Llei / Manual):** Tens un protocol explícit i auditable.
  - P. ex.: "Si el microtext no és llegible" o "si la marca d'aigua no apareix" ⇒ fals.
  - **Avantatge:** transparent, justificable, estable, fàcil de verificar i homologar.
  - **Limitació:** és **fràgil** davant l'*adaptació adversària*: si el falsificador sap les regles, optimitza per "passar el test".
  - **Risc típic:** massa falsos positius (bitllets reals amb defectes) o falsos negatius (falsificacions "noves" que esquiven criteris).

Intuïció operativa: El Llibre de Lleis vs. L'Artesà *(continuació)*

- **Connexionista (Artesà / Intuïció entrenada):** Has vist milers de casos i el teu cervell "comprimeix" patrons.
  - Exemple: "La textura, el contrast i la manera com reflecteix la llum *no quadren*".
  - **Avantatge:** captura **senyals febles** i combinacions subtils difícils d'escriure en regles; és **més robust** a variants.
  - **Limitació:** pot ser **opac** (costa d'explicar), i pot heretar **biaixos** de les dades (si entrenes amb exemples "poc representatius").
  - **Risc típic:** decisions correctes però difícils de defensar ("perquè ho diu el model").

Estratègia professional (què fan els equips bons): no és triar un bàndol, és **composar un sistema**.

- **Guardarraïls simbòlics:** regles mínimes que mai s'han de violar (p. ex. coherència física, rangs, consistència).
- **Nucli connexionista:** el model fa la detecció fina i generalitza a variacions.
- **Sortida explicable:** el sistema retorna *per què* (features, evidències, o un checklist de criteris activats) quan cal auditar.

Lectura operativa: el *manual* dona garanties i traçabilitat; l'*artesà* dona adaptabilitat. En aplicacions crítiques, el valor real ve de la **combinació** (neuro-simbòlica) i del disseny de controls.

Arquitectures Híbrides i el Compromís Enginyeril

Dins del context del Laboratori Integrat de Computació, l'objectiu no és la puresa acadèmica, sinó la resolució efectiva de reptes complexos. Això ens porta inevitablement a dissenys híbrids on es combinen els punts forts de cada paradigma. En l'enginyeria de sistemes crítics, com la conducció autònoma o la diagnosi mèdica, sovint utilitzem l'analogia dels sistemes cognitius de Kahneman:

- **Sistema 1 (Connexionista):** Ràpid, intuïtiu, patró-cèntric, però propens a biaixos i difícil d'auditar.
- **Sistema 2 (Simbòlic):** Lent, deliberatiu, lògic-cèntric, auditable i segur.

Una persona experta en enginyeria dissenya el sistema perquè el component connexionista gestioni la complexitat de la percepció (l'entrada de dades "bruta"), mentre que el component simbòlic gestiona la seguretat i la coherència del sistema (les restriccions fortes).

La taula 1.1 il·lustra que la integració no és una suma, sinó una estratègia de **mitigació de riscos**: utilitzem la lògica per "vallenar" (sandbox) la incertesa inherent de la xarxa neuronal.

Patró d'enginyeria: Injectant Lògica en Xarxes (Logic Regularization)

En sistemes crítics (salut, mobilitat, energia, finances), els equips professionals combinen dades i coneixement amb **Neuro-Symbolic AI**: no només "encaixar" patrons, sinó **respectar lleis i restriccions** del domini.
El truc (idea central): si saps que una magnitud física no pot ser negativa (p. ex. una distància $d \geq 0$), ho pots imposar afegint una penalització a la pèrdua quan el model viola la regla:

$$\mathcal{L}_{\text{total}}(\theta) \;=\; \mathcal{L}_{\text{dades}}(\theta) \;+\; \lambda \cdot \text{ReLU}(-y_{\text{pred}})$$

Taula 1.1: Desglossament Arquitectònic en Sistemes de Navegació Autònoma

Capa Funcional	Mòdul Connexionista (Deep Learning)	Mòdul Simbòlic (Regles/Lògica)
Percepció	*Segmentació Semàntica*: Identificar píxels de "carretera" vs "vorera" en condicions de pluja o enlluernament.	Impossible definir "vorera" amb equacions geomètriques robustes davant el caos visual.
Predicció	*Behavioral Prediction*: Estimar la probabilitat que un vianant creui basant-se en la seva postura corporal (no lineal).	*Física Cinemàtica*: Calcular si la frenada és possible donada la fricció μ i la velocitat v.
Decisió i Control	*Imitació (End-to-End)*: Copiar l'estil de conducció humà per ser "natural" i fluid.	**Hard Constraints (Safety Envelopes)**: $\forall t$, distància $>$ min_seguretat. Si es viola, activar frens d'emergència.

Patró d'enginyeria: Injectant Lògica en Xarxes (Logic Regularization) *(continuació)*

On $\text{ReLU}(-y_{\text{pred}}) = \max(0, -y_{\text{pred}})$ és **0** si $y_{\text{pred}} \geq 0$ i creix linealment si $y_{\text{pred}} < 0$. El paràmetre λ controla el **pes** de la restricció: si és massa petit, el model pot violar la regla; si és massa gran, pot empitjorar l'ajust a les dades.

Quan funciona millor:

- Quan la restricció és **sempre certa** (hard knowledge) i vols evitar prediccions absurdes.
- Quan tens **poca dada** o **molt soroll**: la lògica actua com a regularitzador.

Generalització a altres restriccions útils:

- **Rang:** $a \leq y \leq b \quad \Rightarrow \quad \lambda(\text{ReLU}(a - y) + \text{ReLU}(y - b))$
- **Monotonicitat (suau):** si X creix i y ha d'augmentar, penalitza derivades negatives: $\lambda \cdot \text{ReLU}\left(-\frac{\partial y}{\partial X}\right)$
- **Simetries/invariàncies:** penalitza si $f(X) \neq f(g(X))$ per una transformació g coneguda.

Idea clau: aquesta "lògica a la loss" fa de **guardarraïls matemàtics**: el model aprèn dels patrons, però dins d'un espai de solucions físicament coherent.

Laboratori de construcció: Primer baseline simbòlic vs. connexionista

Objectiu: comparar un model simbòlic senzill i un model connexionista, però fent-ho com a experiment reproduïble. El codi ha de quedar encapsulat en funcions i el resultat ha d'incloure mètriques, no només la figura de frontera de decisió.
Sortides esperades: mètriques de validació, figura comparativa, identificador del run i breu justificació de quin model usaríeu com a baseline.

Per entendre la diferència, visualitzem com cada model particiona l'espai de característiques (Feature Space).

- Els models simbòlics (Arbres) creen fronteres **ortogonals** (hiperplans paral · lels als eixos).
- Els models connexionistes (Xarxes) deformen l'espai topològicament creant **manifolds** suaus.

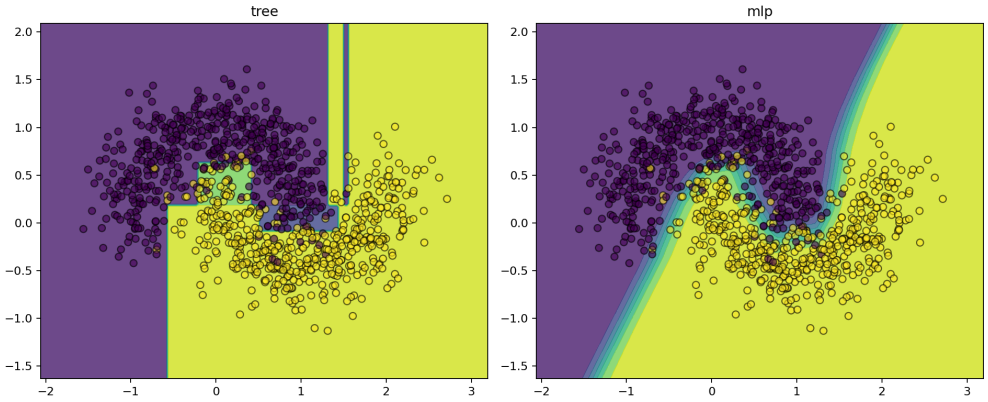

Figura 1.1: Comparació de fronteres de decisió: arbre simbòlic esglaonat i MLP connexionista.

La implementació completa s'ha de fer dins del repositori, no com a script aïllat. La visualització de fronteres es considera una sortida secundària del pipeline; la decisió de baseline s'ha de justificar amb mètriques registrades.

Artefacte lliurable

Criteri d'acceptació. La comparació ha d'incloure com a mínim `accuracy`, `f1`, `roc_auc`, identificador del run i una figura generada des del mateix codi que calcula les mètriques. No s'accepta una figura creada manualment fora del pipeline.

```
# Compara arbre de decisió i MLP sobre el mateix split.
# Registra mètriques i genera reports/figures/comparativa_paradigmes.png
python -m lic_project.experiments.compare_paradigms \
  --config configs/d1_experiments.yaml
```

Listing 1.2: Execució del laboratori de paradigmes

Microcas o incidència de laboratori: Fals positiu en un sistema de visió

Un equip entrena un classificador visual per detectar obstacles en imatges. En validació global obté bones mètriques, però durant una prova manual apareixen falsos positius quan hi ha ombres fortes o reflexos. El problema no és només del model: falta una capa de validació temporal i una regla de consistència física.

Acció d'enginyeria: afegir un guardarraïl simbòlic que exigeixi persistència de l'objecte en diversos frames abans d'activar una alarma crítica.

Lliçó per al D1: un model amb bona mètrica global pot ser insegur si no registrem errors típics i no definim criteris de decisió auditable.

1.2 Teoria de l'Aprenentatge Estadístic i Risc Empíric

Comprendre la distinció entre *memoritzar* i *aprendre* és una necessitat crítica per evitar el fracàs en producció. La decisió de disseny fonamental és garantir que un error baix es mantingui davant dades que mai s'han vist. Mentre que l'optimització busca el mínim matemàtic en una funció de cost coneguda, l'aprenentatge estadístic busca minimitzar el risc en una distribució desconeguda. Aquesta diferència subtil és la causa principal per la qual models amb un 99% de precisió al laboratori fallen estrepitosament en el món real.

Decisió d'arquitectura

D1 no premia el model més sofisticat, sinó el model més comparable. En el primer lliurable, la decisió professional és fixar una línia base que després pugui ser superada o rebutjada amb evidència. Un arbre senzill, una regressió logística o un MLP petit poden semblar poc ambiciosos, però són valuosos si deixen clar el split, la mètrica principal, el cost i el tipus d'error que el projecte accepta inicialment.
Alternativa descartada: començar directament amb un model profund. Aquesta opció pot donar una mètrica més alta, però fa més difícil saber si la millora ve de l'arquitectura, del preprocés, del split o d'un accident experimental.
Evidència que cal registrar: mètrica de validació, seed, configuració, temps aproximat d'execució i limitació principal del baseline.

Risc Real, Risc Empíric i Generalització

Definim un problema d'aprenentatge supervisat com la cerca d'una hipòtesi h dins d'un espai de funcions $\mathcal{H}$ (l'espai d'hipòtesis) que mapegi entrades $\mathcal{X}$ a sortides $\mathcal{Y}$. L'assumpció fonamental, i sovint oblidada, és que les dades (X, y) provenen d'una distribució conjunta **fixa però desconeguda** $\mathcal{P}(X, Y)$. Aquesta distribució encapsula la realitat del fenomen físic que volem modelar.

L'objectiu ideal de qualsevol sistema d'enginyeria és minimitzar l'error esperat sobre tota la distribució possible de dades futures. Això s'anomena **Risc Real** o Generalització ($R(h)$):

$$R(h) = \mathbb{E}_{(X,y)\sim\mathcal{P}}[\mathcal{L}(h(X), y)] = \int_{\mathcal{X}\times\mathcal{Y}} \mathcal{L}(h(X), y) d\mathcal{P}(X, y) \tag{1.4}$$

On $\mathcal{L}$ és la funció de pèrdua (Loss Function). Com que $\mathcal{P}$ és desconeguda, $R(h)$ és una magnitud incalculable directament. És vital entendre que existeix un límit inferior per a aquest risc, conegut com el **Risc de Bayes** (R^*):

$$R^* = \inf_{h\in\mathcal{H}_{all}} R(h) \tag{1.5}$$

Aquest és l'error irreductible degut al soroll estocàstic inherent a les dades o a la falta d'informació en les variables $\mathcal{X}$. Cap model, per complex que sigui, pot superar el Risc de Bayes.

Ara bé, només tenim accés a una "finestra" de la realitat: una mostra finita de dades d'entrenament $S = \{(X_i, y_i)\}_{i=1}^m$, mostrejades i.i.d. (independentment i idènticament distribuïdes) segons $\mathcal{P}$. El que minimitzem algorísmicament és el **Risc Empíric** ($\hat{R}_S(h)$):

$$\hat{R}_S(h) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}(h(X_i), y_i) \tag{1.6}$$

El principi de **Minimització del Risc Empíric (ERM)** postula que la hipòtesi que minimitza l'error en S ($\hat{h}^* = \arg\min_{h\in\mathcal{H}} \hat{R}_S(h)$) també funcionarà bé en la realitat. No obstant això, sense restriccions, l'ERM condueix directament al sobreajustament (overfitting), ja que simplement memoritzar S redueix $\hat{R}_S$ a 0. La teoria de Vapnik–Chervonenkis formalitza una idea essencial per al laboratori: un model amb molta capacitat pot reduir l'error d'entrenament sense millorar el comportament sobre dades noves.

Patró d'enginyeria

La pregunta no és "quin model té menys error d'entrenament?", sinó "quin model conserva millor el rendiment quan canvia la mostra?". Aquesta pregunta només es pot respondre amb un protocol experimental reproduïble.
Un model que obté un 100% d'encert en entrenament i no millora validació no és un model excel · lent: és un candidat a memoritzar. Sense validació i sense tracking, aquesta fallada pot quedar amagada.

Així, per a D1 no necessitem demostrar la fita completa; en necessitem la conseqüència pràctica:

- l'error d'entrenament no és una evidència suficient;
- cal una partició independent de validació;
- cal registrar hiperparàmetres, seed, mètriques i versió del codi;
- quan augmenta la capacitat del model, cal comprovar si baixa també l'error de validació.

Donat que minimitzar només $\hat{R}_S(h)$ és perillós, l'enginyeria moderna adopta el principi de **Minimització del Risc Estructural (SRM)**. En lloc d'optimitzar només l'error, optimitzem una suma ponderada d'error i complexitat:

$$h^*_{SRM} = \underset{h \in \mathcal{H}}{\operatorname{argmin}} \left(\hat{R}_S(h) + \lambda \Omega(h) \right) \tag{1.7}$$

On $\Omega(h)$ és una penalització monòtona amb d_{VC}. Aquest és el fonament teòric darrere de tècniques pràctiques que veurem al Capítol 2, com la regularització L_1/L_2 (Weight Decay) o el Dropout.

Intuïció operativa: L'Estudiant Memoritzador vs. L'Estudiant que Entén

Imagina dos estudiants preparant-se per a l'examen final de "Laboratori Integrat".

- **Estudiant A (Overfitting - Risc Empíric 0%):** Té una memòria fotogràfica. Ha memoritzat les respostes dels 50 exàmens d'anys anteriors. Si li preguntes qualsevol d'aquelles preguntes exactes, treu un 10.
- **Estudiant B (Generalització - Risc Empíric 5%):** Ha estudiat els conceptes fonamentals. De vegades s'equivoca en detalls dels exàmens passats, però entén la lògica.

El dia de l'examen final (Dades de Test), les preguntes són noves. L'Estudiant A suspèn perquè les preguntes han canviat lleugerament i la seva "funció" no generalitza. L'Estudiant B aprova.
Criteri d'enginyeria: Com a professionals de l'enginyeria, el nostre objectiu no és que el model tregui un 10 als deures (Training), sinó que aprovi l'examen final (Production).

Enginyeria del Compromís Biaix-Variança

En el disseny de sistemes reals, la gestió de la capacitat del model no és una qüestió d'intuïció, sinó una conseqüència directa del **Teorema de Descomposició de l'Error**. L'equip d'enginyeria ha de navegar entre dos tipus d'errors antagònics per trobar el punt de generalització òptim.

Si assumim que la realitat es modela com $y = f(X) + \epsilon$, on $\epsilon \sim \mathcal{N}(0, \sigma^2_\epsilon)$ és el soroll irreductible, l'error esperat de la nostra hipòtesi $\hat{f}(X)$ en un punt de prova X es descompon analíticament en tres termes:

Taula 1.2: Matriu de Diagnosi i Tractament del Biaix-Variança

Règim	**Símptomes Tècnics (Diagnosi)**	**Tractament (Enginyeria)**
Alt Biaix (Underfitting)	R_{train} és alt. $R_{val} \approx R_{train}$ (ambdós dolents). El model no captura l'estructura.	↑ **Complexitat:** Afegir capes/neurones. ↓ **Regularització:** Reduir λ o eliminar Dropout. **Feature Eng:** Afegir polinomis o atributs no lineals.
Alta Variança (Overfitting)	R_{train} és molt baix (prop de 0). $R_{val} \gg R_{train}$ (Gran "Generalization Gap"). El model és inestable.	↑ **Dades:** Aconseguir més mostres. ↑ **Regularització:** Augmentar Weight Decay ($\mathcal{L}_2$), Dropout. **Arquitectura:** Early Stopping, Reducció de dimensionalitat.
Règim Òptim (Sweet Spot)	R_{val} és mínim. La bretxa ($R_{val} - R_{train}$) és petita però existent.	**Data Augmentation:** Per trencar la simetria i millorar la robustesa sense canviar el model. **Ensembling:** Combinar models.

$$\mathbb{E}\left[(y - \hat{f}(X))^2\right] = \underbrace{\left(\mathbb{E}[\hat{f}(X)] - f(X)\right)^2}_{\text{Biaix}^2} + \underbrace{\mathbb{E}\left[(\hat{f}(X) - \mathbb{E}[\hat{f}(X)])^2\right]}_{\text{Variança}} + \underbrace{\sigma_\epsilon^2}_{\text{Error Irreductible}} \tag{1.8}$$

- **Biaix (Bias^2):** L'error introduït per aproximar un problema del món real (potser extremadament complicat) amb un model simplificat. Un alt biaix implica que el model ignora les dades (*Underfitting*).
- **Variança:** La quantitat en què canviaria la nostra estimació $\hat{f}$ si utilitzéssim un conjunt d'entrenament diferent. Una alta variança implica que el model "memoritza" el soroll específic de la mostra S (*Overfitting*).
- **Error Irreductible (σ_ϵ^2):** El límit inferior de l'error, associat a la qualitat de les dades i no al model. Cap enginyeria pot eliminar-lo; només millors sensors.

L'habilitat clau en aquest laboratori serà diagnosticar en quin règim es troba el vostre model observant les **Corbes d'Aprenentatge** (la bretxa entre R_{train} i R_{val}).

Patró d'enginyeria

Patró professional: abans de fer créixer el model, tanqueu el protocol.
Abans de provar models més grans, l'equip ha de comprovar que el split és correcte, que el seed està fixat, que les mètriques s'han registrat i que el baseline és reproduïble. Un model més complex pot millorar la validació, però també pot amplificar una fuita de dades o una mètrica mal triada.
Regla pràctica per al D1: no es permet comparar models si no comparteixen el mateix split, la mateixa mètrica principal i la mateixa configuració registrada.

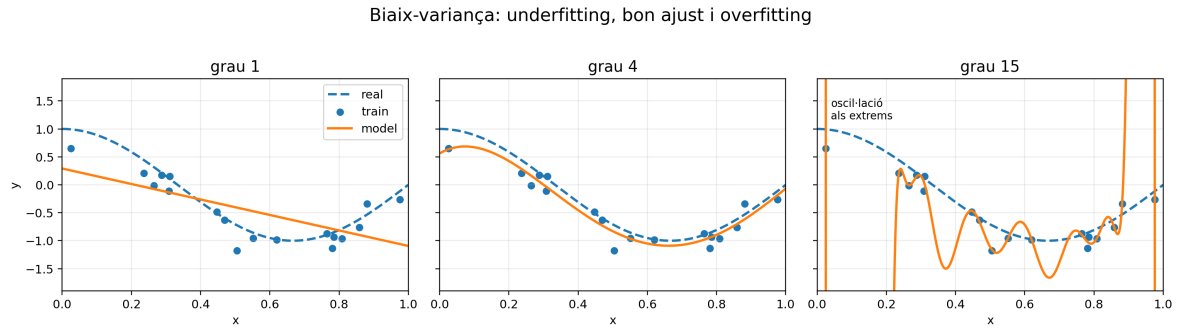

Figura 1.2: Compromís biaix-variança. Esquerra: un model de grau 1 té massa biaix i no captura la curvatura. Centre: un model de grau 4 aproxima bé l'estructura subjacent. Dreta: un model de grau 15 comença a seguir el soroll de la mostra i mostra oscil · lacions als extrems, símptoma típic d'alta variança.

Estratègia de validació

Test mínim de contracte D1. Abans d'acceptar un baseline, el repositori ha de tenir un test que comprovi tres propietats: el dataset es carrega, el split no és buit i les mètriques principals es poden calcular sense dependre d'un notebook.

Què protegeix aquest test? Evita que un canvi aparentment menor en noms de columnes, rutes o format de dades trenqui el pipeline complet. També força l'alumnat a separar codi de càrrega, codi d'entrenament i codi d'avaluació.

Què hauria de fallar si el sistema està malament? El test hauria de fallar si falta una columna esperada, si train i validation queden buits, si la mètrica retorna `nan` o si la configuració no fixa seed.

Laboratori de construcció: Baseline, overfitting i traçabilitat experimental

Objectiu: entrenar tres models de capacitat diferent i registrar per a cadascun l'error d'entrenament, l'error de validació i la diferència entre tots dos. La figura és útil, però el producte principal és la taula de runs.

Comanda mínima:

`python -m lic_project.train --config configs/baseline.yaml` En aquest laboratori compararem models de capacitat diferent i registrarem l'efecte sobre entrenament i validació. La figura d'overfitting és útil, però el resultat avaluable és la taula de mètriques i la justificació de quin model es pot defensar com a baseline.

```
# Entrena models de complexitat diferent, registra train/validation
# i genera reports/figures/bias_variance_demo.png
python -m lic_project.experiments.overfitting \
  --config configs/d1_experiments.yaml
```

Listing 1.3: Execució de l'experiment de biaix-variança

No accepteu conclusions basades en una sola execució sense seed fixat. Si canvieu el seed i la decisió canvia, el problema no és el model: és el protocol experimental.

Checklist de verificació

- El run registra pèrdua o error de train i validació?
- El model amb millor train és també el millor en validació?
- La conclusió proposa una acció: regularitzar, afegir dades o reduir capacitat?

Microcas o incidència de laboratori: Fuita temporal en la validació

Un grup obté resultats excel · lents en un recomanador perquè ha fet un *random split*. En revisar el pipeline, es detecta que el model pot aprendre patrons del futur per predir el passat. La validació és optimista i no representa l'ús real.
Acció d'enginyeria: substituir el *random split* per una partició temporal quan el problema tingui ordre cronològic.
Lliçó per al D1: el split no és un detall tècnic; és una hipòtesi sobre com funcionarà el sistema en producció.

Decisió d'arquitectura

Decisió P0: random split o time-based split?
El `train_test_split(random_state=42)` només és acceptable quan les instàncies poden considerar-se intercanviables i no hi ha dependència temporal, d'usuari, de sessió, de dispositiu o de versió documental. En molts sistemes reals, aquesta hipòtesi és falsa.
Regla del LIC. Si el dataset conté una columna de temps, una seqüència d'esdeveniments, logs, vendes, frau, manteniment predictiu, recomanacions o documents versionats, el split per defecte ha de ser temporal:

- entrenament: dades anteriors a una data de tall;
- validació: finestra posterior per ajustar decisions;
- test: finestra final no tocada fins al tancament.

Risc si es fa malament. Un random split pot entrenar amb patrons del futur i validar sobre el passat. El resultat és una mètrica artificialment alta al laboratori i una caiguda severa quan el sistema s'executa en producció.
Evidència obligatòria. El report D1 ha d'indicar explícitament: estratègia de split, columna temporal si existeix, data de tall, nombre d'exemples per finestra i justificació de per què el split és compatible amb l'ús real.

1.3 Classificació Supervisada: Geometria de l'Espai de Característiques

La classificació no és un mer procés d'etiquetatge, sinó un problema de **topologia en alta dimensió**. La decisió de disseny aquí resideix entre la *representació* i la *separabilitat*: un problema impossible de resoldre en l'espai original pot esdevenir trivial si trobem la projecció geomètrica adequada. L'objectiu no és només trobar una frontera de decisió, sinó garantir que aquesta frontera tingui el màxim **marge de seguretat** possible davant la incertesa inherent de les dades futures. Entendre la geometria subjacent (linealitat, convexitat, connectivitat) és el requisit previ per seleccionar l'algorisme correcte.

Formalitzem el problema de classificació binària supervisada. Tenim un conjunt d'entrenament $S = \{(X_i, y_i)\}_{i=1}^m$, on $X_i \in \mathbb{R}^d$ és el vector de característiques i $y_i \in \{-1, +1\}$ és l'etiqueta de classe. En el cas més simple, busquem un hiperplà definit per un vector normal θ i un biaix b que separi les classes:

$$f(X) = \text{sign}(\theta^T X + b) \tag{1.9}$$

L'optimització busca maximitzar el **marge geomètric** ($\gamma = \frac{2}{||\theta||}$), el que ens porta al problema primal de les *Support Vector Machines* (SVM):

$$\min_{\theta,b} \frac{1}{2}||\theta||^2 + C\sum_{i=1}^{m} \xi_i \quad \text{s.t.} \quad y_i(\theta^T X_i + b) \geq 1 - \xi_i \tag{1.10}$$

Taula 1.3: Geometria dels Algorismes de Classificació

Algorisme	Geometria de la Frontera	Cas d'ús d'enginyeria
Regressió Logística / SVM Lineal	**Hiperplà** (Recte/Pla). Simple, convex i auditable.	Problemes d'alta dimensió (Text, Genòmica) on el Teorema de Cover juga a favor.
Arbres de Decisió / Random Forest	**Ortogonal** (Caixes/Hiperrectangles). Fronteres esglaonades paral · leles als eixos.	Dades tabulars amb característiques categòriques o escales mixtes.
SVM (RBF) / Xarxes Neuronals	**Manifold** (Corbes suaus i tancades). Topologia complexa i pot ser no connexa.	Percepció (Imatges, Àudio) on la classe depèn de relacions no lineals complexes.

On ξ_i són les variables de folgança (*slack variables*) que permeten violacions del marge en dades no linealment separables (Soft Margin).

Per què tendim a projectar les dades a dimensions superiors (via Kernels o Xarxes Neuronals)? La justificació matemàtica és el **Teorema de Cover**. Aquest teorema estableix que la probabilitat que un conjunt de m punts aleatoris en un espai de d dimensions sigui linealment separable tendeix a 1 si la dimensió d és prou gran respecte a m.

$$P(\text{separable}) \to 1 \quad \text{si} \quad d \gg m \tag{1.11}$$

Això justifica l'estratègia del "Kernel Trick": mapar $\Phi : \mathbb{R}^d \to \mathbb{R}^D$ amb $D \gg d$ per linearitzar problemes complexos. De fet, triar la forma geomètrica de la frontera de decisió és la decisió de disseny més crítica, en la Taula 1.3 es resumeixen quan aplicar quin algorisme.

Intuïció operativa: El Tall de Ganivet i l'Art de l'Origami

Imagina que tens bales vermelles i blaves sobre una taula (2D).

- **Classificador Lineal (El Ganivet):** Intentes separar les bales posant un regle recte sobre la taula. Si les bales estan barrejades en cercles concentrics (vermelles al centre, blaves fora), és impossible separar-les sense moure el regle. Això és l'**Underfitting** geomètric.
- **Classificador No Lineal / Kernel (L'Origami):** Imagina que la taula és un full de goma elàstica. Si "estires" el centre del full cap amunt (afegint una 3a dimensió), les bales vermelles pugen. Ara pots passar una làmina plana (hiperplà) horitzontalment que talli la muntanya, deixant les vermelles a dalt i les blaves a baix.

Criteri d'enginyeria: En classificació, sovint no canviem el regle (el model lineal és robust i ràpid); canviem la taula (la representació de les dades) per fer el problema senzill.

Patró d'enginyeria: Linearitat en l'Espai d'Embeddings

L'alumnat novell tendeix a abusar de models no lineals (XGBoost, Deep Nets) per a tot. El patró dels sistemes d'escala massiva (Google Search, Spotify Recs) és utilitzar **Classificadors Lineals** (molt ràpids, $O(d)$ en inferència) sobre **Embeddings Preentrenats**.

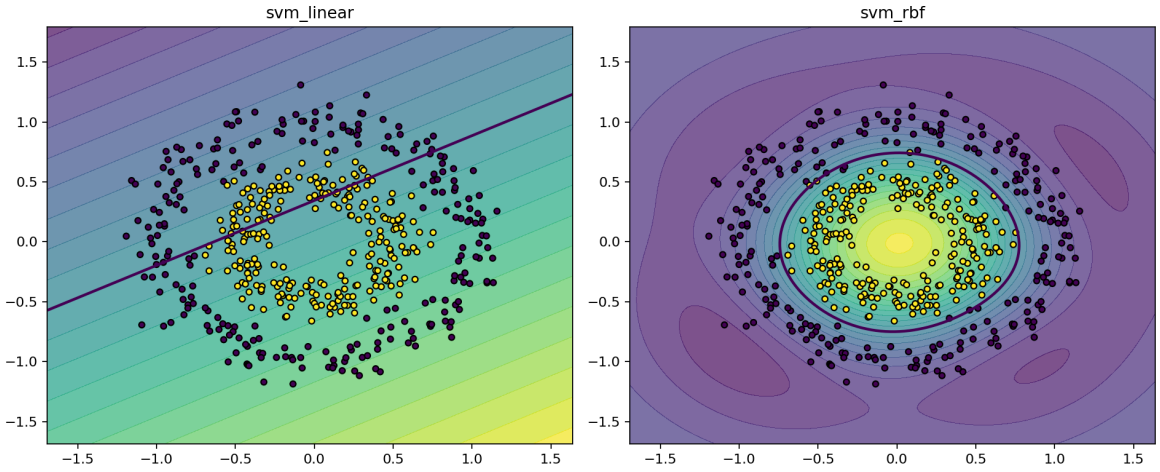

Figura 1.3: Visualització del Teorema de Cover. Esquerra: Un classificador lineal falla estrepitosament davant d'una topologia no convexa. Dreta: El Kernel RBF troba una frontera tancada perfecta, equivalent a tallar un con en 3D.

Patró d'enginyeria: Linearitat en l'Espai d'Embeddings *(continuació)*

En lloc d'entrenar un SVM amb Kernel (que escala malament, $O(m^2)$), utilitzen l'última capa d'una xarxa neuronal (ResNet o BERT) com a extractor de característiques $\phi(X)$.

$$f(X) = \theta^T \phi(X) + b \tag{1.12}$$

Això combina la potència de la representació profunda amb la velocitat i robustesa de la geometria lineal. Aquest és l'estàndard *de facto* en MLOps industrial.

Laboratori de construcció: frontera de decisió i elecció de representació

Objectiu: decidir si convé canviar el model o canviar la representació. En lloc de limitar-vos a dibuixar el kernel trick, heu de comparar una representació original amb una representació transformada i justificar el cost computacional de cadascuna.

En aquest laboratori no ens limitarem a visualitzar el *kernel trick*. Compararem representacions i models com una decisió d'enginyeria: model lineal ràpid, model no lineal més flexible, cost computacional i mètrica de validació.

```
# Compara SVM lineal i SVM RBF amb el mateix protocol d'avaluació.
# Genera reports/figures/kernel_trick_demo.png i metrics/kernel_metrics.json
python -m lic_project.experiments.representation \
  --config configs/d1_experiments.yaml
```

Listing 1.4: Execució de la comparació de representacions

La resposta no pot ser només "el RBF funciona millor". Cal justificar si el guany de validació compensa el cost, la menor interpretabilitat i el risc de sobreajustament.

Microcas o incidència de laboratori: Representació inadequada en un recomanador

Un equip intenta predir preferències de les persones usuàries amb metadades massa pobres: durada, any i categoria. El baseline és estable però no captura similituds semàntiques.

Microcas o incidència de laboratori: Representació inadequada en un recomanador *(continuació)*

En canviar a embeddings, un classificador lineal senzill millora perquè la representació ja conté informació útil.
Acció d'enginyeria: comparar primer representacions abans d'afegir models cada vegada més complexos.
Lliçó per al D1: quan un model falla, no sempre cal canviar l'algorisme; sovint cal canviar l'espai de característiques.

1.4 Agrupament No Supervisat i Reducció de Dimensionalitat

Sovint, les dades no són una font de veritat, sinó una font de soroll d'alta dimensionalitat. La decisió de disseny fonamental en aquest capítol és la **Maledicció de la Dimensionalitat** (*The Curse of Dimensionality*): a mesura que augmentem el nombre de variables (sensors, píxels, característiques), l'espai es torna exponencialment buit i les distàncies perden el seu significat. El repte arquitectònic no és classificar el que ja sabem (Supervisat), sinó descobrir l'estructura latent del que ignorem (No Supervisat). L'objectiu és trobar el "senyal" (manifold de baixa dimensió) amagat dins del "soroll" (espai d'alta dimensió), transformant el caos de les dades brutes en estructures compactes i accionables.

En l'aprenentatge no supervisat, tenim dades $\mathcal{D} = \{X_i\}_{i=1}^N$ sense etiquetes y_i. Busquem una representació simplificada Z_i o una partició C_k.

Reducció de Dimensionalitat: PCA i SVD

L'Anàlisi de Components Principals (PCA) busca una projecció lineal ortogonal que maximitzi la variança de les dades (i minimitzi l'error de reconstrucció). Matemàticament, això es resol mitjançant la **Descomposició en Valors Singulars (SVD)** de la matriu de dades centrada $\mathbf{X} \in \mathbb{R}^{N \times d}$:

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{1.13}$$

On:

- $\mathbf{U}$: Vectors singulars esquerres (coordenades en l'espai reduït).
- $\mathbf{\Sigma}$: Valors singulars diagonals (importància o "energia" de cada dimensió).
- $\mathbf{V}^T$: Vectors singulars drets (els eixos principals o *eigenvectors*).

Per reduir a k dimensions, trunquem la matriu quedant-nos amb els k valors singulars més grans: $\mathbf{X}_k = \mathbf{U}_k\mathbf{\Sigma}_k$.

Clustering: Optimització de la Compacitat (K-Means)

L'algorisme K-Means, tot i la seva simplicitat, planteja un problema d'optimització NP-hard que relaxem iterativament. Busquem K centroides μ_k que minimitzin la inèrcia intra-cluster (suma de distàncies quadràtiques):

$$J(\mu, C) = \sum_{k=1}^{K} \sum_{X_i \in C_k} ||X_i - \mu_k||^2 \tag{1.14}$$

Això equival a una tessel · lació de Voronoi de l'espai de característiques.

Taula 1.4: Trade-off en Tècniques de Reducció i Clustering

Tècnica	Naturalesa Matemàtica	Cas d'ús d'enginyeria
PCA	Lineal, Determinista, Global. Preserva la variança.	Preprocessament per eliminar multicolinealitat abans de Regressió o SVM. Compressió de senyal.
t-SNE / UMAP	No Lineal, Estocàstic, Local. Preserva la topologia de veïnatge.	**Visualització de dades**. Entendre clústers en embeddings complexos Z (ex: Word2Vec). No serveix per a preprocessament (no inverteix).
Autoencoders	No Lineal, Xarxa Neuronal. Aprèn una funció identitat compressa $X \approx g(f(X))$.	Denoising, Detecció d'Anomalies (reconstrucció fallida), Generació de dades.

Intuïció operativa: L'Ombra de Plató i la Compressió

Imagina que estàs en una cova (La Maledicció de la Dimensionalitat) observant ombres en una paret 2D projectades per objectes 3D complexos.

- **PCA (Projecció)**: És com girar l'objecte 3D fins a trobar l'angle que projecta l'ombra més gran i detallada. Si mires un llibre de cantó, l'ombra és una línia (perds informació, variança 0). Si el mires de pla, veus la forma rectangular completa (màxima variança). PCA busca automàticament aquest "millor angle".
- **Clustering (Organització)**: Imagina que tens milers de llibres llençats a terra. No saps els títols. L'algorisme els agrupa per "proximitat": els vermells amb els vermells, els grans amb els grans. No sap que són "Enciclopèdies", però descobreix que "existeix un grup d'objectes grans i vermells".

Criteri d'enginyeria: Reduir la dimensionalitat és **eliminar el soroll redundant** per quedar-se amb l'essència (l'ombra informativa).

No totes les projeccions són iguals. L'equip d'enginyeria ha de decidir si vol preservar distàncies globals (PCA) o veïnatges locals (t-SNE/UMAP), basat en els criteris detallats en la Taula 1.4

Patró d'enginyeria: La Maledicció de la Distància Euclidiana

Quan treballeu amb vectors d'alta dimensió ($d > 100$, com embeddings Z de text o imatge), la intuïció geomètrica falla. El **Fenomen de la Concentració de la Distància** demostra que:

$$\lim_{d\to\infty} \frac{\max_i ||X_i|| - \min_i ||X_i||}{\min_i ||X_i||} \to 0$$

En dimensions altes, **tots els punts són equidistants**. La distància Euclidiana (L_2) perd capacitat discriminatòria. **El Truc:** Utilitzeu sempre la **Similitud del Cosinus** per a espais d'alta dimensió, ja que mesura l'angle i no la magnitud, sent molt més robusta en espais buits.

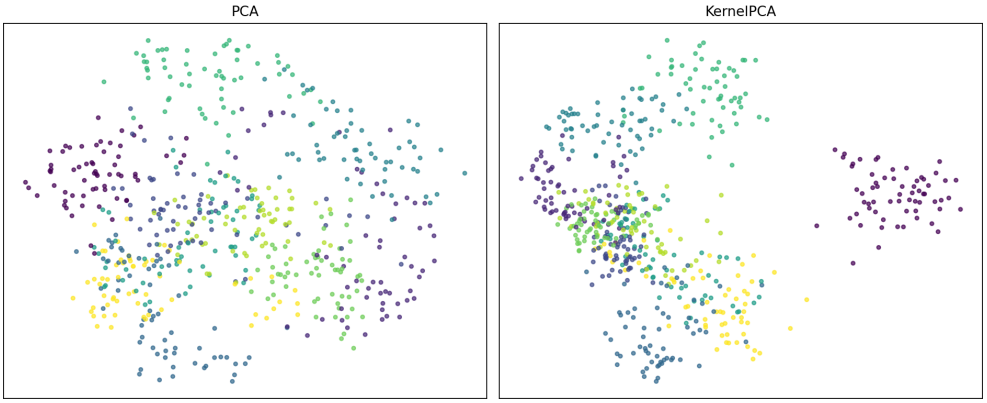

Figura 1.4: Projecció d'Alta Dimensió ($d = 64 \to d = 2$). Esquerra (PCA): Els dígits es barregen perquè la separació no és lineal. Dreta (t-SNE): L'algorisme troba l'estructura local i agrupa els dígits similars en illes separades, revelant l'estructura latent Z.

Laboratori de construcció: Reducció dimensional com a diagnòstic del dataset

Objectiu: usar PCA o t-SNE com a eina de diagnòstic, no com a conclusió final. L'alumnat ha d'explicar què revela la projecció, què no pot demostrar i quina decisió pràctica prendria sobre les dades.
En aquest laboratori utilitzarem PCA i t-SNE com a eines de diagnòstic, no com a producte final. La pregunta a respondre és si la representació actual del dataset permet separació útil o si cal canviar features, model o mètrica.

```
# Genera PCA i una projecció no lineal com a diagnòstic exploratori.
# Desa la figura i un resum de variància explicada.
python -m lic_project.experiments.dim_reduction \
  --config configs/d1_experiments.yaml
```

Listing 1.5: Execució del diagnòstic de dimensionalitat

Una projecció no lineal bonica no demostra que el model generalitzi. Serveix per formular hipòtesis sobre la representació, però la decisió final s'ha de validar amb mètriques independents.

Microcas o incidència de laboratori: Matriu dispersa i problema de cold start

Un sistema de recomanació té una matriu usuari–ítem gairebé buida. El model sembla funcionar en persones usuàries actives, però falla en persones usuàries noves o ítems poc vistos.
Acció d'enginyeria: registrar mètriques separades per segments: persones usuàries noves, persones usuàries recurrents, ítems populars i ítems poc freqüents.
Lliçó per al D1: una mètrica global pot amagar que el baseline només funciona en la part fàcil del dataset.

1.5 Mètriques d'Avaluació i Biaix-Variança en l'Era del Big Data

Reduir el rendiment d'un sistema complex a un sol número (com l'Accuracy) és un acte de negligència professional. La decisió de disseny en l'era del Big Data no és entre encertar o fallar, sinó entre la **Discriminació** (capacitat de distingir classes) i la **Calibració** (fiabilitat de la probabilitat predita). En entorns reals, les dades mai estan equilibrades i els costos dels

errors mai són simètrics. Un model amb un 99.9% de precisió pot ser totalment inútil si la classe d'interès (frau, càncer, fallada crítica) representa només el 0.1% de les dades. El disseny de sistemes ha d'incorporar mètriques que reflecteixin els objectius de negoci, no només la funció de pèrdua matemàtica.

Considerem un classificador binari $h(X) \to \hat{y} \in \{0, 1\}$. Definim la matriu de confusió en termes de probabilitats condicionades:

- **True Positive (TP):** $P(\hat{y} = 1|y = 1)$
- **False Positive (FP):** $P(\hat{y} = 1|y = 0)$ (Error Tipus I)
- **False Negative (FN):** $P(\hat{y} = 0|y = 1)$ (Error Tipus II)

Quan les classes estan desbalancejades, optimitzem la mitjana harmònica entre la puresa de la predicció (Precisió) i la capacitat de recuperació (Recall):

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN} \tag{1.15}$$

La mètrica F_β permet ponderar la importància relativa d'aquests dos factors, on β determina quant més important és el Recall respecte a la Precisió:

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{Precision} \cdot \text{Recall}}{(\beta^2 \cdot \text{Precision}) + \text{Recall}} \tag{1.16}$$

Per avaluar el canvi i la disparitat que causa el model a mesura que creix (o entre distribucions de probabilitat), definim mesures d'anàlisi de desviació com la Divergència KL i distàncies a l'espai de *Embeddings*, essencials als Capítols 4 i 5 d'aquesta obra. Una mètrica clàssica de similitud entre dues distribucions és la Divergència de Kullback-Leibler:

$$D_{KL}(P||Q) = \sum_{x \in \mathcal{X}} P(x) \log \left(\frac{P(x)}{Q(x)} \right) \tag{1.17}$$

Per avaluar l'alineació en l'espai Z es pot utilitzar el Maximum Mean Discrepancy (MMD) en espais d'Hilbert (RKHS), que mesura la distància de la mitjana de dos conjunts en un espai dimensional infinitament gran i ens permet determinar *Drift*.

Per avaluar el model independentment del llindar de decisió τ, integrem la taxa de vertaders positius (TPR) respecte a la taxa de falsos positius (FPR):

$$\text{AUC} = \int_0^1 \text{TPR}(\text{FPR}^{-1}(t))dt \tag{1.18}$$

Matemàticament, l'AUC representa la probabilitat que el model assigni una puntuació més alta a una instància positiva escollida a l'atzar que a una negativa.

Intuïció operativa: El Censor de Vots i el Detector d'Espies

Imagina dos escenaris on l'avaluació de sistemes de Machine Learning requereixen mètodes asimètrics:

- **Escenari A (El Detector de Malware - High Recall):** El CISO vol detectar tot accés maliciós. Si falla un (False Negative), els hackers poden buidar comptes d'usuari i la reputació cau en picat. Si genera falses alarmes de bloqueig (False Positive), el servei de suport ha d'esbrinar-ho però les dades resten segures.

Taula 1.5: Matriu de Decisió de Mètriques segons Restriccions de Domini

Context de Dades	**Mètrica Recomanada**	**Justificació d'enginyeria**
Classes Equilibrades (50/50)	Accuracy, ROC-AUC	L'error és simètric. Volem capacitat de discriminació global.
Imbalance Extrem (Frau 0.1%)	PR-AUC (Area Under Precision-Recall Curve)	ROC-AUC és enganyosament optimista en desbalancejos forts perquè TN és massiu. PR-AUC ignora TN.
Costos Asimètrics (Salut, Justícia)	F_2 Score (si FN és greu) o $F_{0.5}$ (si FP és greu)	Cal penalitzar explícitament el tipus d'error més costós per al sistema.
Calibració de Risc (Assegurances)	Brier Score / Log-Loss	No ens importa tant la classe (0/1) sinó la *probabilitat exacta* per calcular la prima.

Intuïció operativa: El Censor de Vots i el Detector d'Espies *(continuació)*

- **Objectiu:** Maximitzar el **Recall**. Volem atrapar *tots* els casos sospitosos, encarant els problemes d'ajust manual després.
- **Escenari B (El Model de Sentències judicials de crims - High Precision):** Si el model recomana llibertat quan l'acusat és de risc (False Negative), és un problema que pot implicar dany. No obstant això, si es condemna a l'atzar i equivocadament un ciutadà completament lliure per haver sigut marcat pel model (False Positive), estàs privant la justícia fonamental i la vida humana es trenca irreversiblement, creant una alerta de seguretat greu.
- **Objectiu:** Maximitzar la **Precisió**. Només volem bloquejar si n'estem molt segurs.

Criteri d'enginyeria: No existeix el "millor model" en abstracte. El millor model depèn de si et fa més mal perdre una oportunitat o cometre un error legalment sancionat.

L'elecció de la mètrica és una decisió d'arquitectura de negoci que s'ha de prendre abans de començar a entrenar, basat en uns criteris resumits en la Taula 1.5.

Patró d'enginyeria: Optimització del Llindar (Threshold Tuning)

Sovint, l'alumnat fa 'model.predict(X)' i accepten el llindar per defecte de 0.5. El **equips professionals** d'experts mai fa això. La sortida d'un model és una probabilitat p. La decisió final depèn de la **Matriu de Costos** C. El llindar òptim τ^* no és 0.5, sinó aquell que minimitza el Cost Esperat:

$$\tau^* = \frac{Cost(FP) - Cost(TN)}{Cost(FP) - Cost(TN) + Cost(FN) - Cost(TP)} \tag{1.19}$$

El Truc: Entrena el model per maximitzar l'AUC (discriminació pura) i, *en temps d'inferència*, ajusta dinàmicament τ segons els costos operatius del moment (ex: si tens menys operadors per revisar fraus, puges el llindar per reduir els FP).

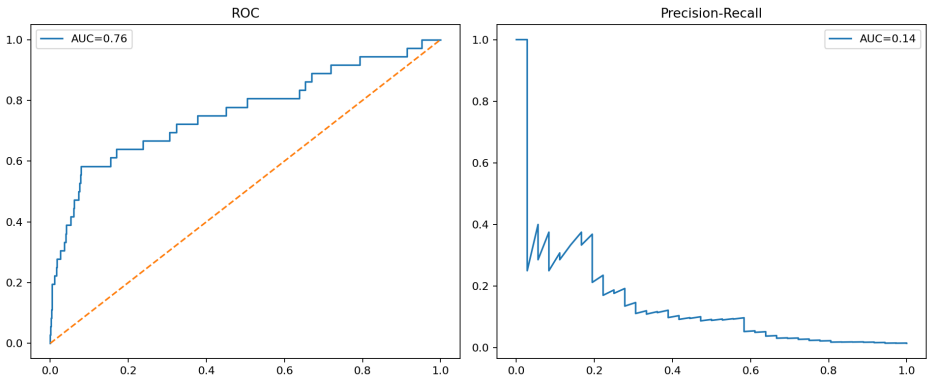

Figura 1.5: ROC vs. PR en dades desbalancejades. La ROC pot semblar excel · lent quan hi ha molts negatius fàcils; la PR revela millor la qualitat de les prediccions positives.

Error típic

No feu servir `model.predict(X)` sense revisar el llindar. El llindar per defecte 0.5 és una decisió implícita, no necessàriament una decisió correcta.

Laboratori de construcció: Mètriques, llindars i registre de resultats

Objectiu: comparar ROC-AUC i PR-AUC en un problema desbalancejat i registrar també el llindar de decisió. El lliurable no és només una corba: és una recomanació argumentada de mètrica i llindar segons el cost dels errors.
En aquest laboratori construireu una avaluació mínima per a dades desbalancejades. L'objectiu és demostrar amb mètriques registrades que l'accuracy i la ROC-AUC poden ser insuficients, i que el llindar de decisió s'ha de justificar segons el cost operatiu.

```
# Compara ROC-AUC, PR-AUC i mètriques a diferents llindars.
# Desa taules i corbes a reports/metrics i reports/figures.
python -m lic_project.experiments.metrics_threshold \
  --config configs/d1_experiments.yaml
```

Listing 1.6: Execució de l'avaluació en dades desbalancejades

Criteris d'acceptació

Per superar aquesta part, l'alumnat ha d'indicar quin llindar triaria, quin tipus d'error està acceptant i quina mètrica faria servir per defensar la decisió davant d'un responsable del sistema.

Microcas o incidència de laboratori: Mètrica global que amaga errors per subgrup

Un classificador presenta una accuracy global molt alta, però en separar els resultats per subgrups apareixen diferències grans en falsos positius i falsos negatius. El problema no es detectava perquè el report només mostrava una mètrica agregada.
Acció d'enginyeria: afegir al pipeline una avaluació per subgrups i un criteri d'alerta quan la diferència d'errors superi un llindar definit.
Lliçó per al D1: no hi ha avaluació professional sense desagregació quan el sistema pot afectar col · lectius diferents.

1.6 Microcas de Recerca: Detecció d'Anomalies en Xarxes de Sensors IoT

En enginyeria moderna, la teoria no serveix de res sense un domini d'aplicació on l'impacte sigui mesurable. En aquesta secció, sortim de la pissarra per entrar a la sala de màquines d'una infraestructura crítica. El problema que abordarem no és acadèmic; és industrial, sorollós i econòmicament vital. La detecció d'anomalies en l'Internet de les Coses (IoT) representa la quinta essència del repte "No Supervisat". No tenim etiquetes de "Fallada" perquè les fallades reals són rares, cares i, sovint, inèdites. Cal dissenyar un model que aprengui la "Normalitat" per, per exclusió, identificar el desastre abans que ocorri.

Definim el flux de dades d'un sensor com una sèrie temporal multivariant $X = \{X_t\}_{t=1}^{T}$, on $X_t \in \mathbb{R}^d$ representa lectures (temperatura, vibració, pressió) a l'instant t. Assumim que en règim normal, $X_t \sim \mathcal{N}(\mu, \Sigma)$.

Per detectar anomalies en un espai correlacionat, la distància Euclidiana falla. Utilitzem la Distància de Mahalanobis, que normalitza per la matriu de covariància Σ:

$$D_M(X) = \sqrt{(X-\mu)^T \Sigma^{-1} (X-\mu)} \tag{1.20}$$

Una observació és anòmala si $D_M(X) > \tau$, on τ es deriva de la distribució χ^2 amb d graus de llibertat.

Per a distribucions no gaussianes i d'alta dimensió, utilitzem l'algorisme *Isolation Forest*. Matemàticament, es basa en el principi que les anomalies són "fàcils d'aïllar". Si $h(X)$ és la longitud del camí mitjà en un bosc d'arbres binaris aleatoris per aïllar el punt X, definim la puntuació d'anomalia $s(X, n)$:

$$s(X, n) = 2^{-\frac{E[h(X)]}{c(n)}} \tag{1.21}$$

On $c(n)$ és la longitud mitjana d'un camí en un BST no exitós. Si $s(X, n) \to 1$, és una anomalia; si $s(X, n) \to 0.5$, és normal.

Intuïció operativa: L'Ovella Negra i el Bosc

Imagineu que voleu trobar una "ovella negra" en un ramat d'ovelles blanques mitjançant un joc de preguntes "Sí/No" (talls aleatoris).

- **L'Ovella Normal (Blanca):** Està envoltada de milers d'altres ovelles idèntiques. Per aïllar-la (quedar-se sol amb ella), has de fer moltes preguntes: "És al nord? Sí. A l'est? Sí. Prop del riu? No...". Camí llarg ($h(X)$ alt).
- **L'Ovella Anòmala (Negra):** Està sola, lluny del grup o té característiques úniques. Amb poques preguntes la separes: "És de color negre? Sí". Ja l'has aïllat. Camí curt ($h(X)$ baix).

Criteri d'enginyeria: No cal saber *què* és l'anomalia (pot ser negra, verda o gegant). Només cal saber que és "fàcil de separar" de la massa homogènia.

En un entorn industrial, l'algorisme és només una peça. El repte és la gestió del cicle de vida de la dada.

Patró d'enginyeria: Autoencoders per a la Reconstrucció de l'Error

Els experts en manteniment predictiu (ex: Siemens, GE) utilitzen **Deep Autoencoders**. La idea no és classificar, sinó comprimir i descomprimir. Entrenes una xarxa neuronal per copiar l'entrada a la sortida ($X \to Z \to \hat{X}$) utilitzant *només* dades normals. El model aprèn a "dibuixar" perfectament la normalitat. Quan arriba una anomalia (vibració

Taula 1.6: Disseny del Sistema de Detecció d'Anomalies IoT

Etapa	Repte Tècnic	Resposta d'enginyeria
Ingesta	Dades sorolloses, sensors morts, latència de xarxa.	Filtres de Kalman per suavitzar ($X_{t\|t-1}$) i finestres lliscants (Rolling Window) per gestionar l'estat.
Modelat	"Concept Drift": La màquina s'escalfa i canvia la seva "normalitat" a l'estiu.	**Aprenentatge Online**: Actualitzar μ i Σ o re-entrenar l'iForest cada setmana amb dades recents.
Decisió	Falsos Positius (FP) massius. Una alarma cada 5 minuts fa que l'operador la ignori (Fatiga d'Alarma).	**Ensembling Temporal**: Només disparar l'alarma si l'anomalia persisteix durant k finestres consecutives.

Patró d'enginyeria: Autoencoders per a la Reconstrucció de l'Error *(continuació)*

estranya), el model no la sap dibuixar bé perquè no l'ha vist mai. L'error de reconstrucció es dispara:

$$\mathcal{L}_{recon}(\theta) = ||X - \hat{X}||^2 > \tau \tag{1.22}$$

Això permet detectar fallades mecàniques subtils que mètodes estadístics simples (com Mahalanobis) passarien per alt degut a no-linealitats.

Laboratori de construcció: Detecció d'anomalies com a pipeline auditable

Objectiu: construir un detector d'anomalies amb un llindar justificat i un informe de falses alarmes. L'alumnat ha de separar clarament normalitat, anomalia, llindar, persistència temporal i criteri d'alarma.
En aquest laboratori construireu un pipeline mínim de detecció d'anomalies. El focus no és només detectar punts estranys, sinó deixar traça del llindar, del percentatge d'alertes i del criteri que faria que un operador confiés o no en el sistema.

```
# Entrena Isolation Forest, desa el llindar i calcula percentatge d'alertes.
# Genera reports/figures/anomaly_detection_demo.png
python -m lic_project.experiments.anomaly_detection \
  --config configs/d1_experiments.yaml
```

Listing 1.7: Execució del pipeline d'anomalies

El lliurable no és només marcar punts vermells. Cal entregar el criteri de llindar, el percentatge d'alertes, una estimació de falsos positius esperables i una decisió sobre si el sistema és utilitzable o necessita revisió humana.

Microcas o incidència de laboratori: Drift de context en sensors

Un detector d'anomalies entrenat amb dades normals d'estiu comença a marcar gairebé totes les lectures com a anòmales quan canvien les condicions ambientals. El model no ha descobert una fallada massiva; ha trobat un canvi de context.

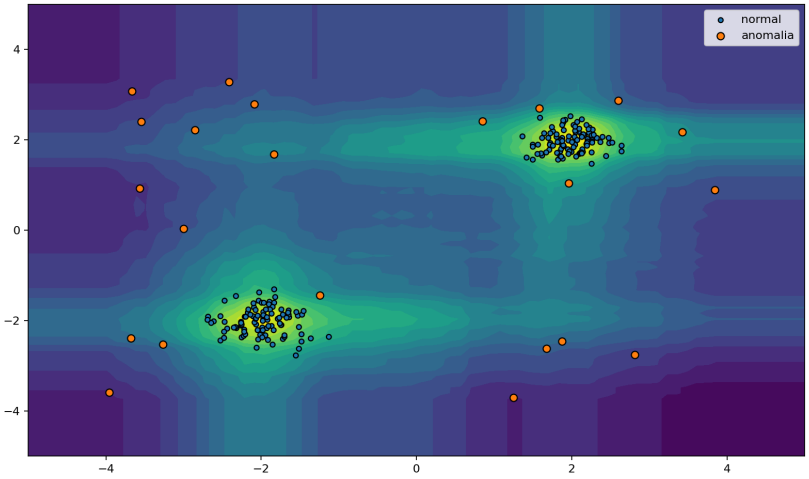

Figura 1.6: Mapa d'anomalies amb Isolation Forest. Les zones fosques indiquen normalitat alta; els punts vermells queden en regions fàcils d'aïllar i activen el llindar d'alerta.

Microcas o incidència de laboratori: Drift de context en sensors *(continuació)*

Acció d'enginyeria: registrar variables de context, comparar distribucions i separar anomalia de *covariate shift*.
Lliçó per al D1: una alerta estadística necessita interpretació operativa abans d'escalar-se com a incidència real.

1.7 Conclusions: De la Inducció a la Generalització

Tanquem aquest primer capítol tornant a la pregunta fonamental de l'enginyeria de computació: com podem construir sistemes que actuïn correctament en situacions que mai han vist abans? Hem viatjat des de la rigidesa dels sistemes simbòlics, on el coneixement s'injecta explícitament, fins a la flexibilitat dels sistemes connexionistes, on el coneixement s'indueix estadísticament.

La lliçó crítica per al vostre projecte al LIC és que **la dada no és la veritat; és només una ombra sorollosa de la realitat**. L'objectiu de l'aprenentatge automàtic no és minimitzar l'error en el dataset (això és trivial i es diu base de dades), sinó minimitzar l'error en el món real respectant uns mínims socials. Aquest salt de fe matemàtic es diu **Generalització**.

Per què no existeix l'algorisme perfecte? El **Teorema del "No Free Lunch" (Wolpert, 1996)** estableix que, si fem una mitjana sobre totes les possibles distribucions generadores de dades $\mathcal{P}$, qualsevol algorisme de classificació (incloent el llançament d'una moneda) té el mateix rendiment esperat.

$$\sum_{\mathcal{P}} \mathbb{E}[R(h_{alg_A})|\mathcal{P}] = \sum_{\mathcal{P}} \mathbb{E}[R(h_{alg_B})|\mathcal{P}] \tag{1.23}$$

Això implica que l'aprenentatge només és possible gràcies al **Biaix Inductiu** (*Inductive Bias*): les assumpcions prèvies que l'equip d'enginyeria introdueix en l'arquitectura.

- **CNNs (Capítol 2):** Assumeixen invariància translacional (un gat és un gat a l'esquerra o a la dreta).
- **SVMs:** Assumeixen que les classes estan separades per marges amples.
- **Veïns Propers:** Assumeixen suavitat local (punts propers tenen etiquetes similars).

Taula 1.7: Matriu de Decisió Arquitectònica (Resum Capítol 1)

Dimensió	Opció A (Baixa Complexitat)	Opció B (Alta Complexitat)
Representació	*Enginyeria de Característiques Manual*: Definir variables físiques (ex: velocitat, massa).	*Representation Learning (DL)*: Deixar que el model descobreixi features latents Z (ex: embeddings).
Modelat	*Models Lineals/Arbres*: Interpretables, ràpids, biaix alt.	*Mètodes de Kernel/Xarxes*: Caixa negra, lents, variança alta (risc d'overfitting).
Avaluació	*Accuracy/MSE*: Per a problemes equilibrats i senzills.	*Calibració/Fairness/OOD*: Per a sistemes crítics en producció.

Sense biaix inductiu, la generalització és matemàticament impossible.

Intuïció operativa: El Mapa i el Territori

Jorge Luis Borges va escriure sobre un imperi on la cartografia era tan perfecta que van crear un mapa a escala 1:1 amb el territori.

- **El Mapa 1:1 (Overfitting):** Coincideix perfectament amb la realitat punt per punt. Però és inútil com a eina de navegació perquè és tan gran i complex com la realitat mateixa. No comprimeix, no generalitza.
- **El Mapa Útil (Generalització):** És una abstracció. Elimina els arbres per mostrar el bosc. Un bon model d'IA és un mapa "incorrecte" (perquè perd detalls) però "útil" (perquè captura l'estructura invariant).

Criteri d'enginyeria: Com a professionals de l'enginyeria, no us enamoreu de la precisió del vostre mapa. Busqueu la utilitat de la vostra abstracció.

La taula 1.7 resumeix les decisions de disseny que heu d'afrontar en la Fase 1 del vostre Projecte Transversal.

Patró d'enginyeria: Incertesa Epistèmica vs. Aleatòria (Beyond Predictions)

Els equips professionals no prediu només y, sinó la confiança en y. Cal distingir dos tipus d'incertesa:

- **Aleatòria (Dades):** Soroll inherent. "La imatge és borrosa". No es pot reduir amb més dades.
- **Epistèmica (Model):** Ignorància del sistema. "Mai he vist un cotxe d'aquest tipus". Es redueix amb més dades.

En sistemes crítics (conducció autònoma, medicina), el vostre model ha de dir "No ho sé" (Incertesa Epistèmica alta) en lloc de fer una predicció aleatòria amb alta confiança. Utilitzarem *Bayesian Neural Networks* o *Dropout at Inference* per mesurar això.

Microcas o incidència de laboratori: Baseline que imita massa el passat

Un model aprèn molt bé les decisions històriques d'un sistema, però falla quan les condicions canvien. No ha après una regla general; ha après la política passada.

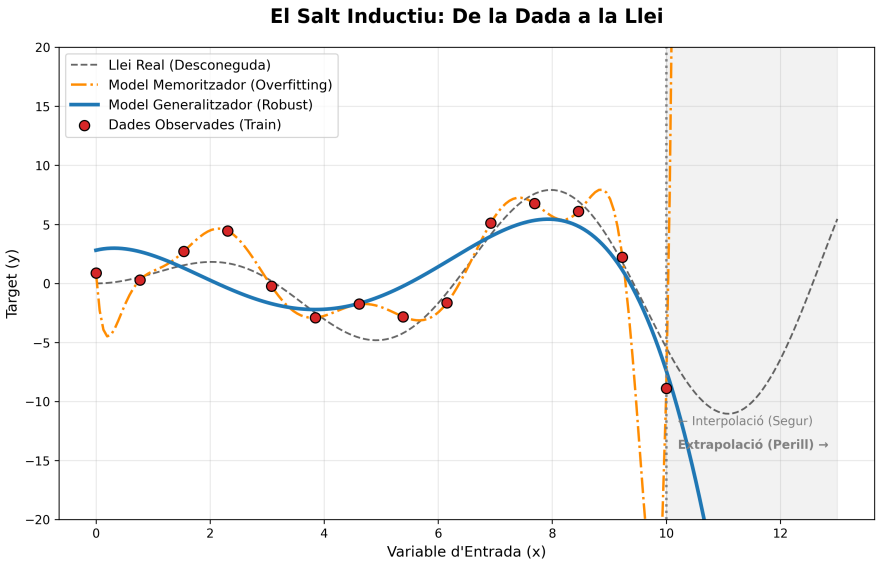

Figura 1.7: Corbes de pèrdua d'entrenament i validació en D2. La figura mostra un patró clàssic d'entrenament: la pèrdua d'entrenament continua baixant, però la pèrdua de validació arriba a un mínim i després empitjora. Això indica que el millor model no és necessàriament el de l'última època, sinó el checkpoint seleccionat segons validació. La decisió final s'ha de justificar amb aquest checkpoint, les mètriques de validació i la comparació amb el baseline D1.

Microcas o incidència de laboratori: Baseline que imita massa el passat *(continuació)*

Acció d'enginyeria: documentar explícitament quina hipòtesi inductiva incorpora el baseline i en quins escenaris pot deixar de ser vàlid.

Lliçó final del capítol: el D1 no busca el model més sofisticat, sinó el primer model defensable, reproduïble i honest sobre les seves limitacions.

Pregunta de defensa

Defensa tècnica del D1. El grup ha de poder respondre tres preguntes: quin baseline manté, quin error considera més crític i quin risc de generalització queda obert per al D2. Una resposta correcta no és "el model A té més accuracy", sinó una decisió defensable amb mètriques, limitacions i hipòtesi inductiva explícita.

Laboratori de construcció: Informe final D1 i comparació de runs

Objectiu: tancar D1 amb un informe tècnic breu. L'alumnat ha de presentar els runs executats, la decisió del millor baseline identificat, les mètriques principals, els riscos detectats i quines hipòtesis queden obertes per al Capítol 2.

```
# Agrega mètriques, figures i decisions en un informe reproduïble.
python -m lic_project.report_d1 \
  --metrics-dir reports/metrics \
  --output reports/d1_report.md
```

Listing 1.8: Generació de l'informe tècnic D1

Tancament del D1

Artefacte lliurable

D1 — Baseline reproduïble. Al final del capítol heu de lliurar:

- repositori amb estructura professional mínima;
- execució documentada de `python -m lic_project.train --config configs/baseline.yaml`;
- mètriques de train, validation i test;
- identificador de run a Weights & Biases o fitxer JSON equivalent;
- una taula comparativa de com a mínim dos models o configuracions;
- tests mínims executats amb `pytest`;
- informe breu de decisió: quin baseline es manté i per què.

Checklist de verificació

L'informe D1 ha de respondre quatre preguntes: quin baseline proposeu, quina evidència el suporta, quin error és més crític i quin risc de generalització queda obert per al Capítol 2. Així, abans de donar D1 per acabat, comproveu:

- el projecte s'executa amb una comanda documentada;
- les mètriques no depenen d'un notebook;
- el seed i la configuració queden registrats;
- hi ha almenys un test de dades i un test de mètriques;
- l'informe explica una decisió de disseny, no només mostra una figura.

Criteris d'acceptació

Criteri	Punts
Pipeline executable i sense dependència d'un notebook	2.0
Split correcte i reproduïble	1.5
Mètriques adequades i interpretades	1.5
Tracking amb Weights & Biases o registre JSON auditable	1.5
Codi encapsulat i estructura de repositori	1.5
Tests mínims	1.0
Informe tècnic breu i decisions justificades	1.0

Capítol 2

Deep Learning i l'Arquitectura Transformer

El Capítol 1 ha deixat tancat el primer contracte del projecte: un baseline reproduïble, amb split, mètriques, tracking i tests. En aquest capítol el repte ja no és demostrar que un model pot aprendre, sinó entrenar una xarxa profunda de manera controlada: configuració externa, bucle d'entrenament, validació, checkpoint, corbes de pèrdua, comparació d'hiperparàmetres i registre d'artefactes.

El *Deep Learning* no s'introdueix aquí com una successió d'arquitectures espectaculars, sinó com una disciplina d'enginyeria: una xarxa neuronal és codi que pot divergir, consumir massa memòria, memoritzar dades, quedar mal calibrada o produir resultats no reproduïbles. Per tant, el producte del capítol no és una figura de mapes d'activació, sinó el segon lliurable del projecte transversal.

Repte d'enginyeria

Repte del capítol. Convertir el baseline D1 en un pipeline d'entrenament profund reproduïble.
Entregable associat: D2.
Artefactes mínims:

- un model PyTorch encapsulat en `src/lic_project/torch_models.py`;
- un bucle d'entrenament executable amb configuració externa;
- registre de *loss*, mètriques i hiperparàmetres amb Weights & Biases o JSON auditable;
- checkpoint del millor model segons validació;
- comparació d'almenys dues configuracions: optimizer, learning rate, regularització o arquitectura;
- tests mínims de dades, model i contracte d'entrenament;
- informe tècnic D2 amb decisió justificada.

Laboratori de construcció: Les visualitzacions no ho són tot

Les visualitzacions d'aquest capítol —corbes d'entrenament, mapes de característiques, mapes d'atenció o fronteres regularitzades— són sortides secundàries del pipeline. El criteri d'acceptació és que el professorat pugui executar el mateix entrenament, recuperar el checkpoint i verificar les mètriques.

Error típic

Un entrenament que només funciona dins d'un notebook no és un entrenament reproduïble. En D2, tota decisió ha de quedar associada a una configuració, un seed, un run, unes mètriques i un artefacte de model.

Artefacte lliurable

Contracte tècnic D2. En acabar el capítol, el professorat ha de poder executar:

```
python -m lic_project.deep_train --config configs/d2_training.yaml
python -m lic_project.report_d2 \
    --metrics-dir reports/metrics \
    --output reports/d2_report.md
pytest
```

Si alguna d'aquestes comandes falla, el capítol no s'ha completat com a laboratori d'enginyeria.

```
lic_project/
|-- configs/
|   |-- d1_baseline.yaml
|   |-- d2_training.yaml
|   `-- d2_experiments.yaml
|-- artifacts/
|   `-- models/
|-- reports/
|   |-- figures/
|   |-- metrics/
|   `-- d2_report.md
|-- src/
|   `-- lic_project/
|       |-- deep_data.py
|       |-- torch_models.py
|       |-- deep_train.py
|       |-- deep_evaluate.py
|       |-- report_d2.py
|       `-- experiments/
|           |-- optimization.py
|           |-- cnn_features.py
|           |-- attention_map.py
|           |-- transfer_probe.py
|           |-- regularization.py
|           |-- segmentation.py
|           `-- scaling_laws.py
`-- tests/
    |-- test_deep_data.py
    |-- test_deep_models.py
    `-- test_deep_training_contract.py
```

Listing 2.1: Estructura incremental D2 sobre el repositori D1

2.1 El Perceptró Multicapa i el Grau de Llibertat de l'Optimització

Una xarxa neuronal no és una caixa màgica: és una composició de funcions parametritzades que només és útil si el seu entrenament és estable i auditable. En D1 hem comparat models clàssics; en D2 construïm el primer model profund i aprenem a detectar tres fallades habituals: divergència, sobreajustament i no reproductibilitat.

La decisió de disseny central és la relació entre **expressivitat** i **optimitzabilitat**. Afegir capes o neurones augmenta la capacitat del model, però també pot fer l'entrenament més sensible al *learning rate*, a la inicialització, al batch size o a la regularització. El model profund només és una millora si el pipeline pot demostrar que generalitza millor que el baseline D1.

Taula 2.1: Decisions operatives en el primer MLP de D2

Component	Decisió recomanada	Què heu de registrar
Activació	ReLU o GELU com a opció inicial robusta	Arquitectura i nombre de paràmetres
Inicialització	Inicialització per defecte de PyTorch, sense manipular fins tenir baseline	Seed i versió del codi
Optimitzador	AdamW com a punt de partida; SGD només com a comparativa didàctica	Learning rate, weight decay i batch size
Criteri d'aturada	Millor checkpoint segons validació, no última època	Mètrica de validació i època del checkpoint

Decisió d'arquitectura

AdamW com a punt de partida, però no com a dogma. En D2 proposem AdamW perquè acostuma a ser estable en xarxes petites i mitjanes i separa millor el paper del *weight decay*. L'alternativa, SGD amb momentum, pot ser més transparent per estudiar dinàmiques d'entrenament, però sol requerir més cura amb el *learning rate*.
Decisió operativa del laboratori: useu AdamW per al primer model profund reproduïble i reserveu SGD per a una comparativa didàctica. No canvieu alhora optimitzador, arquitectura, *batch size* i regularització si voleu interpretar la millora.
Evidència mínima: learning rate, weight decay, batch size, seed, mètrica de validació i època del millor checkpoint.

Un Perceptró Multicapa (MLP) és una composició de capes lineals i activacions no lineals. Per a una entrada $x \in \mathbb{R}^{d_{in}}$, una xarxa de L capes es pot escriure com:

$$h^{(l)} = \sigma\left(W^{(l)}h^{(l-1)} + b^{(l)}\right) \tag{2.1}$$

On $W^{(l)}$ i $b^{(l)}$ són pesos i biaixos, i $\sigma(\cdot)$ és una activació no lineal. Sense activació, la xarxa sencera es reduiria a una sola transformació lineal.

El bucle d'entrenament minimitza una pèrdua $\mathcal{L}$ actualitzant els paràmetres amb gradients:

$$\theta_{t+1} = \theta_t - \eta\nabla_\theta\mathcal{L}(\theta_t; X_i, y_i) \tag{2.2}$$

Aixi, si η és massa gran, el model pot divergir; si és massa petit, no aprèn; si el gradient s'esvaeix, les capes inicials no reben senyal útil.

Intuïció operativa: La burocràcia de l'error

Penseu en una xarxa com una cadena de decisions. El *forward pass* produeix una predicció, la funció de pèrdua mesura l'error, i el *backward pass* reparteix responsabilitat entre capes. Si la xarxa és massa profunda o està mal inicialitzada, el missatge de correcció pot arribar a les primeres capes massa feble o massa amplificat.
Criteri d'enginyeria: no cal memoritzar totes les derivades per superar D2, però sí saber diagnosticar si el problema és de dades, arquitectura, learning rate, batch size o regularització.

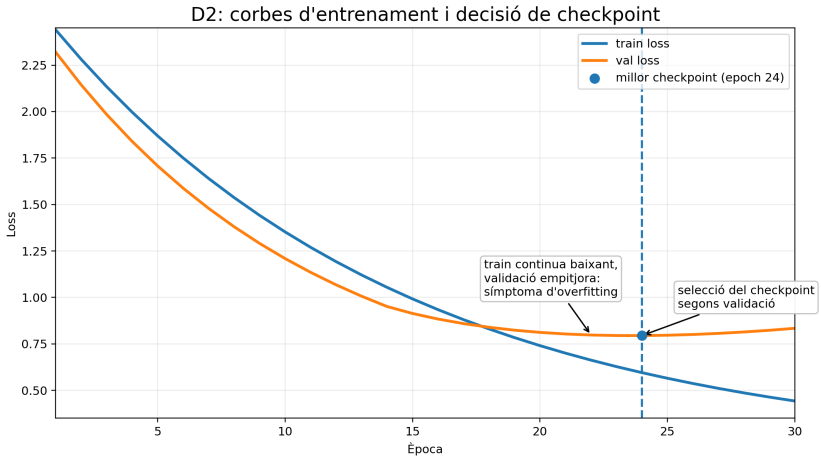

Figura 2.1: Corbes d'entrenament D2. La figura és una sortida del pipeline: mostra pèrdua i mètrica de validació per època i permet decidir si el model encara aprèn, si s'estanca o si comença a memoritzar.

Patró d'enginyeria

Abans d'augmentar profunditat, tanqueu el contracte d'entrenament: mateix split, mateix seed, mateixa mètrica principal i mateix mecanisme de logging. Si canvieu arquitectura i protocol alhora, no sabreu què ha causat la millora.

Laboratori de construcció: Primer entrenament profund amb tracking

Objectiu: entrenar un MLP i una CNN petita sobre el mateix dataset d'imatges simples, comparant el seu rendiment amb el baseline D1. L'objectiu no és obtenir el millor resultat possible, sinó deixar un entrenament traçable.
Sortides esperades: mètriques de train/validation/test, corba de pèrdua, checkpoint del millor model, fitxer de mètriques i run a Weights & Biases o JSON equivalent.

```
python -m lic_project.deep_train --config configs/d2_training.yaml
```

Listing 2.2: Execució del primer entrenament profund D2

Estratègia de validació

Test de no-fuita temporal abans d'entrenar.
D2 no pot entrenar cap model profund si el contracte de split de D1 no és vàlid. Abans de llançar `deep_train`, el pipeline ha de comprovar si existeix una columna temporal i si totes les dates de validació/test són posteriors a les dates d'entrenament.

Què hauria de fallar?

- que una fila de test tingui data anterior a una fila d'entrenament;
- que el preprocessament s'hagi ajustat amb dades posteriors a la data de tall;
- que un mateix usuari, pacient, sessió o dispositiu aparegui simultàniament en train i test si això invalida el cas d'ús;
- que el report no declari l'estratègia de split.

Lliçó per al D2. Una xarxa profunda no arregla un split contaminat; només amplifica la fuita.

Error típic

No compareu un MLP entrenat 50 èpoques amb una CNN entrenada 5 èpoques si no expliqueu el pressupost computacional. En D2, una comparació justa ha d'indicar epochs, batch size, nombre de paràmetres i temps aproximat d'entrenament.

Microcas o incidència de laboratori: El model millora només en train

Un grup substitueix el baseline D1 per una CNN i obté una millora espectacular en entrenament, però validació queda igual o pitjor. En revisar el run, es veu que el model té més capacitat però no hi ha regularització ni early stopping.
Acció d'enginyeria: comparar corbes train/validation, afegir weight decay, reduir capacitat o aplicar data augmentation abans de defensar el model.
Lliçó per al D2: una xarxa profunda només supera el baseline si millora la validació amb el mateix protocol experimental.

2.2 Les Xarxes Convolucionals (CNN)

Les CNN introdueixen un biaix inductiu útil per a imatges: localitat, pesos compartits i jerarquia espacial. En lloc de connectar cada píxel amb tots els altres, una convolució aplica el mateix filtre sobre tota la imatge. Això redueix paràmetres i fa que el model pugui reconèixer patrons encara que apareguin en posicions diferents.

La decisió d'enginyeria no és si una CNN és "millor" en abstracte, sinó si el seu biaix inductiu encaixa amb el dataset i amb el pressupost computacional. En D2, la CNN ha de competir amb el MLP amb el mateix split i la mateixa mètrica.

Intuïció operativa: convolució, pooling i camp receptiu

La convolució discreta 2D entre una imatge I i un filtre K es pot escriure com:

$$(I * K)(i, j) = \sum_m \sum_n I(i - m, j - n)K(m, n) \tag{2.3}$$

Per al laboratori, la lectura important és aquesta: un filtre és un detector local reutilitzable. Les primeres capes detecten vores o textures; capes posteriors combinen aquests patrons en formes més abstractes.

El camp receptiu després de L capes amb kernel k i stride 1 és, en una aproximació simple:

$$R_L = 1 + L(k - 1) \tag{2.4}$$

Això justifica per què una xarxa massa poc profunda pot no tenir context suficient, mentre que una massa profunda pot ser innecessària per a imatges petites.

Intuïció operativa: El Lego i l'arquitecte

Una CNN és com un sistema de peces reutilitzables: primer detecta vores, després formes, després objectes. El valor no és només que classifiqui millor, sinó que incorpora una hipòtesi sobre la naturalesa de les imatges: els patrons locals importen i es poden reutilitzar en diferents posicions.

Taula 2.2: Decisions d'enginyeria en una CNN petita

Decisió	Opció conservadora	Risc si s'exagera
Nombre de filtres	8–32 filtres inicials	Massa paràmetres per un dataset petit
Pooling	Reduir resolució gradualment	Pèrdua d'informació espacial fina
Data augmentation	Rotacions o soroll lleu si té sentit	Augmentacions incompatibles amb el domini
Batch size	Ajustat a memòria disponible	OOM o entrenament molt sorollós

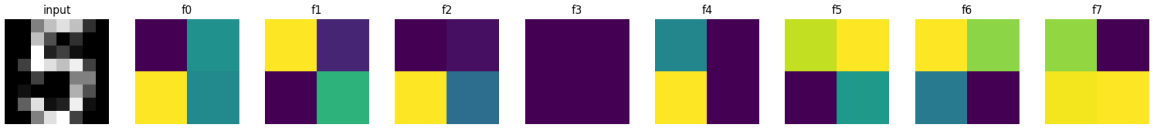

Figura 2.2: Mapes de característiques d'una CNN entrenada. La figura serveix per inspeccionar activacions, però la decisió de mantenir la CNN s'ha de basar en mètriques de validació i test.

Laboratori de construcció: CNN entrenada, checkpoint i mapes de característiques

Objectiu: entrenar una CNN petita, guardar el millor checkpoint i generar mapes de característiques només a partir d'aquest model entrenat. La visualització no pot sortir d'una xarxa amb pesos aleatoris si s'està usant per interpretar el model.

```
python -m lic_project.experiments.cnn_features \
  --config configs/d2_experiments.yaml \
  --checkpoint artifacts/models/d2_best_model.pt
```

Listing 2.3: Generació de mapes de característiques a partir d'un checkpoint

Per superar aquesta part, cal demostrar que el checkpoint existeix, que es pot recarregar i que la figura s'ha generat amb el mateix model que ha produït les mètriques reportades.

Microcas o incidència de laboratori: Mapa bonic, model dolent

Un grup presenta mapes d'activació visualment convincents, però el model té una F1 inferior al baseline D1. La visualització no compensa una avaluació pobra.
Acció d'enginyeria: mantenir la figura com a eina diagnòstica i rebutjar la CNN com a millora si no supera el baseline amb evidència quantitativa.

2.3 Mecanismes d'Atenció i Arquitectura Transformer

L'atenció resol un problema diferent de la convolució: connectar elements distants segons rellevància. En lloc d'aplicar un filtre local fix, el model calcula quins elements de la seqüència han d'influir en cada posició. Aquesta idea és essencial per entendre el mecanisme, el cost i com preparar el camí per al RAG del Capítol 4.

Per a cada element de la seqüència, definim tres vectors: *query*, *key* i *value*. L'atenció calcula

Taula 2.3: Components del Transformer llegits com a decisions d'enginyeria

Component	Funció	Risc operatiu
Multi-head attention	Atendre a diferents relacions alhora	Cost de memòria i interpretació difícil
Positional encoding	Injectar ordre en una arquitectura invariant a permutació	Errors si la longitud o format canvia
Layer normalization	Estabilitzar activacions per mostra	Diferències respecte BatchNorm en visió
Masking	Evitar mirar tokens futurs o tokens de padding	Bugs silenciosos en seqüències

similituds entre queries i keys, i retorna una mitjana ponderada dels values:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.5}$$

La part crítica d'enginyeria és que la matriu QK^T té mida $N \times N$. Això dona context global, però també un cost de memòria que creix quadràticament amb la longitud de seqüència.

Patró d'enginyeria

Abans d'usar un Transformer, estimeu el cost: longitud de seqüència, batch size, dimensió d'embedding i memòria disponible. Molts errors d'entrenament són errors de memòria.

Laboratori de construcció: inspecció d'atenció i cost quadràtic

Objectiu: implementar o usar una capa d'atenció petita, generar un mapa d'atenció i mesurar com creix la memòria teòrica quan augmenta N. La sortida esperada és una figura i una taula de cost, no una afirmació genèrica sobre Transformers.

```
python -m lic_project.experiments.attention_map \
  --config configs/d2_experiments.yaml
```

Listing 2.4: Execució de l'experiment d'atenció

No confongueu mapa d'atenció amb explicació completa. Un patró d'atenció pot ajudar a formular hipòtesis, però no substitueix una avaluació de rendiment, robustesa i biaix.

Microcas o incidència de laboratori: Seqüència massa llarga

Un grup intenta processar documents llargs sencers amb un Transformer sense estimar la memòria. El codi funciona amb frases petites, però falla amb errors OOM quan entra un document real.
Acció d'enginyeria: truncar, dividir en chunks, usar recuperació prèvia o models d'atenció eficient. Aquesta decisió connecta directament amb el RAG del Capítol 4.

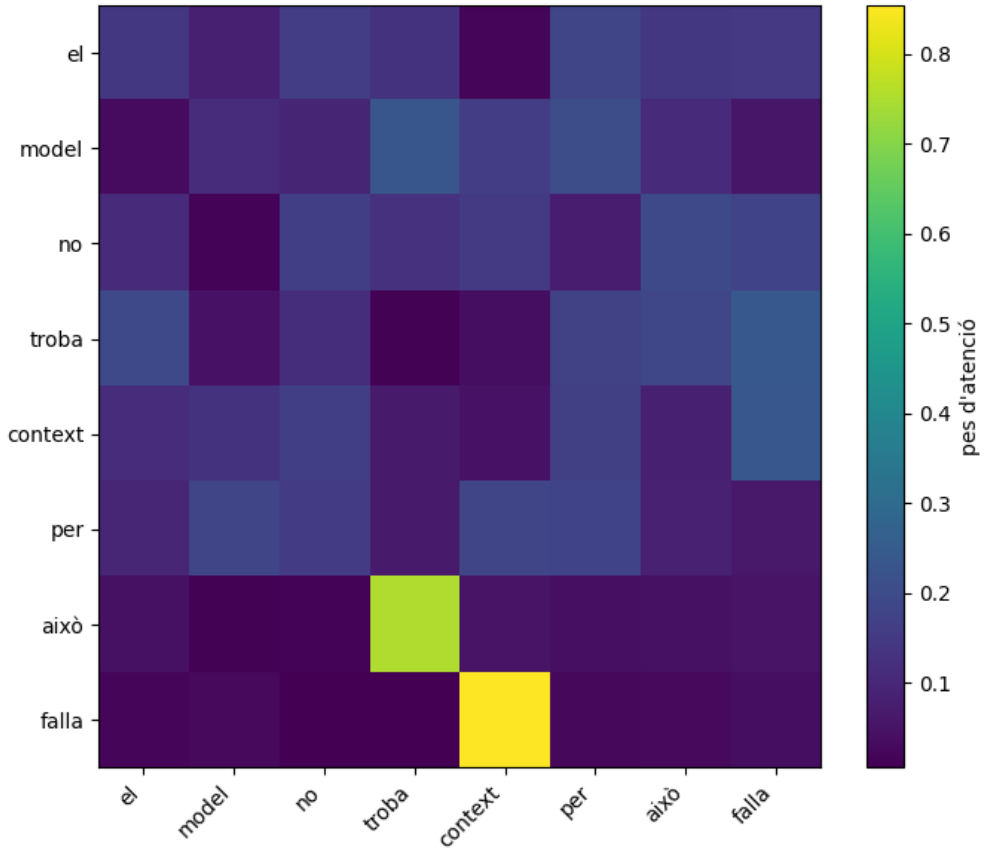

Figura 2.3: Mapa d'atenció generat pel pipeline. La figura ha d'anar acompanyada d'una lectura crítica: l'atenció pot suggerir relacions, però no és per si sola una explicació causal.

2.4 Transfer Learning i Models Preentrenats

Entrenar des de zero és car i sovint innecessari. En una assignatura de grau, el patró professional no és entrenar un ResNet o un Transformer des de zero, sinó saber quan usar un model preentrenat, quan congelar-lo, quan fer *fine-tuning* i com comprovar si la representació transferida és útil.

Formalment, un model preentrenat f_{θ_S} ha après una representació en un domini font $\mathcal{D}_S$. En el nostre domini objectiu $\mathcal{D}_T$, podem:

1. congelar el cos del model i entrenar només un capçal lineal;
2. descongelar parcialment algunes capes;
3. fer fine-tuning complet si hi ha prou dades i pressupost.

Intuïció operativa: El món de la música i l'instrument nou

Un model preentrenat és com un perfil musical que ja sap harmonia, ritme i lectura. Adaptar-lo a una tasca nova és més ràpid que ensenyar música des de zero. Però si l'instrument és massa diferent, el coneixement previ pot no transferir bé.

Decisió d'arquitectura

Linear probing abans de fine-tuning. El *linear probing* és una prova de viabilitat barata: congela el model base i avalua si les representacions ja separen el problema. Si falla

Taula 2.4: Estratègies d'adaptació de models preentrenats

Estratègia	Quan usar-la	Què registrar
Linear probing	Primer test de viabilitat	Mètrica del capçal, model base i features
Fine-tuning parcial	Dataset moderat i domini semblant	Capes descongelades i learning rates
Fine-tuning complet	Dataset gran o domini molt específic	Cost, overfitting i validació robusta

Decisió d'arquitectura *(continuació)*

clarament, el fine-tuning pot ser una mala inversió: consumeix més temps, pot generar overfitting i pot ocultar que el problema real era el split o la mètrica.

Decisió recomanada en D2: compareu baseline D1, model entrenat des de zero i linear probe. Només proposeu fine-tuning si el probe mostra senyal útil i si el cost addicional es pot defensar.

Laboratori de construcció: Prova de viabilitat amb linear probing

Objectiu: fer una prova de viabilitat abans d'un fine-tuning costós. El lliurable és una comparació entre entrenar des de zero i usar una representació congelada. Si el linear probe no supera el baseline, el model base no és una bona representació per al vostre domini.

```
python -m lic_project.experiments.transfer_probe \
  --config configs/d2_experiments.yaml
```

Listing 2.5: Execució del linear probe

La conclusió ha d'indicar si faríeu fine-tuning, si mantindríeu el baseline o si canviaríeu el model base. No s'accepta com a resposta "usaré un foundation model" sense evidència de transferència.

2.5 Regularització, optimitzadors i estabilitat de l'entrenament

Regularitzar és controlar la llibertat del model. En D2 no tractem la regularització com una fórmula decorativa, sinó com una decisió observable: canvia la bretxa entre train i validation? millora calibració? estabilitza el checkpoint? redueix la variança entre seeds?

Afegim una penalització a la pèrdua per evitar pesos massa grans:

$$\tilde{\mathcal{L}}(\theta) = \mathcal{L}(\theta) + \lambda\Omega(\theta) \tag{2.6}$$

En Deep Learning modern, *AdamW* desacobla el *weight decay* de l'actualització adaptativa i acostuma a ser un bon punt de partida. Dropout, data augmentation i early stopping són mecanismes complementaris, però s'han de justificar amb mètriques, no per tradició.

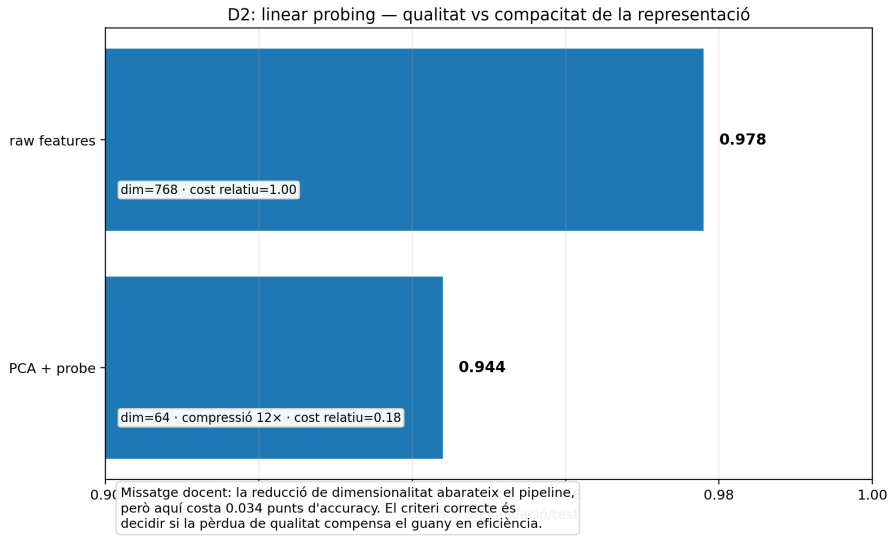

Figura 2.4: Comparació de representacions per a *linear probing*. La figura mostra el compromís entre qualitat i eficiència: treballar amb la representació crua conserva més senyal i obté millor *accuracy*, mentre que la versió comprimida amb PCA redueix dimensionalitat i cost del pipeline a canvi d'una pèrdua moderada de rendiment. El resultat avaluable no és la figura per si sola, sinó la decisió justificada amb mètriques de validació i cost computacional.

Taula 2.5: Regularització llegida com a acció de pipeline

Tècnica	Què intenta corregir	Com ho verifiqueu
Weight decay	Pesos massa grans i frontera massa complexa	Menor gap train/validation
Dropout	Co-adaptació i memorització	Validació més estable
Early stopping	Entrenar més enllà del punt útil	Millor checkpoint segons validació
Label smoothing	Excés de confiança	Calibració o pèrdua de validació

Laboratori de construcció: Comparació de regularització amb runs

Objectiu: executar almenys dues configuracions: una sense regularització i una amb weight decay, dropout o early stopping. La decisió ha de quedar justificada amb mètriques, corbes i checkpoint.

```
python -m lic_project.experiments.regularization \
  --config configs/d2_experiments.yaml
```

Listing 2.6: Comparació de regularització

Checklist de verificació

- Les dues configuracions comparteixen split i seed?
- Heu registrat weight decay, dropout, learning rate i epochs?
- La millora és en validació o només en train?
- El checkpoint seleccionat correspon a la millor validació?

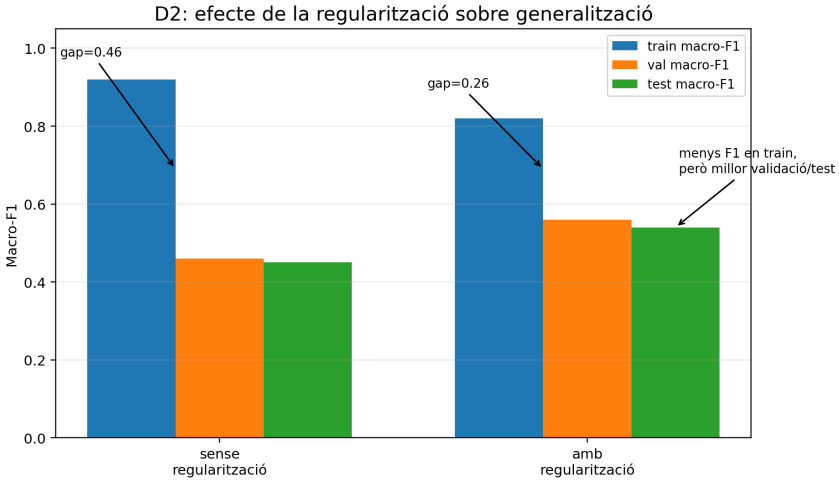

Figura 2.5: Efecte de la regularització sobre la generalització en D2. Sense regularització, el model obté una macro-F1 alta en entrenament però baixa en validació i test, indicant sobreajustament. Amb regularització, el rendiment en train disminueix lleugerament, però milloren validació i test i es redueix el gap de generalització. La decisió no s'ha de basar en la barra més alta, sinó en el millor compromís entre rendiment, estabilitat i capacitat de generalitzar.

2.6 Microcas de Recerca: Segmentació amb U-Net com a pràctica satèl · lit

La segmentació no ha de convertir el capítol en un curs de visió mèdica. El seu valor dins LIC és mostrar una altra família de sortides: no sempre prediem una classe; de vegades produïm una màscara. Això obliga a canviar mètrica, loss, arquitectura i criteri d'acceptació.

Intuïció operativa: encoder-decoder, skip connections i Dice

Una U-Net combina un encoder que captura context amb un decoder que recupera resolució. Les skip connections transfereixen detall espacial:

$$x_{dec}^{(i)} = \text{UpSample}(x_{dec}^{(i-1)}) \oplus x_{enc}^{(i)} \tag{2.7}$$

En segmentació desbalancejada, l'accuracy és perillosa. La Dice loss mesura solapament entre predicció i màscara:

$$\mathcal{L}_{Dice} = 1 - \frac{2\sum_i p_i g_i + \epsilon}{\sum_i p_i + \sum_i g_i + \epsilon} \tag{2.8}$$

Patró d'enginyeria

Quan la sortida és una màscara, el contracte de qualitat canvia: cal mirar Dice, falsos positius espacials, falsos negatius i coherència de contorn. Una accuracy alta pot significar simplement que el model prediu fons.

Laboratori de construcció: Mini-U-Net amb mètrica Dice

Objectiu: entrenar una mini-U-Net sobre dades sintètiques petites, calcular Dice i guardar una figura comparativa. Aquesta pràctica és satèl · lit: mostra com adaptar el pipeline quan la sortida no és una classe sinó una imatge.

```
python -m lic_project.experiments.segmentation \
  --config configs/d2_experiments.yaml
```

Listing 2.7: Execució de la pràctica de segmentació

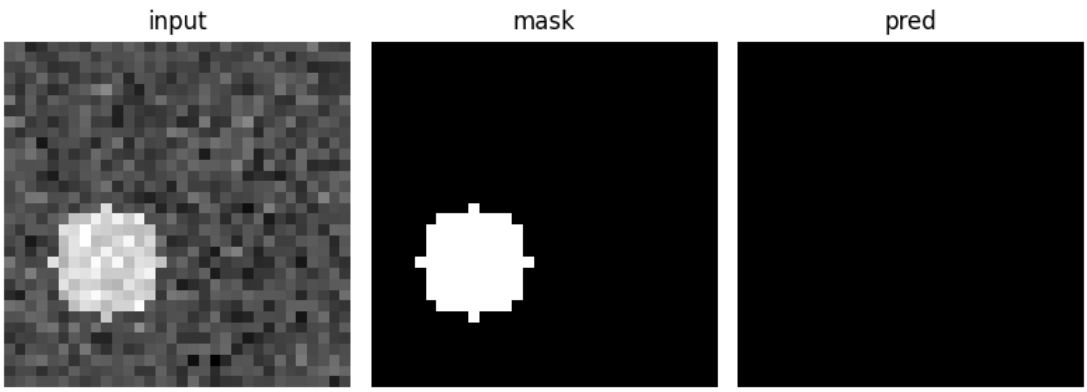

Figura 2.6: Segmentació sintètica amb mini-U-Net. El panell visual s'ha d'acompanyar de Dice i d'una decisió sobre si el resultat és acceptable.

Laboratori de construcció: Mini-U-Net amb mètrica Dice *(continuació)*

No feu servir accuracy com a mètrica principal en segmentació amb molt fons. Un model que prediu tot fons pot tenir una accuracy aparentment alta i ser inútil.

2.7 Conclusions: escala, pressupost i transició cap a verificació

Aquest capítol ha transformat el baseline D1 en un entrenament profund D2. El resultat no és "hem provat una CNN", sinó un pipeline amb model, configuració, logging, checkpoint, tests i informe. Aquesta és la frontera entre jugar amb Deep Learning i fer enginyeria de Deep Learning.

Les *scaling laws* són útils com a lectura avançada: indiquen que més dades, més paràmetres i més computació poden millorar el rendiment de manera previsible. Però en una assignatura de grau el missatge operatiu és més modest i més important: abans d'escalar, cal poder reproduir; abans de fine-tuning, cal fer un probe; abans d'afegir capes, cal mirar validació; abans de confiar en una predicció, cal auditar-la.

Una forma simplificada de les lleis d'escalat expressa que la pèrdua disminueix quan augmentem paràmetres N i dades D:

$$\mathcal{L}(N, D) \approx \left(\frac{N_c}{N}\right)^{\alpha_N} + \left(\frac{D_c}{D}\right)^{\alpha_D} \tag{2.9}$$

En D2 no farem entrenaments massius. El que sí farem és simular el principi: comparar errors quan creix la mida del dataset o la capacitat del model, i discutir quin recurs és el coll d'ampolla.

Pregunta de defensa

Defensa tècnica del D2. Abans de tancar el capítol, el grup ha de poder respondre: per què el model profund escollit és millor que el baseline D1 i no només més complex? què necessita un altre grup per recarregar el checkpoint? quin hiperparàmetre tocaríeu primer davant un error OOM?
Una resposta acceptable ha de citar evidències: mètriques de validació, configuració, run, checkpoint i limitacions.

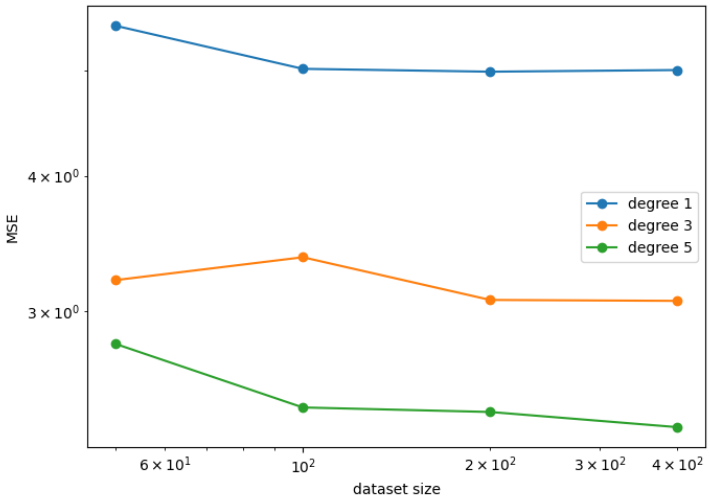

Figura 2.7: Simulació de scaling laws. La figura serveix per discutir pressupost, no per justificar entrenaments massius dins l'assignatura.

Microcas o incidència de laboratori: Xarxa profunda que no es pot defensar

Un model profund supera lleugerament el baseline, però no hi ha checkpoint, no es coneix el seed, no s'han registrat hiperparàmetres i ningú pot reproduir el resultat. Tècnicament pot semblar millor, però no és defensable com a lliurable LIC.

Lliçó final del capítol: D2 no busca el model més gran, sinó el primer entrenament profund que es pugui executar, auditar i comparar.

Troubleshooting i depuració

Incidència de laboratori: la pèrdua no baixa. Símptoma: la *loss* queda gairebé constant durant diverses èpoques o apareixen valors `nan`. Abans de canviar arquitectura, cal depurar el contracte d'entrenament.

Causes probables: learning rate massa alt, dades sense normalitzar, etiquetes mal codificades, funció de pèrdua incoherent amb la sortida del model o gradients que exploten.

Com detectar-ho: registreu `train/loss`, `val/loss`, `learning_rate` i una prova d'overfit sobre 16 exemples. Si el model no pot memoritzar un mini-batch controlat, el problema no és de generalització sinó d'implementació.

Acció d'enginyeria: reduir learning rate, validar formes de tensors, comprovar etiquetes i repetir l'experiment amb el mateix seed.

Consell d'enginyeria

Checkpoint defensable. Un checkpoint no és només un fitxer `.pt`. Per ser defensable ha de conservar arquitectura, dimensions d'entrada, nombre de classes, preprocessament, seed, mètrica usada per seleccionar-lo i època de validació.

El patró recomanat és guardar dos artefactes: `artifacts/models/d2_best_model.pt` i `artifacts/models/d2_best_model.metadata.json`. Si no es pot recarregar el model sense llegir el notebook original, el checkpoint no és un artefacte d'enginyeria.

Costos i eficiència

Pressupost de memòria de l'atenció. Si dupliqueu la longitud de seqüència N, la matriu d'atenció passa de N^2 a $(2N)^2 = 4N^2$. Amb *batch size*, múltiples caps i diverses capes, aquest creixement pot provocar errors OOM encara que el model sembli petit.

Costos i eficiència *(continuació)*

Regla pràctica per a D2: abans d'augmentar N, reduïu el *batch size*, mesureu memòria i registreu el cost. Aquesta és la motivació operativa del RAG del Capítol 4: quan el document és massa llarg, es recupera context rellevant en lloc d'enviar-ho tot al model.

Tancament del D2

Artefacte lliurable

D2 — Entrenament profund reproduïble. Al final del capítol heu de lliurar:

- execució documentada de `python -m lic_project.deep_train --config configs/d2_training.yaml`;
- checkpoint del millor model i configuració associada;
- mètriques de train, validation i test;
- corbes d'entrenament i comparació d'almenys dues configuracions;
- registre a Weights & Biases o fitxer JSON auditable;
- tests mínims executats amb `pytest`;
- informe D2 amb decisió: quin model profund es manté, quin baseline supera i quin risc queda obert.

Checklist de verificació

Abans de donar D2 per acabat, comproveu:

- el model s'entrena amb una comanda documentada;
- les mètriques i el checkpoint no depenen d'un notebook;
- el seed, hiperparàmetres i versió del codi queden registrats;
- hi ha comparació amb el baseline D1 o amb una arquitectura més simple;
- els tests passen;
- l'informe explica una decisió d'enginyeria i no només mostra una figura.

Criteris d'acceptació

Criteri	Punts
Pipeline d'entrenament executable i sense dependència d'un notebook	2.0
Model PyTorch encapsulat i checkpoint recarregable	1.5
Configuració externa, seed i reproductibilitat	1.5
Tracking amb Weights & Biases o registre JSON auditable	1.5
Comparació de configuracions i interpretació de corbes	1.5
Tests mínims de dades, model i entrenament	1.0
Informe tècnic D2 i decisió justificada	1.0

Capítol 3

Verificació de Models: Robustesa, Explicabilitat i Equitat Algorísmica

Els capítols 1 i 2 han construït els dos primers contractes del projecte transversal: D1 ha establert un baseline reproduïble i D2 ha afegit un entrenament profund amb configuració, tracking, checkpoint i tests. En aquest capítol el repte canvia: ja no n'hi ha prou amb entrenar un model que millori una mètrica global. Ara cal demostrar que el model és auditable, que els errors són visibles, que les diferències entre subgrups es mesuren i que la robustesa s'ha comprovat abans de defensar el sistema.

El producte del capítol no és una explicació visual bonica ni una discussió abstracta sobre ètica computacional. El producte és el lliurable D3: un paquet d'auditoria executable que llegeix un model o un conjunt de prediccions, calcula mètriques globals i desagregades, genera explicacions locals i globals, executa una prova de robustesa i produeix un informe tècnic amb criteris d'acceptació. Les figures són útils, però continuen sent sortides secundàries del pipeline.

Repte d'enginyeria

Repte del capítol. Convertir el model D2 en un sistema auditable: explicabilitat, equitat, robustesa i informe de verificació.
Entregable associat: D3.
Artefactes mínims:

- un script d'auditoria executable amb configuració externa;
- mètriques globals i mètriques per subgrup;
- càlcul de *disparate impact*, diferència de paritat demogràfica i diferència d'oportunitat;
- una explicació global i una explicació local d'una decisió crítica;
- una prova de robustesa davant pertorbacions o canvis de distribució;
- un fitxer `reports/audit/d3_audit_metrics.json`;
- un informe `reports/d3_report.md`;
- tests mínims de mètriques, explicacions i contracte d'auditoria.

Laboratori de construcció:

L'auditoria no és un apartat narratiu afegit al final del projecte. En aquest llibre, una auditoria és codi executable. El professorat ha de poder executar el pipeline, revisar els llindars configurats, reproduir les figures i veure si el model passa o no els criteris definits. Una explicació generada manualment en un notebook no és una explicació auditable. En D3, cada gràfic i cada conclusió ha de sortir d'un fitxer de mètriques o d'un script reproduïble. Si una figura no es pot regenerar, no compta com a evidència.

Artefacte lliurable

Contracte tècnic D3. En acabar el capítol, el professorat ha de poder executar:

```
python -m lic_project.audit --config configs/d3_audit.yaml
python -m lic_project.report_d3 --audit-dir reports/audit --output reports/d3_report.md
pytest
```

Si alguna d'aquestes comandes falla, el capítol no s'ha completat com a laboratori d'enginyeria.

```
lic_project/
|-- configs/
|   |-- d1_baseline.yaml
|   |-- d2_training.yaml
|   `-- d3_audit.yaml
|-- artifacts/
|   `-- models/
|-- reports/
|   |-- audit/
|   |-- figures/
|   |-- metrics/
|   `-- d3_report.md
|-- src/
|   `-- lic_project/
|       |-- audit_data.py
|       |-- audit_metrics.py
|       |-- explainability.py
|       |-- robustness.py
|       |-- audit.py
|       `-- report_d3.py
`-- tests/
    |-- test_audit_metrics.py
    |-- test_explainability.py
    `-- test_audit_contract.py
```

Listing 3.1: Estructura incremental D3 sobre el repositori D1–D2

3.1 De la mètrica global al contracte d'auditoria

Un model pot superar D2 i continuar sent inadequat per a un sistema real. Pot tenir bona F1 global i fallar de manera sistemàtica en un subgrup. Pot ser molt precís però impossible d'explicar. Pot funcionar en el conjunt de test i col · lapsar quan una característica es desplaça lleugerament. D3 introdueix una nova unitat de qualitat: el **contracte d'auditoria**, que defineix què s'ha de mesurar, quins llindars no es poden vulnerar i quina evidència s'ha de guardar. No substitueix el judici humà, però evita que l'avaluació depengui d'una lectura informal de gràfics.

Decisió d'arquitectura

Llindar únic o llindars per segment? Un llindar global és fàcil de defensar i reproduir, però pot distribuir l'error de manera desigual. Els llindars per segment poden reduir un tipus d'error en un grup, però també introdueixen complexitat normativa i necessitat de governança.

Decisió D3: comenceu amb un llindar global auditat per subgrups. Només proposeu llindars diferenciats si l'informe mostra una degradació clara, si la calibració per grup és fiable i si la decisió és justificable dins del domini.

Taula 3.1: Del model D2 al contracte d'auditoria D3

Element	Pregunta d'auditoria	Artefacte D3
Rendiment global	El model supera el baseline?	`global_metrics.json`
Subgrups	L'error es reparteix de manera desigual?	`group_metrics.json`
Explicabilitat	Quines variables dominen la decisió?	figura i taula d'importàncies
Robustesa	La decisió canvia amb soroll raonable?	corba de degradació i taxa de flips
Governança	El model passa els llindars?	`audit_summary.json` i informe D3

Per a cada instància x_i, el model produeix una puntuació $s_i = f(x_i)$ i una decisió $\hat{y}_i = \mathbb{I}[s_i \geq \tau]$. L'auditoria afegeix tres capes:

- **resultat global**: mètriques com F1, ROC-AUC o PR-AUC;
- **resultat desagregat**: les mateixes mètriques per grup, segment o condició operativa;
- **evidència**: explicació local/global, robustesa davant pertorbacions i registre de llindars.

La idea matemàtica mínima és que una mètrica global M pot amagar diferències importants:

$$M_{global} \neq \{M_{A=0}, M_{A=1}, M_{segment\_1}, \ldots\} \tag{3.1}$$

Per tant, el pipeline D3 no pregunta només "quin és el rendiment?", sinó "qui suporta l'error, quan falla i amb quina evidència ho podem veure?".

Patró d'enginyeria

Abans de discutir si un model és "just" o "robust", fixeu un contracte observable: mètrica principal, subgrups, llindars d'alerta, tipus d'explicació i prova de robustesa. Sense aquest contracte, l'auditoria és opinió.

Laboratori de construcció:

Objectiu: executar una auditoria completa sobre un model entrenat o sobre un model sintètic de crèdit. El resultat ha de ser un directori `reports/audit/` amb mètriques globals, mètriques per subgrup, figures i resum de criteris d'acceptació.

```
python -m lic_project.audit --config configs/d3_audit.yaml
```

Listing 3.2: Execució de l'auditoria D3

Per superar aquesta primera part, cal que el pipeline generi `d3_audit_metrics.json`, `d3_audit_summary.json` i almenys tres figures: equitat, explicabilitat i robustesa. Les figures no poden estar dibuixades a mà.

Microcas o incidència de laboratori: Model bo, informe pobre

Un grup obté una F1 alta i presenta una figura d'importàncies, però no mostra mètriques per subgrup ni llindars d'auditoria. El model pot ser prometedor, però no és defensable com a sistema verificat.
Acció d'enginyeria: bloquejar l'acceptació fins que l'auditoria generi mètriques desagregades, criteris d'alerta i informe reproduïble.
Lliçó per al D3: una bona mètrica global no és una auditoria.

3.2 Explicabilitat: de les importàncies globals a l'explicació local

L'explicabilitat no consisteix a convertir una xarxa neuronal en una frase tranquil · litzadora. Consisteix a produir evidència útil per depurar, comunicar i auditar decisions. En grau, el nucli no és demostrar tota la teoria de SHAP o LIME, sinó saber quan una explicació és global, quan és local, quina hipòtesi fa i quins riscos té.

Una explicació global intenta respondre: "quines variables influeixen més en el model, de mitjana?". Una explicació local intenta respondre: "per què aquesta instància concreta ha rebut aquesta decisió?". Les dues poden divergir.

En D3 treballarem amb dues idees suficients:

- **importància per permutació**: si desordenar una variable degrada la mètrica, aquella variable aporta informació al model;
- **subrogat local**: al voltant d'una instància x, ajustem un model interpretable g que aproxima f en un veïnatge.

Una formulació compacta del subrogat local és:

$$g^* = \arg\min_{g \in \mathcal{G}} \sum_{z \in \mathcal{N}(x)} \pi_x(z) \left(f(z) - g(z)\right)^2 + \lambda \Omega(g) \tag{3.2}$$

On $\mathcal{N}(x)$ és un conjunt de pertorbacions al voltant de x, π_x pondera la proximitat i $\Omega(g)$ penalitza explicacions massa complexes. La lectura pràctica és directa: si el subrogat local no és fidel, l'explicació no s'ha d'usar per justificar la decisió.

Error típic

No confongueu explicabilitat amb veritat causal. Una variable pot ser important perquè és un proxy d'una altra variable, perquè el dataset està esbiaixat o perquè el model ha explotat una correlació espúria. L'explicació obre una pregunta; no tanca l'auditoria.

Troubleshooting i depuració

Explicació estable, model inestable. Un grup mostra una explicació local convincent, però en repetir l'auditoria amb un altre seed canvien les variables dominants. El problema no és només l'explicador: el model o el subrogat local no són prou estables.
Com detectar-ho: repetiu l'explicació per dos seeds o dues mostres properes i registreu la fidelitat local. Si la fidelitat baixa o les variables canvien radicalment, l'explicació s'ha de presentar com a diagnòstic provisional, no com a justificació final.

Taula 3.2: Tipus d'explicació i riscos operatius

Mètode	Ús recomanat	Risc
Permutació global	Depurar dependències del model	Pot ocultar efectes locals o interaccions
Subrogat local	Explicar una decisió concreta	Pot ser poc fidel si el veïnatge és mal definit
SHAP	Repartir contribucions amb una base de teoria de jocs	Cost computacional i dependència del background
Mapes d'atenció	Inspeccionar relacions en seqüències	No equivalen automàticament a causalitat

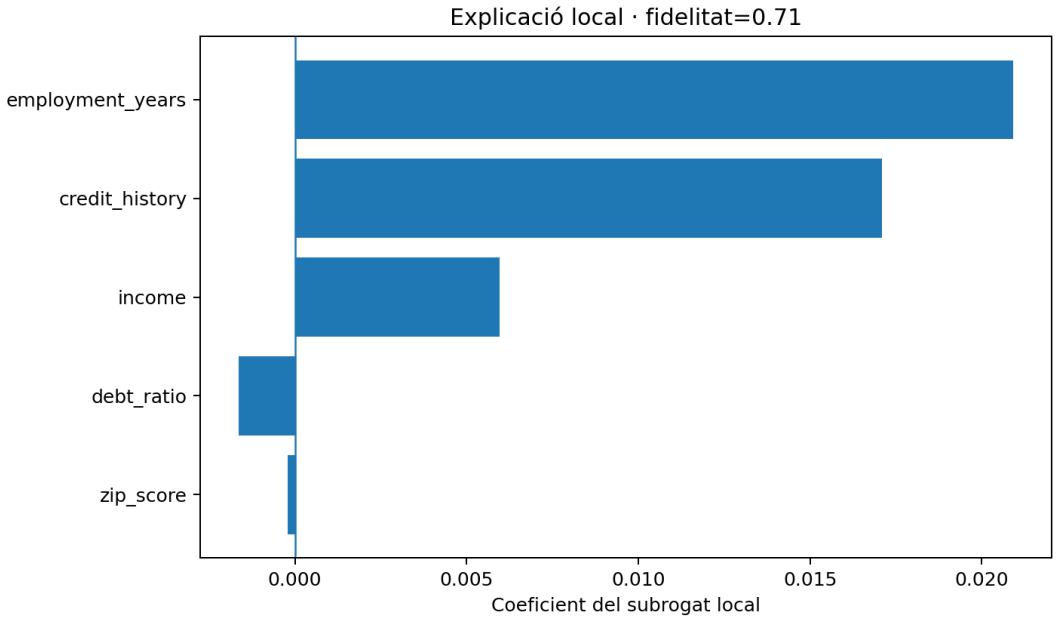

Figura 3.1: Explicació local generada pel pipeline D3. Les barres indiquen quines variables empenyen la decisió cap a una classe o cap a l'altra. Aquesta figura només és vàlida si el fitxer de mètriques indica també la fidelitat aproximada del subrogat local.

Laboratori de construcció:

Objectiu: generar una explicació global i una explicació local des del mateix model auditat. L'alumnat ha de seleccionar una decisió crítica, explicar quines variables l'han empès cap a l'aprovació o denegació i indicar si l'explicació sembla coherent amb el domini.

```
python -m lic_project.audit --config configs/d3_audit.yaml
```

Listing 3.3: Execució de l'explicabilitat dins l'auditoria D3

Una explicació D3 ha de respondre: quina instància s'explica, quin model s'ha usat, quines variables dominen, quina fidelitat té el subrogat i quina limitació cal declarar.

Taula 3.3: Mètriques d'equitat llegides com a decisions d'enginyeria

Mètrica	Pregunta que respon	Risc si s'usa sola
Paritat demogràfica	Es concedeixen decisions positives a ritmes semblants?	Pot ignorar diferències reals de prevalença
Disparate impact	Hi ha un grup amb una taxa molt inferior?	No explica la causa ni els errors condicionats a Y
Equal opportunity	Els positius reals reben oportunitat semblant?	No controla falsos positius
Equalized odds	TPR i FPR són semblants per grup?	Pot entrar en tensió amb la calibració

Microcas o incidència de laboratori: Incidència de laboratori: explicació que revela un proxy

Un model de crèdit no rep l'atribut protegit com a entrada, però l'explicació mostra que `zip_score` i `income` dominen moltes denegacions. Això no prova discriminació per si sol, però indica que el model pot estar usant proxies socioeconòmics.
Acció d'enginyeria: afegir l'auditoria per subgrups, revisar la generació de l'etiqueta i comparar un model sense proxies amb el model original.

3.3 Equitat algorísmica: mètriques per subgrup i llindars d'alerta

L'equitat algorísmica es presenta com un conjunt de preguntes mesurables. En D3 no afirmem si un model és "just", sinó comprovem si vulnera llindars explícits en unes mètriques definides.

Intuïció operativa: paritat, oportunitat i impacte dispar

Siguin $A \in \{0, 1\}$ un atribut de grup, $\hat{Y}$ la decisió i Y l'etiqueta. Tres mètriques bàsiques són:

$$\text{SelectionRate}_a = P(\hat{Y} = 1 \mid A = a) \tag{3.3}$$

$$\text{DisparateImpact} = \frac{\min_a P(\hat{Y} = 1 \mid A = a)}{\max_a P(\hat{Y} = 1 \mid A = a)} \tag{3.4}$$

$$\Delta_{EO} = \left| P(\hat{Y} = 1 \mid Y = 1, A = 0) - P(\hat{Y} = 1 \mid Y = 1, A = 1) \right| \tag{3.5}$$

La primera mesura quants casos reben una decisió positiva per grup. La segona compara taxes de selecció. La tercera mira si els positius reals tenen la mateixa probabilitat de ser detectats en cada grup.

Patró d'enginyeria

No existeix una mètrica d'equitat universal. La paritat demogràfica pot ser inadequada si els grups tenen prevalències diferents; la igualtat d'oportunitats pot ser més rellevant quan el cost principal és deixar fora positius reals. El que no és acceptable és no mesurar res.

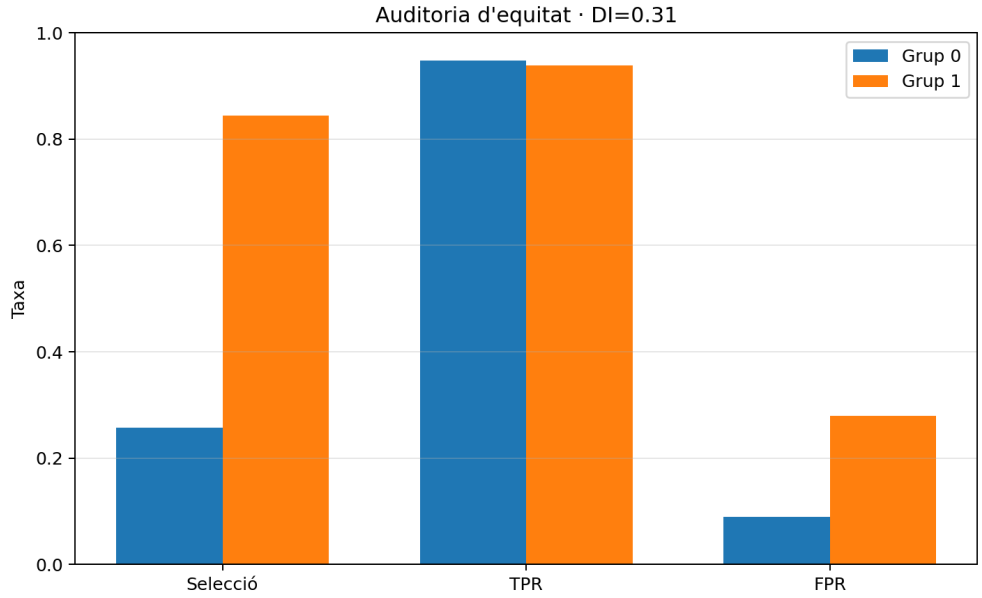

Figura 3.2: Auditoria d'equitat D3. La figura compara taxes de selecció, TPR i FPR per grup. La decisió final depèn dels llindars configurats, no d'una inspecció visual informal.

Estratègia de validació

Tests d'auditoria per subgrup. El codi D3 ha d'incloure tests sobre un dataset mínim on es coneix el resultat esperat de paritat, disparate impact i equal opportunity. Aquest test protegeix contra errors silenciosos de definició: canviar numerador i denominador pot fer que una auditoria sembli correcta quan no ho és.
Què hauria de fallar? El test hauria de fallar si un grup sense positius reals no es gestiona explícitament, si hi ha divisions per zero o si una mètrica retorna `nan` sense advertiment.

Laboratori de construcció:

Objectiu: calcular mètriques globals i per grup, generar una figura comparativa i decidir si el model passa els llindars configurats. El lliurable no és dir "hi ha biaix", sinó indicar quina mètrica s'ha vulnerat i quina acció tècnica es proposa.

```
cat reports/audit/d3_audit_summary.json
```

Listing 3.4: Consulta del resum d'equitat generat pel pipeline

Eliminar l'atribut protegit del dataset no elimina necessàriament el biaix. Altres variables poden actuar com a proxies. En D3 s'ha d'auditar el resultat, no només la llista de columnes d'entrada.

Microcas o incidència de laboratori: Incidència de laboratori: llindar únic, efecte desigual

Un model usa el mateix llindar per a tots els grups. La mètrica global és bona, però l'auditoria mostra que el grup minoritari té una taxa de falsos negatius molt més alta.
Acció d'enginyeria: revisar calibració per grup, qualitat de l'etiqueta, variables proxy i criteris de decisió abans de desplegar.
Lliçó per al D3: el problema no és només el model; pot ser el llindar, la dada, l'etiqueta o el procés de decisió.

Taula 3.4: Lectura operativa de les proves de robustesa

Prova	Què simula	Acció si falla
Soroll gaussià petit	Variació normal de mesures	Regularitzar o revisar preprocessament
Variables absents	Fallades d'ingesta o sensors	Imputació robusta i tests de contracte
Canvi de distribució	Drift o domini nou	Monitoratge i reentrenament
Baixa confiança	Decisió incerta	Abstenció o revisió humana

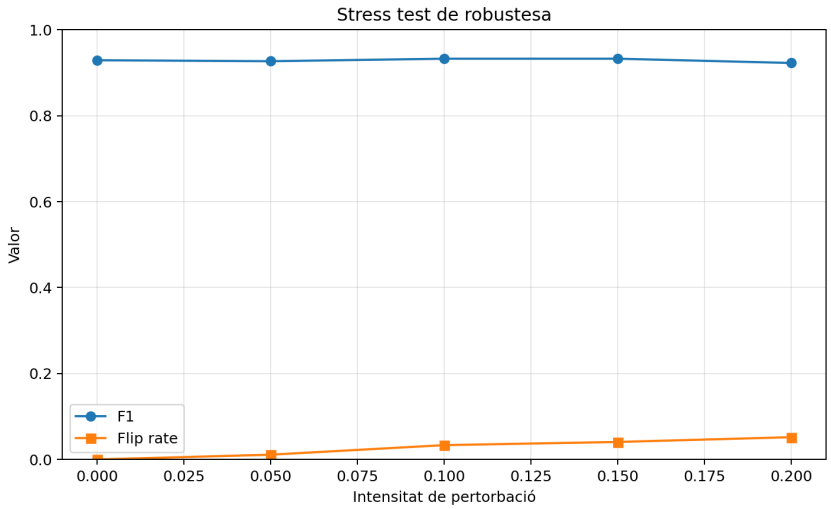

Figura 3.3: Prova de robustesa D3. La figura mostra com es degrada la mètrica i com augmenta la taxa de canvis de decisió quan creix la pertorbació.

3.4 Robustesa i incertesa: quan la decisió deixa de ser fiable

La robustesa mesura fins a quin punt el model manté el comportament quan l'entrada canvia dins d'un marge raonable. En un laboratori de grau no cal convertir aquesta secció en una derivació completa d'atacs adversarials. Cal que l'alumnat implementi una prova de tensió i decideixi quan el model hauria d'abstenir-se o escalar la decisió a revisió humana.

Una prova de robustesa senzilla consisteix a afegir una pertorbació δ a l'entrada i mesurar la taxa de canvis de decisió:

$$\text{FlipRate}(\sigma) = P\left(f(x) \neq f(x + \delta_\sigma)\right) \tag{3.6}$$

Si una petita pertorbació canvia moltes decisions, el model pot ser massa fràgil. També podem mesurar la caiguda de F1 o AUC respecte al conjunt sense soroll.

Laboratori de construcció:

Objectiu: executar una prova de robustesa amb pertorbacions controlades, calcular la degradació de mètriques i proposar un criteri d'abstenció o revisió humana.
Un model robust no és un model que mai falla, sinó que ho fa de manera visible, amb criteris d'alerta i amb una política d'escalat quan la confiança o l'estabilitat no són suficients.

Laboratori de construcció: (continuació)

El stress test ha d'indicar: tipus de pertorbació, intensitat, mètrica base, caiguda màxima acceptada, taxa de flips i decisió operativa: acceptar, regularitzar, monitorar o escalar.

Microcas o incidència de laboratori: Incidència de laboratori: predicció molt segura en dades fora de domini

Un model assigna probabilitats molt altes a entrades que no s'assemblen al dataset d'entrenament. La softmax dona confiança, però no sap dir "no ho sé".
Acció d'enginyeria: afegir proves OOD, calibració, llindars d'abstenció i revisió humana quan la confiança no sigui fiable.

3.5 Audit gate: convertir l'ètica en una prova automàtica

El punt central del capítol és passar de la discussió a l'automatització. Si una restricció és important, ha d'aparèixer com a prova, llindar o criteri d'acceptació. Un *audit gate* és una etapa del pipeline que decideix si un model pot avançar cap a integració, demostració o desplegament.

Un *audit gate* no substitueix una revisió humana, però evita que un model clarament problemàtic passi per inèrcia. El fitxer de configuració defineix llindars com:

- `min_disparate_impact`;
- `max_demographic_parity_diff`;
- `max_equal_opportunity_diff`;
- `max_robustness_drop`;
- `min_local_fidelity`.

El resum D3 ha d'indicar explícitament `PASS`, `WARN` o `FAIL` i quina acció cal fer.

Pregunta de defensa

Defensa tècnica del D3. El grup ha de defensar una decisió PASS, WARN o FAIL. Una bona defensa explica quina mètrica bloqueja el model, quin llindar s'ha vulnerat, quina acció tècnica es proposa i quin risc queda obert encara que el model passi l'audit gate.

```
audit:
  min_disparate_impact: 0.80
  max_demographic_parity_diff: 0.15
  max_equal_opportunity_diff: 0.15
  max_robustness_drop: 0.10
  min_local_fidelity: 0.70
```

Listing 3.5: Exemple de llindars al fitxer de configuració D3

Laboratori de construcció:

Objectiu: generar l'informe D3 a partir dels fitxers d'auditoria. L'informe ha d'explicar quins criteris passen, quins fallen i quina decisió d'enginyeria es proposa.

```
python -m lic_project.report_d3 --audit-dir reports/audit --output reports/d3_report.md
```

Listing 3.6: Generació de l'informe tècnic D3

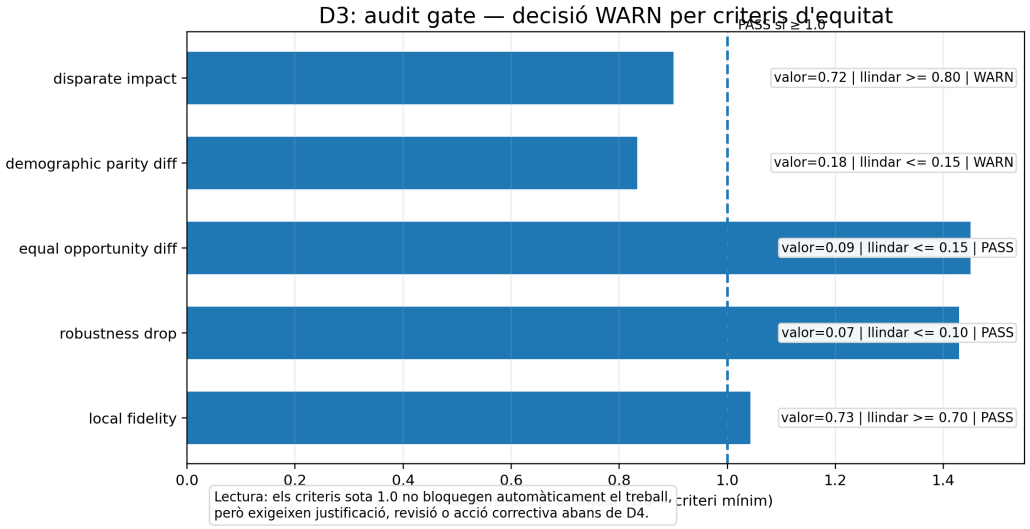

Figura 3.4: Resum del *audit gate* D3. Cada barra mostra el marge del criteri respecte al llindar configurat: el valor 1.0 indica el mínim necessari per passar. En aquest exemple, robustesa i fidelitat local compleixen el contracte, però els criteris d'equitat queden per sota del llindar i generen una decisió global `WARN`. El model no queda automàticament descartat, però l'informe ha de justificar el risc i proposar una acció correctiva abans de continuar cap a D4.

Laboratori de construcció: (continuació)

Per superar aquesta part, l'informe D3 ha de contenir una decisió clara: **acceptar**, **acceptar amb advertiments**, **reentrenar** o **bloquejar**. Una auditoria que només enumera mètriques sense decisió no compleix el contracte del capítol.

3.6 Microcas de Recerca: auditoria d'un proxy en triatge sanitari

Aquesta activitat aplica el contracte D3 a un cas docent inspirat en problemes reals de biaix de proxy en sistemes de triatge. El cas es formula com a exercici de laboratori: un model intenta prioritzar pacients usant una etiqueta històrica relacionada amb cost o ús de recursos. El risc és que el cost no mesuri necessitat clínica, sinó accés desigual al sistema.

El model pot estar ben implementat i alhora aprendre una etiqueta inadequada. Si Y representa cost històric, però l'objectiu real és necessitat de salut, el model optimitza una aproximació contaminada. D3 obliga a preguntar:

- què representa exactament l'etiqueta?;
- quins grups poden tenir infrarepresentació en aquesta etiqueta?;
- quines variables actuen com a proxies?;
- quina mètrica desagregada revela la fallada?;
- quina decisió de reetiquetatge o redisseny proposem?

Laboratori de construcció:

Objectiu: usar les eines del capítol per detectar si un model entrenat amb un proxy produeix diferències sistemàtiques entre grups. El resultat ha de ser una recomanació concreta: canviar etiqueta, canviar features, revisar llindar, afegir monitoratge o bloquejar el model.
No afirmeu causalitat si només teniu una auditoria predictiva. Podeu detectar un patró

Laboratori de construcció: (continuació)

preocupant i proposar una revisió de dades, però no convertir una correlació en diagnòstic social sense evidència addicional.
L'informe del cas ha d'incloure: hipòtesi de proxy, mètriques per grup, explicació local d'una decisió crítica, prova de robustesa i decisió final. Si el model falla equitat però funciona globalment, no es pot vendre com a èxit tècnic.

Microcas o incidència de laboratori: Cas docent: cost històric com a proxy de necessitat

Un sistema de triatge aprèn a predir cost futur perquè és una etiqueta fàcil d'obtenir. En auditar-lo, es detecta que un grup amb menys accés històric al sistema genera menys cost tot i tenir necessitat real elevada. El model no ha descobert menor risc; ha après menor ús històric de recursos.
Acció d'enginyeria: substituir o complementar l'etiqueta amb variables més properes a necessitat real, com indicadors clínics, episodis de risc, reingressos o valoracions professionals revisables.
Lliçó per al D3: la qualitat d'un model no pot ser superior a la qualitat conceptual de l'objectiu que optimitza.

3.7 Conclusions: tancament del D3 i pas cap a sistemes amb recuperació

Aquest capítol ha transformat el model entrenat en un sistema auditable. La progressió és clara: D1 va construir el baseline, D2 va entrenar un model profund i D3 ha afegit verificació. A partir d'ara, un resultat no és defensable si no inclou mètriques globals, subgrups, explicabilitat, robustesa i decisió d'auditoria.

L'ètica computacional s'ha de traduir en fitxers, tests, llindars i informes. Això no resol tots els dilemes socials, però evita una fallada bàsica d'enginyeria: desplegar un model que ningú ha comprovat de manera reproduïble.

Microcas o incidència de laboratori: Auditoria que bloqueja un model "bo"

Un model supera D2 amb bona mètrica global, però D3 revela diferències fortes en taxa de falsos negatius per subgrup i una robustesa molt baixa davant petites pertorbacions. El model és tècnicament interessant, però no passa el contracte d'auditoria.
Lliçó final del capítol: D3 no busca destruir models, sinó fer visibles els riscos que una mètrica global amagaria. El següent capítol afegirà recuperació documental i RAG; aquesta auditoria serà imprescindible per controlar què respon el sistema i amb quina evidència.

Tancament del D3

Artefacte lliurable

D3 — Auditoria reproduïble. Al final del capítol heu de lliurar:

- execució documentada de `python -m lic_project.audit --config configs/d3_audit.yaml`;
- fitxers `reports/audit/d3_audit_metrics.json` i `reports/audit/d3_audit_summary.json`;
- figures d'equitat, explicabilitat, robustesa i *audit gate*;
- informe `reports/d3_report.md`;
- tests mínims executats amb `pytest`;
- decisió final: acceptar, acceptar amb advertiments, reentrenar o bloquejar.

Checklist de verificació

Abans de donar D3 per acabat, comproveu:

- l'auditoria s'executa amb una comanda documentada;
- les mètriques per subgrup no depenen d'un notebook;
- els llindars són explícits i configurables;
- les explicacions indiquen limitacions;
- hi ha una prova de robustesa i els tests passen;
- l'informe conté una decisió d'enginyeria, no només gràfics.

Criteris d'acceptació

Criteri	Punts
Pipeline d'auditoria executable i sense dependència d'un notebook	2.0
Mètriques globals i per subgrup correctes	1.5
Explicació global/local amb limitacions declarades	1.5
Prova de robustesa i política d'abstenció o escalat	1.5
Audit gate amb llindars configurables i resum PASS/WARN/FAIL	1.5
Tests mínims de mètriques, explicacions i contracte	1.0
Informe tècnic D3 amb decisió justificada	1.0

Capítol 4

Sistemes amb Recuperació: RAG local, evidència i traçabilitat

Els capítols anteriors han construït una progressió d'enginyeria: D1 ha definit un baseline reproduïble, D2 ha entrenat un model profund amb checkpoint i D3 ha convertit el model en un objecte auditable. En aquest capítol el sistema deixa de respondre només a partir dels seus pesos o d'una taula de dades: ha de recuperar evidència documental, citar-la i abstenir-se quan el corpus no dona suport a la resposta.

El producte del capítol no és una demostració visual d'un espai latent ni una discussió abstracta sobre models fundacionals. El producte és el lliurable D4: un *Retrieval-Augmented Generation* (RAG) local mínim, executable i auditable, que ingereix documents, els fragmenta, construeix un índex, recupera fragments rellevants, aplica diversitat amb *Maximal Marginal Relevance* (MMR), genera una resposta fonamentada i conserva cites, scores i llindars. Les figures són sortides del pipeline, no substituts del pipeline.

Repte d'enginyeria

Repte del capítol. Construir un sistema RAG local amb evidència traçable i política d'abstenció.
Entregable associat: D4.
Artefactes mínims:

- un corpus local a `data/corpus/`;
- un mòdul de càrrega i *chunking* a `src/lic_project/rag_data.py`;
- un índex vectorial reproduïble a `src/lic_project/rag_index.py`;
- recuperació *top-k* i MMR amb configuració externa;
- una resposta fonamentada amb cites i llindar de *grounding*;
- fitxers `reports/rag/d4_retrieval_results.json` i `reports/rag/d4_rag_metrics.json`;
- figures generades pel pipeline;
- informe `reports/d4_report.md`;
- tests mínims de *chunking*, recuperació i política d'abstenció.

Laboratori de construcció:

En aquest capítol no s'accepta una resposta generada sense evidència. El professorat ha de poder executar el pipeline, veure quins fragments s'han recuperat, inspeccionar les puntuacions, regenerar les figures i comprovar que el sistema respon "no tinc evidència suficient" quan el corpus no conté la informació.
Un RAG que sempre respon no és un RAG segur. Si el sistema no sap abstenir-se quan la similitud és baixa o quan cap fragment dona suport a la resposta, està convertint una consulta sense evidència en una al · lucinació amb format professional.

Artefacte lliurable

Contracte tècnic D4. En acabar el capítol, el professorat ha de poder executar:

```
python -m lic_project.rag_pipeline --config configs/d4_rag.yaml
python -m lic_project.report_d4 --rag-dir reports/rag --output reports/d4_report.md
pytest
```

Si alguna d'aquestes comandes falla, el capítol no s'ha completat com a laboratori d'enginyeria.

```
lic_project/
|-- data/
|   `-- corpus/
|-- configs/
|   |-- d1_baseline.yaml
|   |-- d2_training.yaml
|   |-- d3_audit.yaml
|   `-- d4_rag.yaml
|-- artifacts/
|   `-- indexes/
|-- reports/
|   |-- figures/
|   |-- rag/
|   `-- d4_report.md
|-- src/
|   `-- lic_project/
|       |-- rag_data.py
|       |-- rag_index.py
|       |-- rag_answer.py
|       |-- rag_eval.py
|       |-- rag_pipeline.py
|       `-- report_d4.py
`-- tests/
    |-- test_rag_chunking.py
    |-- test_rag_retrieval.py
    `-- test_grounding_gate.py
```

Listing 4.1: Estructura incremental D4 sobre el repositori D1–D3

Decisió d'arquitectura

Decisió P0: API stateless i base vectorial persistent.
En un prototip petit podem guardar un índex local a `artifacts/indexes/`. Però quan el RAG s'exposa com a servei, l'API no ha de reconstruir l'índex a l'arrencada ni mantenir-lo només en memòria. Això crea tres problemes: arrencades lentes, pèrdua d'estat quan el contenidor es reinicia i inconsistència quan hi ha més d'una rèplica de l'API.

Regla del LIC. El servei FastAPI ha de ser *stateless*: rep consultes, demana recuperació a una base vectorial persistent i retorna resposta amb cites. L'estat documental —col · leccions, embeddings, metadades i versions d'índex— viu fora de l'API, en una base vectorial persistent o en un volum explícit.

Arquitectura mínima de producció docent:

- `rag_ingest`: procés separat que ingereix PDFs, genera chunks i actualitza l'índex;
- `chromadb` o servei equivalent: base vectorial persistent amb volum;
- `rag_api`: servei FastAPI sense estat, que només consulta l'índex;

Taula 4.1: Decisions operatives en la ingesta documental D4

Decisió	Opció recomanada en D4	Risc si es fa malament
Mida del fragment	80–150 paraules o unitat semàntica equivalent	Fragments massa grans dilueixen l'embedding; massa petits perden context
Solapament	15–25% aproximadament	Sense solapament es poden tallar clàusules o condicions
Metadades	`doc_id`, `chunk_id`, posició i versió d'índex	Respostes impossibles d'auditar
Cobertura	Corpus explícit i llista de buits coneguts	Falsos "no hi ha risc" quan simplement falten documents

Decisió d'arquitectura *(continuació)*

- `reports/rag/`: mètriques, consultes de prova i evidència de recuperació.

Risc si es fa malament. Si l'API carrega PDFs i construeix l'índex en arrencar, cada reinici pot canviar el comportament, cada rèplica pot tenir un estat diferent i el temps d'arrencada pot bloquejar el servei.

4.1 Del corpus al fragment: ingesta, chunking i contracte documental

Un sistema RAG és tan fiable com el seu corpus i com la manera com aquest corpus es fragmenta. Abans de parlar de models fundacionals, cal resoldre una decisió molt més concreta: quina és la unitat mínima d'evidència que el sistema pot citar? Un document complet pot ser massa llarg i diluir el significat; un fragment massa curt pot perdre context. D4 converteix aquesta decisió en configuració, no en intuïció informal.

Donat un corpus $\mathcal{C} = \{d_1, \ldots, d_N\}$, cada document es divideix en fragments c_{ij} amb metadades mínimes:

$$c_{ij} = \{\texttt{chunk\_id}, \texttt{doc\_id}, \texttt{text}, \texttt{start}, \texttt{end}\}. \tag{4.1}$$

La part important no és la notació, sinó el contracte: tota resposta posterior ha de poder apuntar a un `chunk_id`. Sense identificador de fragment no hi ha traçabilitat.

Decisió d'arquitectura

Chunk petit o chunk gran? Un fragment petit millora precisió local i facilita cites concretes, però pot tallar condicions importants. Un fragment gran conserva context, però dilueix l'embedding i pot recuperar text massa genèric.

Decisió D4: començar amb chunks semàntics curts amb solapament moderat i metadades completes. Si el sistema falla per manca de context, abans d'augmentar molt la mida reviseu si el solapament i les metadades resolen el problema.

Troubleshooting i depuració

Incidència de laboratori: chunking que talla l'evidència crítica. Símptoma: el recuperador troba fragments relacionats amb la consulta, però cap conté la condició que canvia la resposta. La causa probable és que el tall ha separat una regla de la seva excepció.

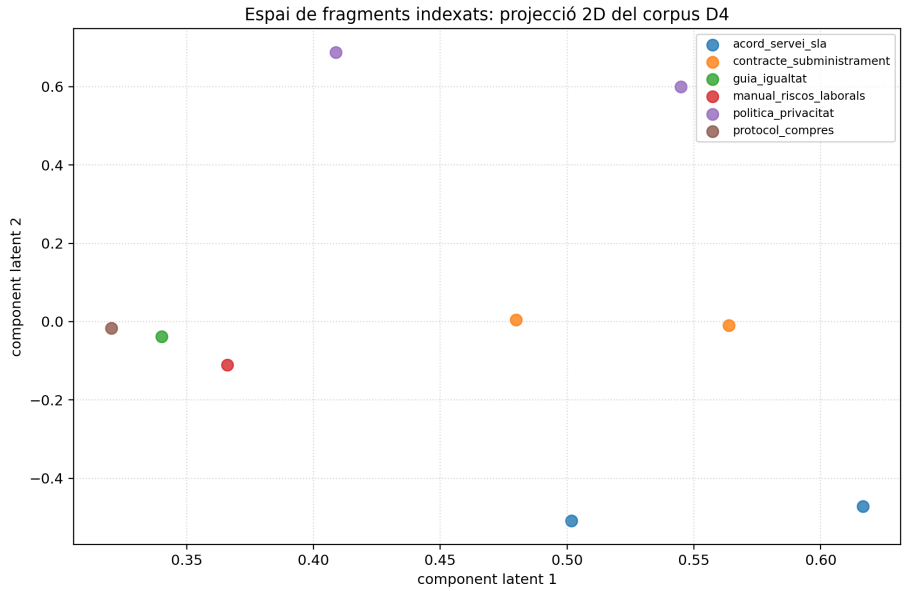

Figura 4.1: Projecció 2D dels fragments indexats. La figura permet inspeccionar cobertura i separació temàtica, però la qualitat del RAG s'ha de validar amb consultes i evidència recuperada.

Troubleshooting i depuració *(continuació)*

Com detectar-ho: reviseu consultes on el top-k recupera fragments d'un mateix document però la resposta no és verificable. Compareu amb una versió amb més solapament o amb tall per paràgraf. **Acció d'enginyeria:** ajustar la unitat de chunking, conservar posició i document d'origen, i afegir un test que comprovi que una clàusula i la seva excepció no queden sempre separades.

Laboratori de construcció: Indexar el primer corpus D4

Objectiu: carregar un corpus local, crear fragments, construir un índex vectorial i guardar els artefactes de l'índex. La figura de l'espai latent només és una eina per inspeccionar si els fragments s'agrupen de manera raonable.
Abans de millorar el model d'embeddings, tanqueu el contracte documental: corpus, chunking, metadades, versió i criteri d'exclusió. Molts errors de RAG semblen errors semàntics, però són errors d'ingesta.

```
python -m lic_project.rag_pipeline --config configs/d4_rag.yaml
```

Listing 4.2: Execució de la ingesta i indexació D4

Una projecció bonica no demostra recuperació correcta. Serveix per detectar corpus massa barrejat, duplicats o fragments fora de domini, però la decisió final depèn de consultes de prova i mètriques de recuperació.

4.2 Recuperació vectorial: top-k, MMR i diversitat de context

La recuperació és el cor del capítol. Si el sistema recupera malament, el generador respondrà malament encara que el model lingüístic sigui excel · lent. En D4 no entrenem un LLM; construïm un mecanisme perquè el sistema sàpiga quin context pot usar i quin context ha de rebutjar.

Un recuperador vectorial transforma una consulta q i cada fragment c_i en vectors normalitzats.

Taula 4.2: Estratègies de recuperació llegides com a decisions d'enginyeria

Estratègia	Quan usar-la	Risc
Top-k vectorial	Consulta semàntica oberta	Pot recuperar fragments molt redundants
Cerca lèxica	Codis, identificadors, noms exactes	No captura sinònims o formulacions equivalents
Híbrida	Corpus amb llenguatge natural i codis exactes	Cal calibrar el pes entre semàntica i literalitat
MMR	Context limitat i risc de redundància	Pot perdre el fragment més similar si λ és massa baix

La rellevància es pot estimar amb similitud cosinus:

$$s(q, c_i) = \frac{e_q \cdot e_i}{\|e_q\|\|e_i\|}. \tag{4.2}$$

El *top-k* pur selecciona els fragments amb més puntuació. El problema és que sovint retorna fragments redundants. Per això afegim MMR:

$$\text{MMR}(d_i) = \lambda s(q, d_i) - (1 - \lambda) \max_{d_j \in S} s(d_i, d_j). \tag{4.3}$$

La lectura operativa és clara: el sistema ha de recuperar fragments rellevants, però també prou diversos per no malgastar la finestra de context.

Costos i eficiència

Cost de recuperació i mida de l'índex. Un índex local és suficient per al laboratori, però no és gratuït: cada reindexació consumeix temps, cada embedding ocupa memòria i cada consulta té latència. El criteri professional no és indexar-ho tot sense pensar, sinó saber què s'ha versionat i quin cost té regenerar-ho.

Regla D4: registreu nombre de documents, nombre de chunks, mida aproximada de l'índex i temps de reindexació. Aquestes dades alimentaran el monitoratge del Capítol 5.

Laboratori de construcció: Recuperar evidència amb MMR

Objectiu: comparar la selecció *top-k* amb la selecció MMR sobre la mateixa consulta. L'alumnat ha d'explicar si el context injectat és més divers i si aquesta diversitat ajuda o perjudica la resposta.
Per superar aquesta part, cal presentar una consulta, els fragments recuperats, les puntuacions, el valor de λ i una justificació de si MMR millora la cobertura del context.

Microcas o incidència de laboratori: Incidència de laboratori: cinc fragments iguals no fan una bona resposta

Un grup obté una similitud alta perquè el recuperador retorna cinc fragments gairebé idèntics sobre la mateixa clàusula. La resposta sembla ben citada, però no cobreix excepcions ni condicions.

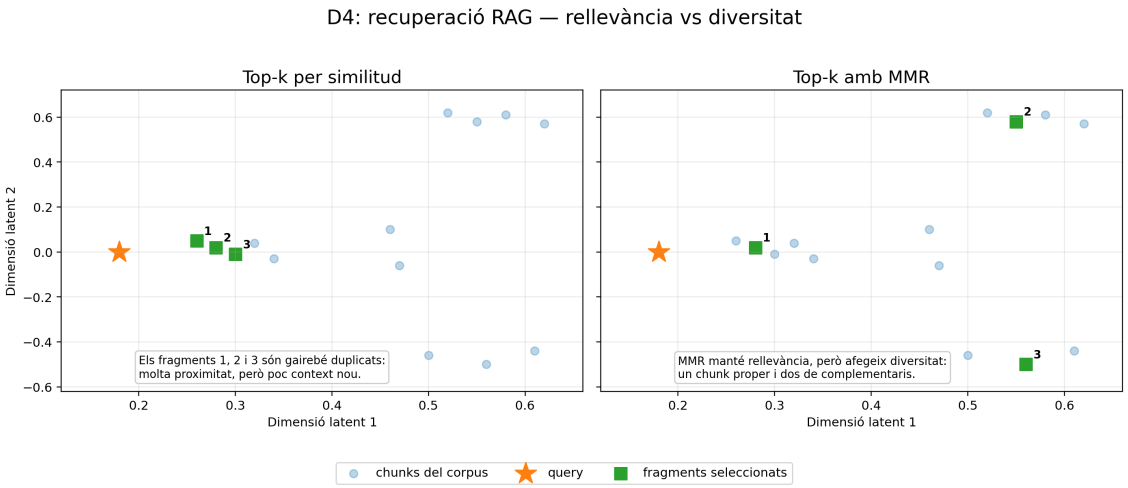

Figura 4.2: Comparació entre recuperació *top-k* i recuperació amb MMR en D4. A l'esquerra, el criteri de similitud pura selecciona fragments molt propers a la consulta, però també molt semblants entre si, de manera que la resposta pot quedar sustentada per evidència redundant. A la dreta, MMR manté rellevància, però introdueix diversitat i afavoreix fragments complementaris de diferents zones del corpus. La decisió d'enginyeria no és només recuperar el que és més proper, sinó construir un context útil, no redundant i més robust per a la resposta final.

Microcas o incidència de laboratori: Incidència de laboratori: cinc fragments iguals no fan una bona resposta *(continuació)*

Acció d'enginyeria: activar MMR, revisar el chunking i comprovar si el context inclou fragments complementaris.
Lliçó per al D4: la finestra de context és un recurs limitat; omplir-la de redundància és una forma silenciosa de perdre recall.

4.3 Generació fonamentada: resposta, cites i política d'abstenció

En un RAG docent, la generació pot ser una plantilla extractiva. Això és intencionat: abans de connectar un LLM extern, cal demostrar que el sistema sap recuperar evidència, citar-la i rebutjar preguntes sense suport documental. Un LLM afegeix fluïdesa; no ha de substituir el contracte d'evidència. Podem expressar RAG com una descomposició simple:

$$\text{resposta} = G(q, Z_k), \quad Z_k = R(q, \mathcal{C}), \tag{4.4}$$

on R és el recuperador i G és el generador. La pregunta d'enginyeria no és si G escriu bé, sinó si Z_k conté evidència suficient. Per això D4 introdueix un llindar mínim:

$$\max_i s(q, c_i) < \tau \Rightarrow \text{abstenció}. \tag{4.5}$$

Laboratori de construcció: Resposta amb cites i gate de grounding

Objectiu: generar respostes amb cites i provar almenys una consulta sense evidència al corpus. El sistema ha de retornar una abstenció clara, no una resposta inventada.
Una resposta amb cites no és automàticament una resposta correcta. Les cites han de donar suport explícit a la frase generada. Si el sistema cita fragments tangencials, el problema és de *groundedness*, no de redacció.

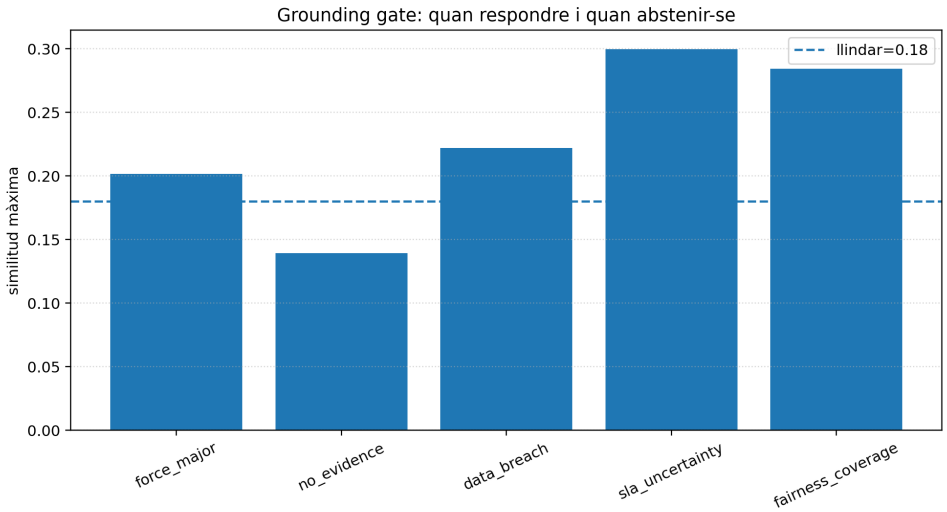

Figura 4.3: *Grounding gate* de D4. Les consultes sota el llindar no han de generar resposta substantiva; han d'activar abstenció o revisió humana.

Laboratori de construcció: Resposta amb cites i gate de grounding *(continuació)*

No escriviu mai "no s'ha trobat risc" si el que ha passat és "no s'ha trobat evidència". La primera frase és una conclusió; la segona és una limitació del sistema.

4.4 Restriccions estructurals: metadades, grafs i validació de cites

La recuperació semàntica és potent, però no entén totes les relacions. En documents corporatius, sovint cal saber si una clàusula pertany a un contracte concret, si una entitat és filial d'una altra, si una política és vigent o si un fragment ha estat derogat. D4 no construeix un GraphRAG complet, però introdueix una idea professional: les metadades i les relacions estructurals poden actuar com a *gate* abans de generar.

Una consulta pot recuperar fragments semànticament propers però estructuralment inadequats. Per exemple, un fragment pot parlar de força major però pertànyer a una versió antiga del contracte. Per evitar-ho, el pipeline ha de conservar metadades i permetre filtres:

- document d'origen;
- data o versió;
- tipus de document;
- estat: vigent, esborrany, derogat;
- relació amb entitats o expedients.

Microcas o incidència de laboratori: Fragment correcte, document equivocat

El sistema recupera una clàusula correcta sobre notificació de bretxes de seguretat, però prové d'una política antiga no vigent. La resposta és semànticament raonable i jurídicament problemàtica.
Acció d'enginyeria: afegir metadades de versió i un filtre que impedeixi citar documents derogats o no aplicables.
Lliçó per al D4: la similitud semàntica és una condició útil, però no és una condició suficient d'acceptació.

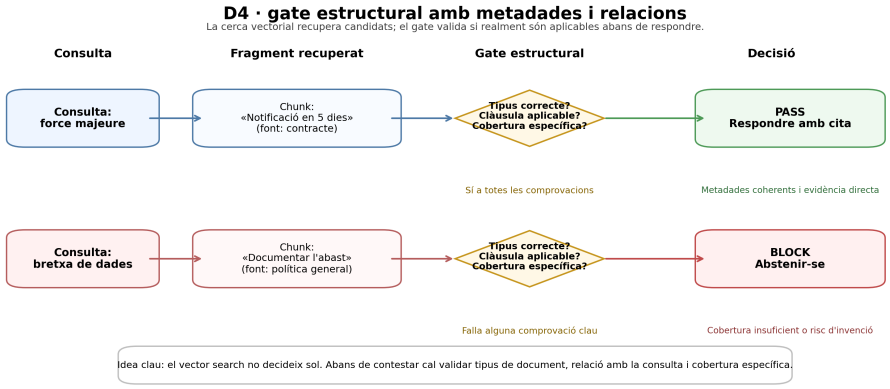

Figura 4.4: Gate estructural de D4 basat en metadades i relacions. La recuperació vectorial només proposa fragments candidats; la decisió final depèn d'un segon filtre que comprova si el fragment és aplicable al domini i a la consulta. Quan les metadades i les relacions estructurals són coherents, el sistema pot respondre amb cita; quan la cobertura és indirecta o insuficient, el gate ha de bloquejar la resposta i activar abstenció.

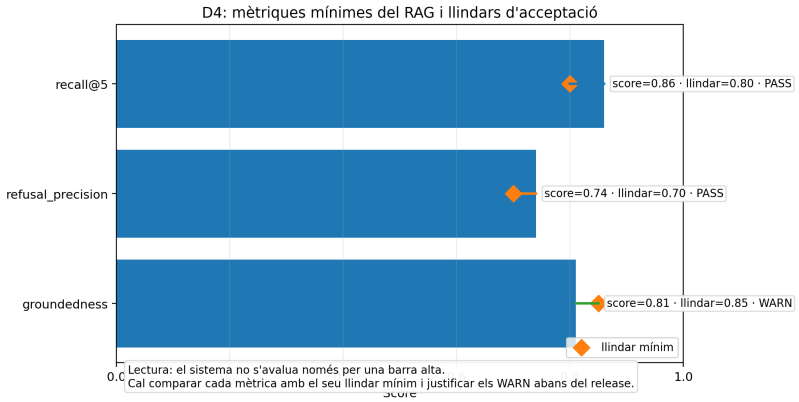

Figura 4.5: Avaluació mínima del RAG en D4. Cada barra mostra el valor observat d'una mètrica clau i el marcador indica el llindar mínim d'acceptació. L'objectiu no és obtenir tres valors màxims, sinó verificar si el sistema recupera prou evidència (`recall@5`), s'absté quan no en té prou (`refusal_precision`) i manté traçabilitat factual (`groundedness`). Les mètriques per sota del llindar no bloquegen necessàriament tot el projecte, però exigeixen justificació i pla de millora abans del release.

4.5 Avaluació del RAG: recall, abstenció i regressió de qualitat

Un RAG no s'avalua només llegint una resposta que "sona bé". Cal un conjunt de consultes de prova amb documents esperats, consultes sense resposta i criteris de regressió. Això connecta directament amb D3: l'auditoria no desapareix quan afegim recuperació; s'estén al pipeline documental. El paquet D4 calcula tres mètriques mínimes:

- **Recall@k**: el document esperat apareix entre els k fragments recuperats?
- **Refusal accuracy**: el sistema s'absté quan no hi ha evidència al corpus?
- **Groundedness operativa**: les respostes contestades tenen cites associades?

Estratègia de validació

Test de recuperació amb corpus mínim. El repositori ha d'incloure un corpus de prova amb tres fragments i dues preguntes: una amb evidència present i una sense evidència. El test ha de comprovar que la primera recupera el fragment esperat i que la segona activa abstenció.

Taula 4.3: Quan considerar tècniques avançades després d'un D4 funcional

Tècnica	Quan té sentit	Quan no toca encara
LoRA/QLoRA	El model no domina el to o format del domini tot i recuperar bona evidència	Si el problema és que no recupera els fragments adequats
DPO/RLHF	Cal imposar preferències de resposta o política d'abstenció en un generador real	Si encara no hi ha dataset de preferències ni mètriques de grounding
GraphRAG	Les relacions estructurals són imprescindibles per respondre	Si només cal recuperar fragments textuals simples
Vector DB externa	El corpus supera la memòria o cal concurrència real	Si el prototip local encara no té tests ni informe

Estratègia de validació *(continuació)*

Aquest test és petit però crític: protegeix contra regressions en chunking, vectorització, ordenació top-k i gate de grounding.

Laboratori de construcció: Informe D4 i regressió de qualitat

Objectiu: generar un informe D4 amb consultes, fragments recuperats, cites, decisions d'abstenció i mètriques globals. L'informe ha de permetre detectar si un canvi de chunking, vectorizer o llindar empitjora el sistema.
No intenteu arreglar un mal RAG amb fine-tuning. Si els fragments recuperats no contenen la resposta, cap LoRA honest ho solucionarà. Primer recupereu bé; després decidiu si cal adaptar el generador.

```
python -m lic_project.report_d4 --rag-dir reports/rag --output reports/d4_report.md
```

Listing 4.3: Generació de l'informe tècnic D4

L'informe D4 ha de respondre: quines consultes s'han provat, quins fragments s'han recuperat, quines cites sustenten cada resposta, quina consulta s'ha rebutjat i quin canvi de configuració revisaríeu abans del Capítol 5. Així, ha d'explicar quin fragment suporta cada afirmació, quin llindar activa abstenció, quin cas sense evidència s'ha provat i quin canvi de corpus podria degradar el sistema.

4.6 Conclusions: cap al monitoratge

Aquest capítol ha convertit el projecte en un sistema amb memòria externa. La progressió és ara completa: D1 construeix el baseline, D2 entrena, D3 audita i D4 afegeix recuperació documental amb cites. A partir d'ara, una resposta no és acceptable perquè sembli fluent; és acceptable si es pot reconstruir el camí des de la consulta fins als fragments i des dels fragments fins a la decisió de respondre o abstenir-se.

Microcas o incidència de laboratori: resposta perfecta sense cap cita

Un equip connecta un LLM extern i obté una resposta molt ben redactada sobre una clàusula contractual. En revisar el pipeline, cap fragment recuperat conté la clàusula citada. La resposta pot ser plausible, però no és defensable.

Lliçó final del capítol: D4 no busca el chatbot més brillant, sinó el primer sistema amb memòria documental que sap mostrar proves, declarar incertesa i deixar rastre. El Capítol 5 haurà de convertir aquest comportament en monitoratge: regressió de qualitat, drift documental i proves contínues.

Tancament del D4

Artefacte lliurable

D4 — RAG local amb evidència traçable. Al final del capítol heu de lliurar:

- corpus local documentat a `data/corpus/`;
- execució de `python -m lic_project.rag_pipeline --config configs/d4_rag.yaml`;
- índex i fragments guardats a `artifacts/indexes/` en mode prototip, o bé col · lecció vectorial persistent amb volum Docker en mode servei;
- separació explícita entre procés d'ingesta i servei d'inferència;
- prova que l'API pot reiniciar-se sense reingerir el corpus;
- fitxers `reports/rag/d4_retrieval_results.json` i `reports/rag/d4_rag_metrics.json`;
- figures d'espai latent, MMR, gate de grounding, restriccions i avaluació;
- informe `reports/d4_report.md`;
- tests mínims executats amb `pytest`;
- decisió final: el RAG és acceptable, acceptable amb advertiments o bloquejat per manca d'evidència.

Checklist de verificació

Abans de donar D4 per acabat, comproveu:

- el corpus és explícit i versionable;
- cada resposta conté cites o abstenció;
- els llindars són configurables;
- les figures es regeneren des del pipeline;
- hi ha almenys una consulta sense evidència que el sistema rebutja;
- els tests passen;
- l'informe explica una decisió d'enginyeria, no només mostra una resposta generada.

Criteris d'acceptació

Criteri	Punts
Pipeline RAG executable i sense dependència d'un notebook	2.0
Chunking, índex i metadades reproduïbles	1.5
Recuperació top-k/MMR amb configuració externa	1.5
Resposta amb cites i política d'abstenció	1.5
Avaluació amb consultes de prova i mètriques D4	1.5
Tests mínims de chunking, recuperació i gate	1.0
Informe tècnic D4 amb decisió justificada	1.0

Capítol 5

Monitoratge i Avaluació Contínua: Drift, regressió i qualitat en producció

Els capítols anteriors han construït una progressió d'enginyeria: D1 ha definit un baseline reproduïble, D2 ha entrenat un model profund, D3 ha convertit el model en un objecte auditable i D4 ha afegit recuperació documental amb cites i política d'abstenció. En aquest capítol el sistema passa a ser un servei que pot degradar-se amb el temps.

El producte del capítol és el lliurable D5: un pipeline de monitoratge que compara el comportament actual amb una línia base, detecta regressió de qualitat, identifica deriva de dades i d'embeddings, calibra un jutge automàtic i genera una cua de casos per revisió humana. Les figures són sortides del pipeline; el resultat avaluable són els fitxers de mètriques, els llindars i la decisió del gate.

Repte d'enginyeria

Repte del capítol. Convertir el RAG D4 i el model auditat D3 en un sistema monitoritzable.
Entregable associat: D5.
Artefactes mínims:

- configuració `configs/d5_monitoring.yaml`;
- pipeline executable `src/lic_project/monitoring_pipeline.py`;
- mètriques de regressió de qualitat respecte D4;
- detecció de drift amb KS, PSI i MMD;
- calibratge mínim del jutge automàtic;
- cua d'*active learning* per revisar casos incerts;
- fitxers `reports/monitoring/d5_monitoring_metrics.json` i `d5_monitoring_summary.json`;
- informe `reports/d5_report.md`;
- figures generades pel pipeline;
- tests mínims de mètriques i contracte de monitoratge.

Laboratori de construcció:

En D5 no s'accepta dir que el sistema "continua funcionant" sense evidència. El professorat ha de poder executar el pipeline, veure quines mètriques han caigut, quines distribucions han canviat, si el jutge automàtic està calibrat i quins casos s'han enviat a revisió humana.

Una bona demo de RAG no és un servei estable. Si canvia el corpus, el patró de consultes, el llenguatge de les persones usuàries o la distribució dels documents, el sistema pot començar a fallar encara que el codi no hagi canviat.

Artefacte lliurable

Contracte tècnic D5. En acabar el capítol, el professorat ha de poder executar:

```
python -m lic_project.monitoring_pipeline --config configs/d5_monitoring.yaml
python -m lic_project.report_d5 \
    --monitoring-dir reports/monitoring \
    --output reports/d5_report.md
pytest
```

Si alguna d'aquestes comandes falla, el capítol no s'ha completat com a laboratori d'enginyeria.

```
lic_project/
|-- configs/
|   |-- d1_baseline.yaml
|   |-- d2_training.yaml
|   |-- d3_audit.yaml
|   |-- d4_rag.yaml
|   `-- d5_monitoring.yaml
|-- reports/
|   |-- figures/
|   |-- monitoring/
|   |   |-- d5_monitoring_metrics.json
|   |   `-- d5_monitoring_summary.json
|   `-- d5_report.md
|-- src/
|   `-- lic_project/
|       |-- monitoring_data.py
|       |-- monitoring_metrics.py
|       |-- monitoring_pipeline.py
|       `-- report_d5.py
`-- tests/
    |-- test_monitoring_metrics.py
    `-- test_monitoring_contract.py
```

Listing 5.1: Estructura incremental D5 sobre el repositori D1–D4

Estratègia de validació

No tots els tests han de carregar el model real. En un pipeline MLOps, els tests es divideixen en capes:

- **Unit tests:** comproven funcions petites, contractes de dades, formes de tensors, parsers, gates i validacions. Han de ser ràpids i no han de carregar models grans.
- **Integration tests:** executen el model real, l'índex real o el pipeline complet. Són més lents i poden executar-se només en releases o manualment.
- **Mocks i dummies:** substitueixen un model gran per una xarxa mínima o un objecte que retorna una sortida controlada. Serveixen per comprovar que el flux de dades funciona.

Regla del LIC. El CI ha de poder executar els unit tests en menys d'uns minuts i sense GPU. Els tests que carreguen models grans han d'estar marcats com a integració.

5.1 De l'avaluació puntual a la regressió de qualitat

D4 ha definit un sistema que recupera evidència, cita fragments i s'absté quan el corpus no dona suport a la resposta. Aquest resultat, però, només és una fotografia. Un sistema real rep consultes noves, documents nous, canvis de política i distribucions diferents. D5 introdueix una unitat nova: la regressió de qualitat.

Taula 5.1: Mètriques de regressió de qualitat per a D5

Mètrica	Què detecta	Acció si cau
Recall@k	El document o fragment esperat ja no apareix entre els recuperats	Revisar chunking, embeddings, índex o corpus
Refusal accuracy	El sistema ja no s'absté quan no hi ha evidència	Revisar llindar de grounding i conjunt de consultes negatives
Groundedness	Les respostes contestades no tenen suport suficient en cites	Bloquejar respostes i revisar prompt o política de generació

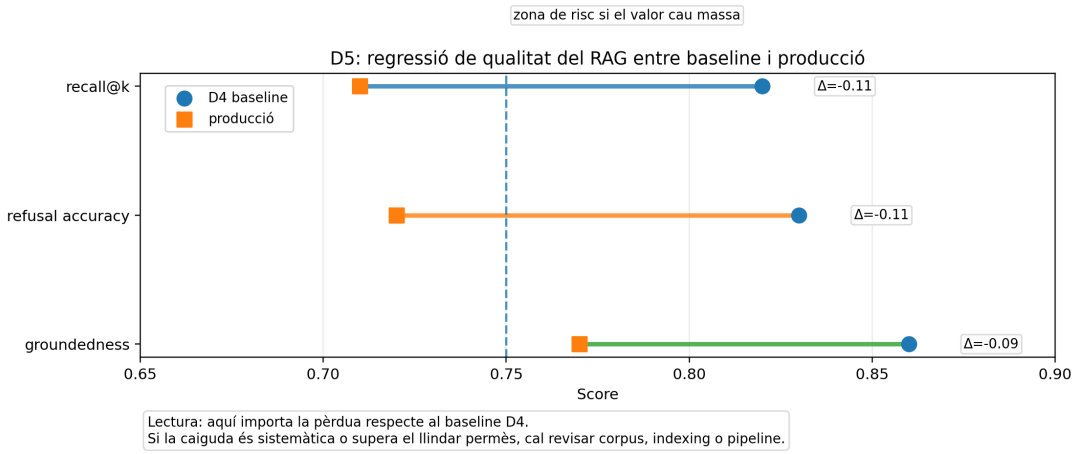

Figura 5.1: Regressió de qualitat del RAG en D5. Cada línia compara el valor de referència del baseline D4 amb la finestra actual de producció i en destaca la caiguda Δ. L'objectiu no és comparar barres gairebé iguals, sinó detectar si la degradació és sistemàtica i si supera el llindar de tolerància operativa. Quan la pèrdua és persistent, el sistema ha d'obrir una investigació sobre corpus, indexació, prompting o qualitat de les fonts abans de continuar en producció.

Una regressió de qualitat es produeix quan una mètrica rellevant cau respecte a una línia base acceptada. Si M_t és una mètrica actual i M_0 és la mètrica de referència, definim una caiguda operativa com:

$$\Delta M = M_0 - M_t. \tag{5.1}$$

El valor important no és només M_t, sinó si ΔM supera un llindar acordat. En un RAG, les mètriques mínimes a monitoritzar són les mateixes que hem construït a D4: *recall@k*, *refusal accuracy* i *groundedness*.

Decisió d'arquitectura

Monitorar mètriques agregades o monitorar per segments? Les mètriques agregades són simples i barates, però poden amagar degradacions concentrades en un tipus de consulta, un subgrup o una família documental. Les mètriques per segment augmenten complexitat, però converteixen el monitoratge en una eina real de diagnosi.

Decisió D5: com a mínim, manteniu mètriques globals i un segment crític definit pel projecte. Si el sistema afecta col · lectius o documents de naturalesa diferent, el monitoratge només global no és suficient.

Laboratori de construcció: Regressió respecte a D4

Objectiu: comparar les mètriques D4 amb una finestra simulada de producció. El lliurable no és un gràfic, sinó un fitxer de mètriques i una decisió: `PASS`, `WARN` o `FAIL`.

```
python -m lic_project.monitoring_pipeline --config configs/d5_monitoring.yaml
```

Listing 5.2: Execució del monitoratge D5

No convertiu el monitoratge en una taula decorativa. Si una mètrica cau per sota del llindar, el pipeline ha de produir una decisió operacional: continuar, advertir, reindexar, reavaluar o bloquejar.

5.2 Drift de dades: quan el món canvia sota el model

El drift apareix quan la distribució actual ja no és equivalent a la distribució usada per validar el sistema. En classificació, pot afectar variables tabulars; en RAG, pot afectar consultes, embeddings, documents recuperats o patrons de resposta.

En D5 no intentem demostrar tota la teoria estadística del drift; implementem proves suficients per detectar que el sistema ha sortit del règim conegut.

Per a una variable contínua, una prova simple és l'estadístic Kolmogorov–Smirnov:

$$D_{KS} = \sup_x |F_{ref}(x) - F_{prod}(x)|. \tag{5.2}$$

També podem usar el *Population Stability Index* per quantificar canvis per bins. En embeddings, on cada consulta o fragment és un vector, usem una distància de distribucions com MMD:

$$\mathrm{MMD}^2(P, Q) = \mathbb{E}_{x,x' \sim P}[k(x, x')] + \mathbb{E}_{y,y' \sim Q}[k(y, y')] - 2\mathbb{E}_{x \sim P, y \sim Q}[k(x, y)]. \tag{5.3}$$

Troubleshooting i depuració

Incidència de laboratori: drift silenciós. Símptoma: les respostes continuen semblant raonables, però baixa *groundedness* i augmenta la distància MMD dels embeddings de consulta. El sistema no ha fallat de cop; ha sortit gradualment del règim validat.
Com detectar-ho: compareu finestres temporals, registreu MMD i PSI, i reviseu si els errors es concentren en consultes noves. Si només mireu exemples individuals, el drift pot passar desapercebut.
Acció d'enginyeria: reexecutar D4, revisar corpus, reindexar si cal i actualitzar el conjunt de consultes de regressió.

Checklist de verificació

Monitoritzeu drift de forma estratificada quan el sistema afecta col · lectius o segments diferents. Una distribució global estable pot amagar que un subgrup ha canviat molt més que la resta.

El report de drift ha d'indicar: finestra de referència, finestra de producció, variable o embedding avaluat, estadístic calculat, llindar i decisió. Sense finestra temporal explícita, no hi ha monitoratge reproduïble.

Microcas o incidència de laboratori: El corpus canvia sense reavaluació

Un equip afegeix documents nous al corpus del RAG i assumeix que el sistema millorarà. En realitat, els nous fragments introdueixen vocabulari i estructura diferents; el recuperador torna documents més recents però menys pertinents.

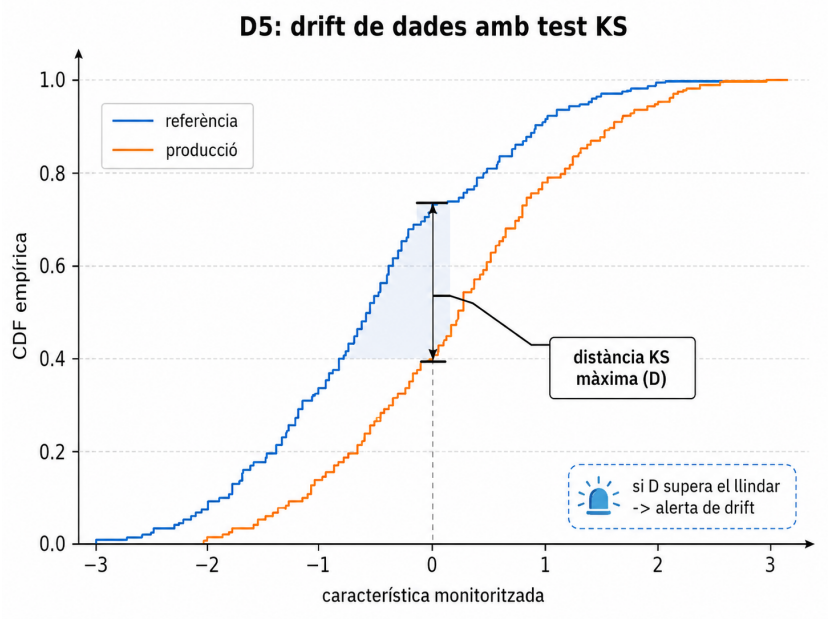

Figura 5.2: Comparació de distribucions amb KS. La diferència entre CDFs indica si una característica de producció s'ha allunyat de la referència.

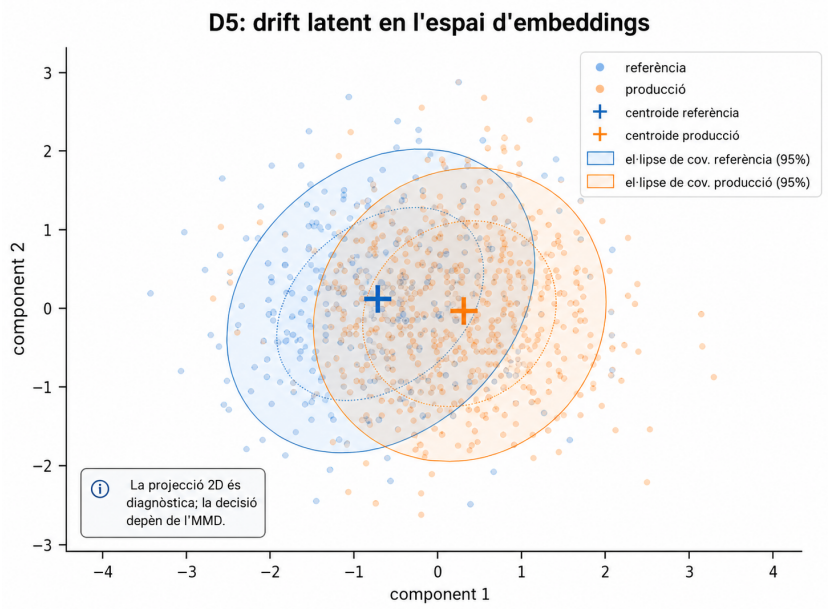

Figura 5.3: Drift latent en embeddings. La projecció 2D és només diagnòstica; el valor avaluable és l'MMD registrat al fitxer de monitoratge.

Microcas o incidència de laboratori: El corpus canvia sense reavaluació *(continuació)*

Acció d'enginyeria: reexecutar D4 i D5 després de cada canvi de corpus: recall@k, grounding, consultes negatives i drift d'embeddings.

5.3 Jutges automàtics: útils, però calibrats

Els sistemes generatius són difícils d'avaluar amb coincidència exacta. BLEU i ROUGE poden ser útils en traducció o resum extractiu, però són insuficients quan hi ha múltiples respostes

Taula 5.2: Ús responsable d'un LLM-as-a-Judge a D5

Ús	Quan és raonable	Risc
Triatge de respostes	Molts casos i revisió humana limitada	El jutge pot heretar biaixos o preferir estil sobre evidència
Avaluació de groundedness	Hi ha cites i fragments disponibles	Pot acceptar cites tangencials si el prompt és feble
Regressió de qualitat	Comparar versions del mateix sistema	Pot canviar de criteri si canvia el model jutge

vàlides. En D5 podem usar un jutge automàtic, però només com a instrument calibrat, no com a autoritat infal · lible.

Un jutge automàtic assigna una puntuació $s_j \in [0, 1]$ a una resposta. Per usar-lo en un gate cal comparar-lo amb revisions humanes o criteris coneguts. Una estimació simple és l'acord:

$$\text{Agreement} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}[\hat{y}_i^{judge} = y_i^{human}]. \tag{5.4}$$

La calibració demana una pregunta diferent: quan el jutge diu 0.8 de confiança, aproximadament un 80% d'aquests casos són realment acceptables?

Estratègia de validació

Test de calibratge del jutge. El jutge automàtic ha de validar-se amb una mostra petita de casos etiquetats manualment. El test no ha de demostrar que el jutge és perfecte; ha de detectar quan el jutge accepta cites tangencials o penalitza respostes correctes només per estil.

Criteri D5: si canvia el model jutge, el prompt o la rúbrica, les puntuacions històriques deixen de ser estrictament comparables. La versió del jutge forma part del contracte de monitoratge.

No useu un LLM-as-a-Judge per evitar pensar. El jutge ha de tenir rúbrica, exemples, llindars, mostra humana de control i versió fixada. Si el jutge canvia, les mètriques històriques deixen de ser comparables.

5.4 Active learning: convertir errors en cua de millora

Quan D5 detecta incertesa, drift o desacord entre jutge i humans, no hauria de limitar-se a produir una alerta. Ha de crear una cua de casos revisables. L'*active learning* prioritza exemples que poden millorar més el sistema: casos incerts, nous, representatius de subgrups afectats o amb desacord entre avaluadors.

Un score de prioritat simple pot combinar tres termes:

$$P(x) = \alpha U(x) + \beta N(x) + \gamma D(x), \tag{5.5}$$

on U és incertesa, N és novetat i D és desacord. Aquesta fórmula no pretén ser òptima; pretén convertir la revisió humana en una decisió traçable.

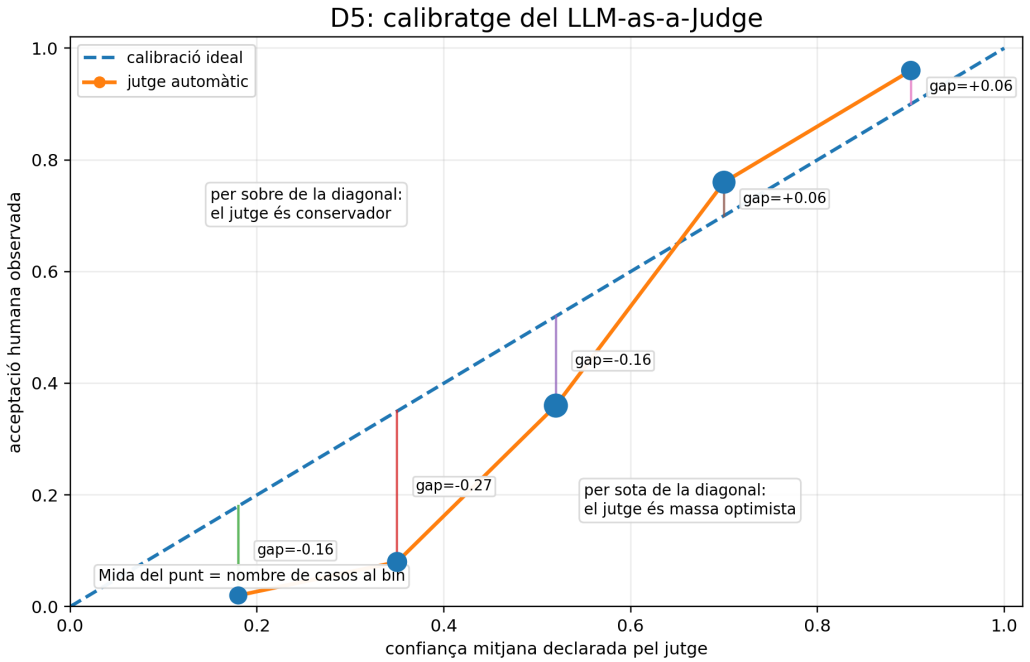

Figura 5.4: Calibratge del *LLM-as-a-Judge* en D5. La diagonal representa un jutge perfectament calibrat: quan declara una confiança mitjana del 70%, aproximadament el 70% dels casos haurien de ser acceptats per revisió humana. Els punts mostren bins de confiança i la seva mida indica el nombre de casos. Els segments verticals indiquen el gap de calibratge: si el punt queda per sota de la diagonal, el jutge és massa optimista; si queda per sobre, és massa conservador. Aquesta figura no valida el jutge per si sola, però permet decidir si cal ajustar prompts, llindars o revisió humana abans d'utilitzar-lo com a mètrica operativa.

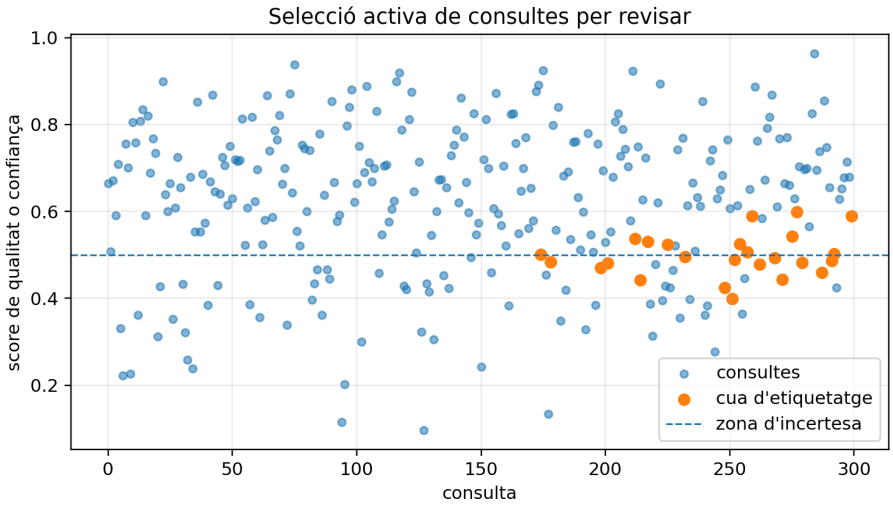

Figura 5.5: Cua d'active learning. Els casos seleccionats combinen incertesa, novetat i desacord per orientar la revisió humana.

Patró d'enginyeria

Una cua d'etiquetatge no ha de contenir només els casos més difícils. També ha d'incloure casos representatius dels segments on el sistema falla o on el drift és més fort. Si no, el bucle de millora pot amplificar biaixos.

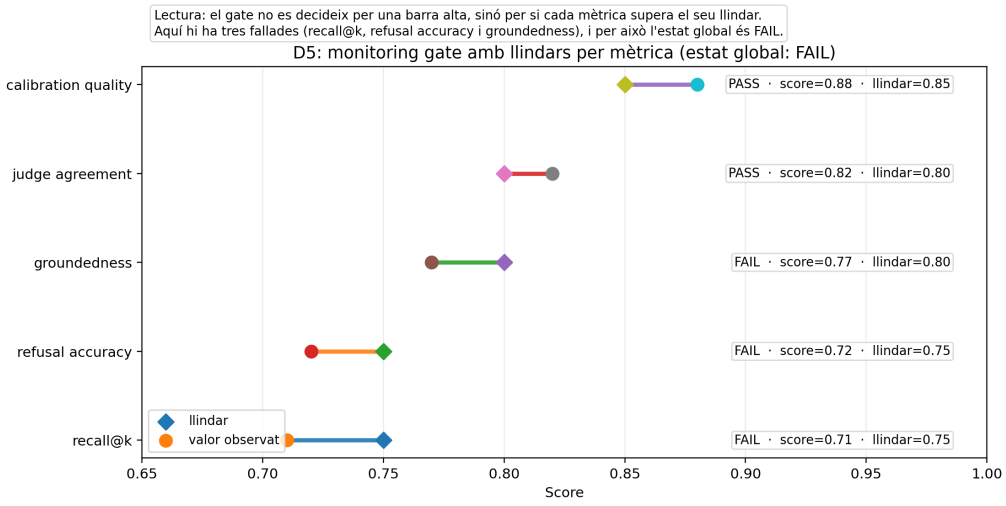

Figura 5.6: Monitoring gate de D5 amb llindars per mètrica. Cada fila compara el valor observat amb el llindar mínim exigible i indica si la mètrica passa o falla. El gate no es decideix per una impressió global ni per una única barra alta: l'estat global només és PASS si totes les mètriques crítiques superen els seus llindars. En aquest exemple, la finestra actual falla en `recall@k`, `refusal_accuracy` i `groundedness`; per això el resum global és FAIL i cal obrir una anàlisi de causes abans de continuar en producció.

5.5 Monitoring gate: decidir si el sistema continua sent defensable

D5 culmina en un gate. El gate no diu si el sistema és perfecte; diu si hi ha evidència suficient per continuar usant-lo, si cal advertència o si s'ha de bloquejar. Aquesta decisió combina regressió de qualitat, drift, calibratge del jutge i cua de revisió.

Pregunta de defensa

Defensa tècnica del D5. El grup ha de justificar una decisió PASS, WARN o FAIL indicant quina mètrica ha canviat, quin llindar s'ha vulnerat, quina acció proposa i quina evidència exigiria abans de tornar a activar el sistema.

```
python -m lic_project.report_d5 \
    --monitoring-dir reports/monitoring \
    --output reports/d5_report.md
```

Listing 5.3: Generació de l'informe tècnic D5

Per superar D5, l'informe ha de contenir una decisió clara: continuar, continuar amb advertiments, reindexar/reavaluar o bloquejar. Un monitoratge que només enumera mètriques sense decisió no compleix el contracte del capítol.

5.6 Conclusions: cap als sistemes amb acció

Aquest capítol ha convertit el sistema en un servei observat. La progressió és ara: D1 construeix baseline, D2 entrena, D3 audita, D4 recupera evidència i D5 monitoritza si tot això continua sent vàlid amb el temps. A partir d'ara, un sistema no és defensable només perquè va funcionar el dia de la demo; ha de demostrar que detecta quan deixa de funcionar.

Microcas o incidència de laboratori: Incidència de laboratori: alerta ignorada fins que és massa tard

Un equip detecta una caiguda sostinguda de groundedness i un drift latent en consultes noves, però manté el servei perquè les respostes continuen semblant fluïdes. Setmanes després, apareixen respostes amb cites tangencials i el sistema perd confiança.

Lliçó final del capítol: D5 no busca generar dashboards bonics, sinó evitar que un sistema aparentment correcte continuï operant quan les evidències indiquen degradació. El Capítol 6 haurà de tractar què passa quan el sistema no només respon, sinó que actua o coordina accions.

Tancament del D5

Artefacte lliurable

D5 — Monitoratge i regressió de qualitat. Al final del capítol heu de lliurar:

- execució de `python -m lic_project.monitoring_pipeline --config configs/d5_monitoring.yaml`;
- fitxers `reports/monitoring/d5_monitoring_metrics.json` i `d5_monitoring_summary.json`;
- figures de regressió, drift, MMD, calibratge del jutge, active learning i gate;
- informe `reports/d5_report.md`;
- tests mínims executats amb `pytest`;
- decisió final: `PASS`, `WARN` o `FAIL`, amb motius i acció recomanada.

Checklist de verificació

Abans de donar D5 per acabat, comproveu:

- el monitoratge s'executa amb una comanda documentada;
- totes les mètriques tenen llindars configurables;
- hi ha comparació amb una línia base D4;
- hi ha prova de drift tabular o latent;
- el jutge automàtic no s'usa sense calibratge;
- la cua d'etiquetatge és reproduïble;
- els tests passen;
- l'informe conté una decisió d'enginyeria, no només gràfics.

Criteris d'acceptació

Criteri	Punts
Pipeline de monitoratge executable i sense dependència d'un notebook	2.0
Regressió de qualitat respecte D4 amb llindars explícits	1.5
Detecció de drift amb KS/PSI/MMD i interpretació operativa	1.5
Calibratge mínim del jutge automàtic i limitacions declarades	1.5
Cua d'active learning amb criteri reproduïble	1.0
Tests mínims de mètriques i contracte	1.0
Informe tècnic D5 amb decisió justificada	1.5

Capítol 6

Sistemes amb Acció: agents, eines, guardrails i traçabilitat

Els capítols anteriors han construït una progressió d'enginyeria: D1 ha definit un baseline reproduïble, D2 ha entrenat un model profund, D3 ha convertit el model en un objecte auditable, D4 ha afegit recuperació documental amb cites i D5 ha introduït monitoratge, regressió de qualitat i gates de producció. En aquest capítol el sistema passa a ser un servei que pot executar eines, coordinar passos i modificar estat.

Aquesta transició és delicada. Un error en una resposta pot ser corregit; un error en una acció pot enviar un correu, modificar un registre, reservar un recurs, iniciar un procés o activar una decisió que afecti altres persones. Per això el capítol tracta l'agent com un contracte d'acció controlada: cada pas ha de tenir permisos, risc estimat, pressupost, registre, verificació i possibilitat d'escalat humà. El producte del capítol és el lliurable D6: un agent amb eines simulades, política de seguretat, traça d'execució, avaluació de trajectòries i gate agentic. Les figures són sortides del pipeline; el resultat avaluable són els fitxers de traces, mètriques, llindars i la decisió final del gate.

Repte d'enginyeria

Repte del capítol. Convertir el sistema monitoritzat D5 en un agent controlat que pot usar eines sense perdre traçabilitat ni seguretat.
Entregable associat: D6.
Artefactes mínims:

- configuració `configs/d6_agents.yaml`;
- orquestrador executable `src/lic_project/agent_pipeline.py`;
- registre d'eines disponibles, risc i reversibilitat;
- política de permisos, aprovació humana i bloqueig;
- traces d'execució a `reports/agents/d6_agent_traces.json`;
- mètriques d'avaluació a `reports/agents/d6_agent_metrics.json`;
- resum del gate a `reports/agents/d6_agent_summary.json`;
- informe `reports/d6_report.md`;
- figures generades pel pipeline;
- tests mínims de seguretat, contracte d'eines i regressió de trajectòries.

Laboratori de construcció:

En D6 no s'accepta un agent que simplement "fa coses". El professorat ha de poder veure quina observació ha rebut, quin pla ha proposat, quina eina volia cridar, quin gate ha passat, quin resultat ha obtingut i com s'ha registrat la decisió.
Un agent sense traça és pitjor que un script manual: pot semblar intel · ligent, però no és auditable. Si no podeu reconstruir per què una eina s'ha cridat, amb quins permisos i amb quin resultat, el sistema no compleix D6.

Artefacte lliurable

Contracte tècnic D6. En acabar el capítol, el professorat ha de poder executar:

```
python -m lic_project.agent_pipeline --config configs/d6_agents.yaml
python -m lic_project.report_d6 --agents-dir reports/agents --output reports/d6_report.md
pytest
```

Si alguna d'aquestes comandes falla, el capítol no s'ha completat com a laboratori d'enginyeria.

```
lic_project/
|-- configs/
|   |-- d1_baseline.yaml
|   |-- d2_training.yaml
|   |-- d3_audit.yaml
|   |-- d4_rag.yaml
|   |-- d5_monitoring.yaml
|   `-- d6_agents.yaml
|-- reports/
|   |-- agents/
|   |   |-- d6_agent_traces.json
|   |   |-- d6_agent_metrics.json
|   |   `-- d6_agent_summary.json
|   |-- figures/
|   `-- d6_report.md
|-- src/
|   `-- lic_project/
|       |-- agent_tools.py
|       |-- agent_safety.py
|       |-- agent_orchestrator.py
|       |-- agent_eval.py
|       |-- agent_pipeline.py
|       `-- report_d6.py
`-- tests/
    |-- test_agent_safety.py
    `-- test_agent_contract.py
```

6.1 De respondre a actuar: el contracte agentic

D4 i D5 han controlat què respon el sistema i si la qualitat es degrada. D6 afegeix una dimensió nova: l'acció. Un agent no és només un LLM amb instruccions; és un bucle que observa un estat, decideix un pas, crida una eina, interpreta el resultat i decideix si continua, s'atura o demana ajuda. Podem representar un agent controlat com una política limitada:

$$a_t \sim \pi_\theta(a \mid s_t, C, P), \tag{6.1}$$

on s_t és l'estat observat, C és el context recuperat o monitoritzat i P és la política de permisos. La diferència respecte a una formulació clàssica d'aprenentatge per reforç és que en D6 no busquem maximitzar una recompensa mitjançant exploració lliure. Busquem executar una trajectòria segura sota restriccions explícites. Una trajectòria agentic mínima és:

$$\tau = \{(o_t, p_t, a_t, r_t, v_t, \ell_t)\}_{t=1}^{T}, \tag{6.2}$$

on o_t és l'observació, p_t el pla, a_t l'acció o eina, r_t el resultat, v_t la verificació i ℓ_t el registre. El valor de D6 és que aquesta seqüència no queda amagada dins d'una conversa.

Taula 6.1: Del chatbot al sistema agentic D6

Element	Chatbot sense acció	Agent D6 controlat
Sortida	Text final	Trajectòria amb passos i eines
Evidència	Opcional o narrativa	Cites, resultats d'eina i log
Risc	Error de resposta	Error de resposta i error d'acció
Control	Prompt i instruccions	Política, permisos, HITL i rollback
Avaluació	Qualitat de resposta	Èxit de tasca, seguretat, cost i traces

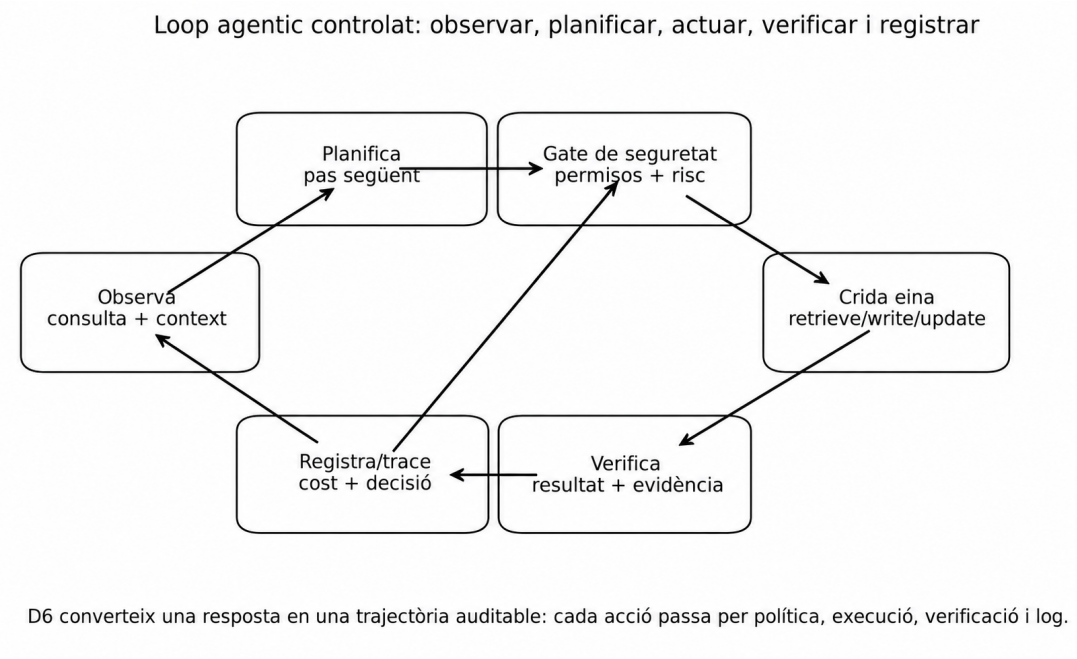

Figura 6.1: Loop agentic controlat. D6 converteix cada resposta en una seqüència observable: planificació, gate de seguretat, crida d'eina, verificació i registre.

Decisió d'arquitectura

Agent determinista o agent amb planificació oberta? Un flux determinista és menys flexible, però és més fàcil de testar, auditar i bloquejar. Un agent amb planificació oberta pot adaptar-se millor, però incrementa el risc d'accions inesperades i de costos no controlats.

Decisió D6: començar amb un agent controlat per una llista tancada d'eines, pressupost de passos i mode *dry-run*. La planificació oberta només és defensable si hi ha traça, permisos i criteri de rollback.

Laboratori de construcció: Observació, pla, acció i verificació

Objectiu: executar una primera trajectòria controlada. L'agent ha de rebre una tasca, recuperar evidència, proposar una acció de baix risc i escriure un esborrany sense enviar res automàticament.

```
python -m lic_project.agent_pipeline --config configs/d6_agents.yaml
```

Un agent de producció no ha de tenir accés directe a totes les eines. Ha de tenir una llista tancada d'eines, entrades tipades, permisos per tasca, pressupost de passos, mode *dry-run* quan calgui i registre immutable de cada crida.

Taula 6.2: Classificació d'eines en D6

Tipus d'eina	Exemple	Política recomanada
Lectura	`retrieve`, `search_logs`	Permetre amb registre i límit de consultes
Escriptura reversible	`write_draft`, `create_ticket`	Permetre si queda en estat revisable
Escriptura d'estat	`update_record`	Requerir aprovació o *dry-run*
Acció externa	`send_email`, `submit_form`	Bloquejar o exigir aprovació explícita

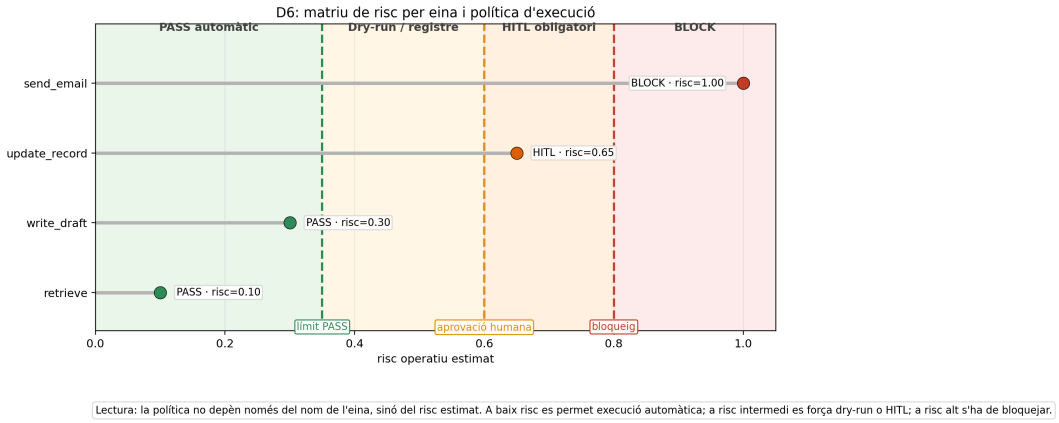

Figura 6.2: Matriu de risc per eina en D6. L'eix horitzontal representa el risc operatiu estimat i les franges de color defineixen la política d'execució: pas automàtic a baix risc, *dry-run* o registre obligatori a risc intermedi, aprovació humana (HITL) en accions sensibles i bloqueig en accions d'impacte extern o irreversible. Les línies verticals indiquen els llindars de decisió i cada eina es posiciona segons la seva política resultant.

Laboratori de construcció: Observació, pla, acció i verificació *(continuació)*

No confongueu planificar amb executar. En D6, una proposta d'acció pot ser correcta i alhora no estar autoritzada. La decisió final no la pren el text generat, sinó el gate d'acció.

6.2 Eines, permisos i reversibilitat

Una eina és una funció amb efectes. Pot ser innòcua, com recuperar documents; reversible, com crear un esborrany; o irreversible, com enviar un correu o modificar una base de dades. El contracte D6 obliga a classificar les eines abans d'usar-les.

Definim un score de risc operatiu:

$$R(a) = w_c C(a) + w_i I(a) + w_r(1 - Rev(a)) + w_p P(a), \tag{6.3}$$

on $C(a)$ és cost potencial, $I(a)$ és impacte extern, $Rev(a)$ indica reversibilitat i $P(a)$ penalitza permisos insuficients. Una política simple pot ser:

$$\begin{aligned} R(a) < \tau_{\text{pass}} &\Rightarrow \text{executar}, \\ \tau_{\text{hitl}} \leq R(a) < \tau_{\text{block}} &\Rightarrow \text{aprovació humana}, \\ R(a) \geq \tau_{\text{block}} &\Rightarrow \text{bloquejar}. \end{aligned} \tag{6.4}$$

Troubleshooting i depuració

Incidència de laboratori: eina correcta, permís incorrecte. Símptoma: l'agent tria una eina pertinent, però la crida amb un nivell de permís massa alt o sense aprovació humana. El problema no és de raonament, sinó de contracte d'eina.
Com detectar-ho: cada eina ha de declarar risc, reversibilitat, esquema d'entrada i política d'aprovació. Un test ha de simular una acció d'alt risc i comprovar que queda bloquejada o passa a HITL.
Acció d'enginyeria: separar planificació de permisos. Que l'agent proposi una acció no implica que el sistema tingui dret a executar-la.

Laboratori de construcció: Eina segura amb *dry-run*

Objectiu: definir almenys tres eines: una de lectura, una d'escriptura reversible i una d'acció externa. L'agent ha d'executar les dues primeres quan la tasca ho permeti i bloquejar l'acció externa si no hi ha aprovació humana.

```
tools:
  retrieve:
    risk: 0.10
    reversible: true
  write_draft:
    risk: 0.30
    reversible: true
  send_email:
    risk: 0.85
    reversible: false
```

El contracte d'eina ha d'indicar: nom, entrada, sortida, risc, reversibilitat, permisos i si admet *dry-run*. Una eina sense contracte no s'ha d'exposar a l'agent.

Microcas o incidència de laboratori: L'agent envia massa aviat.

Un grup construeix un agent que resumeix una alerta D5 i envia automàticament un correu d'escalat. El contingut és raonable, però l'alerta era un fals positiu. El problema no és el resum: és que l'eina d'enviament no tenia aprovació humana.
Acció d'enginyeria: separar `write_draft` de `send_email`, bloquejar l'enviament automàtic i registrar qui aprova l'acció.
Lliçó per al D6: una eina externa converteix un error de model en un error operatiu.

6.3 Guardrails, HITL i rollback

Els guardrails no són frases al prompt. Són comprovacions executables abans i després de l'acció. En D6, el gate comprova permisos abans de cridar l'eina i verifica resultat després de rebre'n la sortida. Un esquema mínim és:

1. validar que la tasca permet l'eina;
2. calcular risc i reversibilitat;
3. decidir `PASS`, `NEEDS_APPROVAL` o `BLOCK`;
4. executar només si la política ho permet;
5. registrar entrada, sortida i decisió;
6. comprovar si cal rollback o revisió humana.

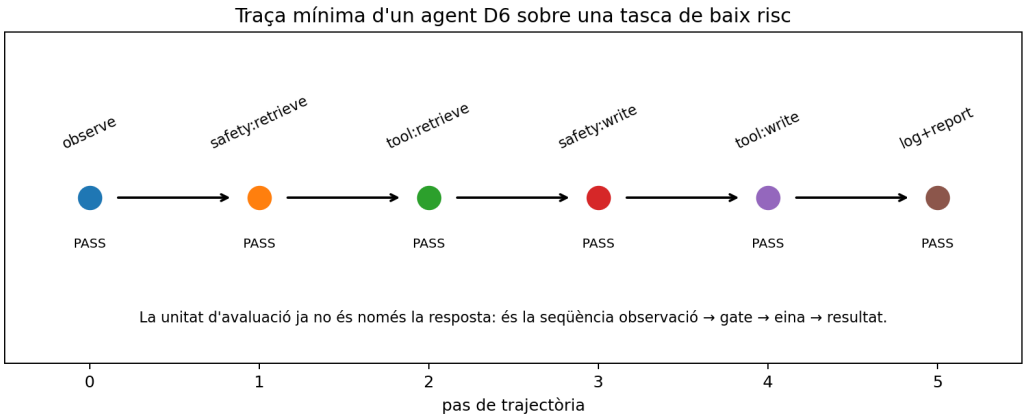

Figura 6.3: Traça mínima d'una tasca D6. Cada pas indica fase, eina, estat i risc. Aquesta traça és el material auditable del capítol.

Estratègia de validació

Test de regressió de trajectòries. Per a D6 no n'hi ha prou amb comparar resposta final. Cal guardar trajectòries esperades: observació, pla, eina, resultat i verificació. El test ha de detectar si una nova versió crida una eina diferent, salta el gate o omet la verificació final.

Què protegeix? Evita que una millora aparent de fluïdesa empitjori seguretat operativa.

Laboratori de construcció: Bloqueig d'accions no autoritzades

Objectiu: provar una tasca que demana una acció externa no autoritzada. El sistema ha de bloquejar-la i deixar una traça que expliqui el motiu del bloqueig.

```
python -m lic_project.agent_pipeline --config configs/d6_agents.yaml
cat reports/agents/d6_agent_traces.json
```

No accepteu una resposta del tipus "he decidit no enviar-ho" si el codi no conté un gate que impedeixi l'enviament. La seguretat no pot dependre només de la bona voluntat del model.
Human-in-the-loop no vol dir "preguntar sempre". Vol dir demanar aprovació només quan el risc, l'impacte o la irreversibilitat ho exigeixen. Si tot requereix humans, l'agent no aporta valor; si res requereix humans, l'agent és perillós.

6.4 Avaluació d'agents: trajectòries, no només respostes

Un agent s'avalua per trajectòries. Pot arribar a la resposta correcta fent massa passos, usant una eina prohibida, gastant massa cost o amagant un error d'eina. Per això D6 afegeix mètriques específiques:

- **task success:** percentatge de tasques completades segons el resultat esperat;
- **unsafe action rate:** percentatge d'accions insegures no bloquejades;
- **unapproved high-risk actions:** accions d'alt risc executades sense aprovació;
- **mean steps:** longitud mitjana de trajectòria;
- **tool error rate:** errors d'eina no gestionats.

$$\text{UnsafeRate} = \frac{\#\{\text{accions insegures executades}\}}{\#\{\text{accions proposades}\}}. \tag{6.5}$$

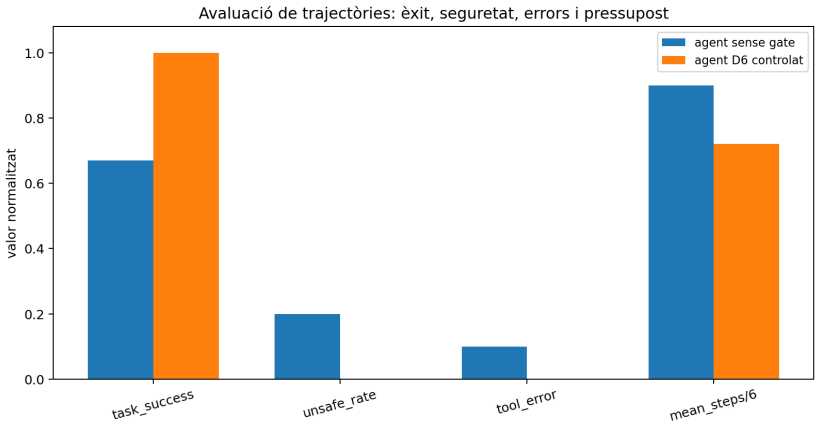

Figura 6.4: Avaluació de trajectòries. Un agent controlat pot ser menys agressiu, però ha de reduir accions insegures i errors d'eina sense destruir l'èxit de tasca.

Costos i eficiència

Pressupost de passos i cost operatiu. Cada pas agentic pot consumir temps, tokens, crides a eines o revisió humana. Un agent que resol la tasca en deu passos quan un flux determinista ho faria en dos pot ser tècnicament interessant i operacionalment inadequat.
Circuit breaker agentic: límit dur de passos, eines i cost. Un agent no pot tenir una targeta de crèdit oberta. Cada trajectòria ha de tenir límits durs abans d'executar-se: nombre màxim d'iteracions, nombre màxim de crides a eines, temps màxim i pressupost màxim estimat. Si arriba al límit sense resoldre la tasca, no ha de continuar raonant: ha de retornar `STOPPED_BY_CIRCUIT_BREAKER` i deixar traça.
Configuració mínima D6:

- `max_iterations: 5`
- `max_tool_calls: 8`
- `max_repeated_tool_calls: 2`
- `timeout_seconds: 30`
- `max_estimated_cost: 0.25`

Risc si falta. Un agent ReAct encallat pot repetir la mateixa eina, acumular cost, saturar un servei extern o generar una traça impossible d'auditar. El circuit breaker no és una optimització: és un requisit de seguretat i FinOps.
Regla D6: registreu nombre de passos, eines cridades, accions bloquejades, revisions humanes i motiu d'aturada. Aquestes mètriques formen part del gate agentic.

Laboratori de construcció: Regressió de trajectòries

Objectiu: crear un conjunt petit de tasques de prova i comprovar que el comportament de l'agent no canvia silenciosament quan es modifica el prompt, la política o el registre d'eines.

```
python -m lic_project.agent_pipeline --config configs/d6_agents.yaml
python -m lic_project.report_d6 --agents-dir reports/agents --output reports/d6_report.md
pytest
```

El test de regressió ha de comprovar com a mínim tres casos: una tasca permesa, una tasca que requereix aprovació i una tasca que s'ha de bloquejar. Si un canvi de prompt altera aquest comportament, el test ha de fallar.

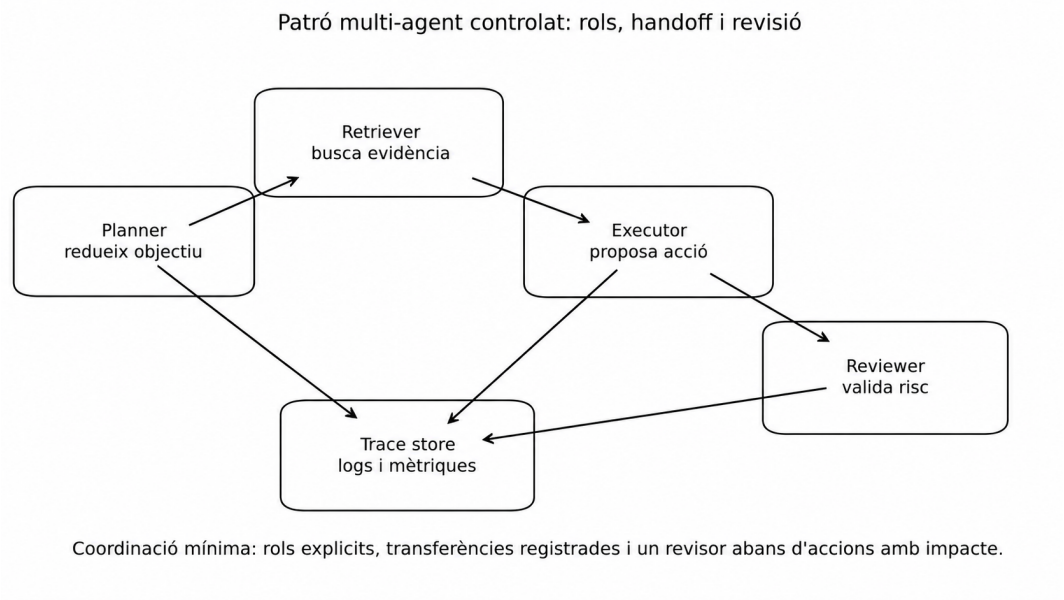

Figura 6.5: Coordinació multi-agent controlada. El valor no és tenir molts agents, sinó separar responsabilitats i registrar cada transferència.

6.5 Coordinació multi-agent mínima: rols abans que autonomia

El text original del capítol tractava sistemes multi-agent i teoria de jocs com a bloc central. Són idees importants, però en un laboratori de grau no han de substituir el primer objectiu: construir un sistema amb accions controlades. Per això D6 introdueix només una coordinació mínima basada en rols. Un patró suficient és:

- **Planner:** descompon la tasca i proposa passos;
- **Retriever:** recupera evidència o estat;
- **Executor:** prepara l'acció;
- **Reviewer:** valida risc, permisos i resultat;
- **Trace store:** conserva logs i mètriques.

No creeu cinc agents si un flux seqüencial amb un revisor és suficient. La coordinació afegeix complexitat: més latència, més punts de fallada i més difícil atribució de responsabilitat. Afegiu rols només quan redueixin risc o millorin verificació.

Pregunta de defensa

Defensa tècnica del D6. El grup ha d'explicar quina acció pot executar l'agent, quina acció queda bloquejada, quan s'activa HITL, com es pot reconstruir una traça i quin rollback és possible si la decisió resulta incorrecta.
D6 culmina en un gate. Aquest gate no decideix si l'agent és "intel · ligent"; decideix si és operativament acceptable. Combina èxit de tasca, seguretat, accions d'alt risc, errors d'eina i longitud de trajectòria.

```
python -m lic_project.report_d6 --agents-dir reports/agents --output reports/d6_report.md
```

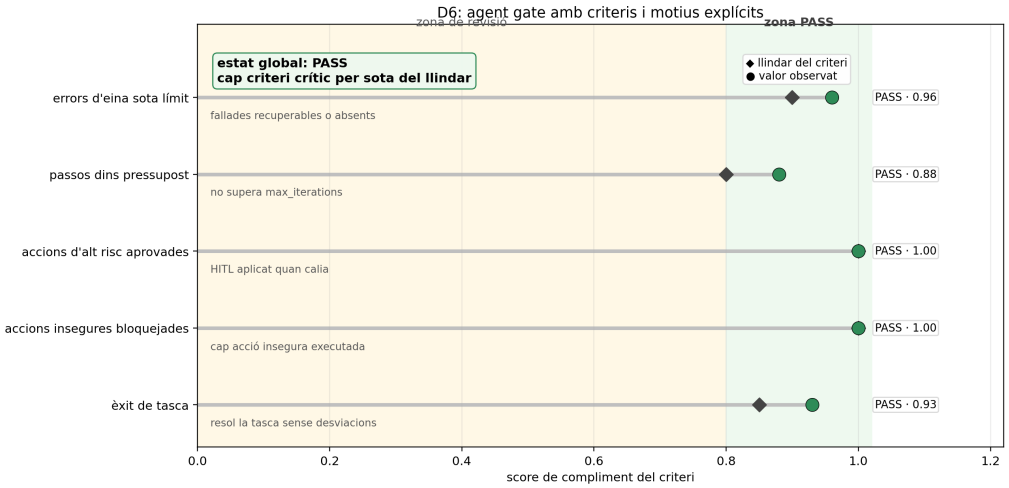

Figura 6.6: Agent gate D6. El resultat acceptable és un resum `PASS`, `WARN` o `FAIL` amb motius explícits i acció recomanada.

Microcas o incidència de laboratori: Agents que es donen la raó

Un equip crea un planner, un executor i un reviewer, però tots comparteixen el mateix prompt i la mateixa informació. El reviewer legitima el pla inicial i no detecta cap risc.
Acció d'enginyeria: donar al reviewer una rúbrica diferent, accés al registre de permisos i capacitat real de bloqueig.
Lliçó per al D6: un reviewer sense criteris i sense poder de bloqueig és decoració.

Estratègia de validació

Test obligatori de circuit breaker.
El repositori ha d'incloure una tasca impossible o mal especificada que forci l'agent a arribar al límit d'iteracions. El test ha de comprovar que l'agent s'atura, que no executa accions externes després del límit i que la traça conté el motiu de parada.

Què hauria de fallar? El test hauria de fallar si l'agent continua cridant eines després de `max_iterations`, si repeteix indefinidament la mateixa eina o si no registra `STOPPED_BY_CIRCUIT_BREAKER`.
Per superar D6, l'informe ha de contenir una decisió clara: permetre execució controlada, permetre només en *dry-run*, exigir aprovació humana o bloquejar l'agent. Un agent que només mostra exemples d'ús sense gate no compleix el contracte del capítol.

6.6 Conclusions: cap a la integració final

Aquest capítol ha convertit el sistema monitoritzat en un agent amb accions controlades. La progressió queda així: D1 construeix baseline, D2 entrena, D3 audita, D4 recupera evidència, D5 monitoritza degradació i D6 actua amb guardrails. A partir d'ara, el projecte ja no es pot valorar només per si respon bé, sinó per si sap actuar sense perdre seguretat, traçabilitat i responsabilitat.

Microcas o incidència de laboratori:

Incidència de laboratori: agent brillant però ingovernable. Un equip presenta un agent que resol tasques complexes amb molta fluïdesa, però no guarda traces, no separa eines reversibles d'irreversibles i no té cap prova que bloquegi accions no autoritzades. El resultat és impressionant en demo i indefensable en producció.

Microcas o incidència de laboratori: (continuació)

Lliçó final del capítol: D6 no busca l'agent més autònom, sinó el primer agent que pot actuar amb límits, evidència, responsabilitat i proves. La integració final del LIC haurà d'unificar D1–D6 en un sistema complet: reproduïble, entrenat, auditat, documentat, monitoritzat i capaç d'actuar sense perdre control.

Tancament del D6

Artefacte lliurable

D6 — Agent amb eines, guardrails i traçabilitat. Al final del capítol heu de lliurar:

- execució documentada del pipeline D6 amb configuració externa;
- fitxers `d6_agent_traces.json`, `d6_agent_metrics.json` i `d6_agent_summary.json`;
- figures de loop agentic, risc d'eines, traça, avaluació de trajectòries, handoff multi-agent i gate;
- informe `reports/d6_report.md`;
- tests mínims executats amb `pytest`;
- decisió final: `PASS`, `WARN` o `FAIL`, amb motius i acció recomanada.

Checklist de verificació

Abans de donar D6 per acabat, comproveu:

- l'agent s'executa amb una comanda documentada;
- cada eina té risc, permisos i reversibilitat declarats;
- les accions d'alt impacte requereixen aprovació o queden bloquejades;
- totes les trajectòries queden registrades;
- hi ha almenys una tasca permesa, una amb aprovació i una bloquejada;
- els tests passen;
- l'informe conté una decisió d'enginyeria, no només una demo.

Criteris d'acceptació

Criteri	Punts
Pipeline agentic executable i sense dependència d'un notebook	2.0
Registre d'eines, permisos, risc i reversibilitat	1.5
Guardrails amb HITL, bloqueig i mode *dry-run*	1.5
Traces d'execució auditable per tasca	1.5
Avaluació de trajectòries i regressió de comportament	1.0
Tests mínims de seguretat i contracte agentic	1.0
Informe tècnic D6 amb decisió justificada	1.5

Capítol 7

Integració Final: release, defensa i portafoli d'evidències

Els capítols anteriors han construït una progressió completa d'enginyeria. D1 ha establert un baseline reproduïble; D2 ha entrenat un model profund amb checkpoint; D3 ha auditat el model; D4 ha afegit recuperació documental amb cites; D5 ha introduït monitoratge i regressió de qualitat; i D6 ha fet del sistema un agent controlat amb eines, permisos, traces i gates.

Aquest capítol converteix tot el que s'ha construït en un **release defensable**. Un sistema és defensable quan es pot executar, explicar, auditar, demostrar i criticar amb evidència. Per tant, D7 tanca el projecte transversal amb una matriu de traçabilitat, un registre de riscos, un informe final, una demo i una defensa tècnica.

Repte d'enginyeria

Repte del capítol. Integrar D1–D6 en un portafoli tècnic final i decidir si el sistema pot defensar-se com a release de laboratori.
Entregable associat: D7.
Artefactes mínims:

- matriu de traçabilitat entre requisits, lliurables i evidències;
- registre de riscos amb estat, mitigació i propietari;
- informe final `reports/d7_final_report.md`;
- release summary amb decisió `PASS`, `WARN` o `FAIL`;
- figures generades pel pipeline;
- guió de defensa i demo reproduïble;
- tests mínims del contracte de release.

Laboratori de construcció:

D7 no accepta un conjunt de captures de pantalla ni una carpeta de notebooks com a resultat final. El professorat ha de poder executar el validador de release, revisar el registre de riscos, regenerar les figures i entendre quines decisions d'enginyeria justifiquen l'estat final del projecte.
Un projecte que funciona en una demo però no té traçabilitat no és un projecte final defensable. En D7, la pregunta és "es pot reproduir, auditar i explicar quan falla?"

Artefacte lliurable

Contracte tècnic D7. En acabar el capítol, el professorat ha de poder executar:

```
python -m lic_project.final_report \
    --config configs/d7_final.yaml \
    --output reports/d7_final_report.md
python -m lic_project.validate_release --config configs/d7_final.yaml
pytest
```

Artefacte lliurable *(continuació)*

Si alguna d'aquestes comandes falla, el projecte no està tancat com a laboratori d'enginyeria.

```
lic_project/
|-- configs/
|   |-- d1_baseline.yaml
|   |-- d2_training.yaml
|   |-- d3_audit.yaml
|   |-- d4_rag.yaml
|   |-- d5_monitoring.yaml
|   |-- d6_agents.yaml
|   `-- d7_final.yaml
|-- reports/
|   |-- final/
|   |   |-- d7_traceability_matrix.json
|   |   |-- d7_risk_register.json
|   |   `-- d7_release_summary.json
|   |-- figures/
|   `-- d7_final_report.md
|-- src/lic_project/
|   |-- final_data.py
|   |-- final_metrics.py
|   |-- final_report.py
|   `-- validate_release.py
`-- tests/
    |-- test_traceability.py
    |-- test_release_gate.py
    `-- test_final_report_contract.py
```

Listing 7.1: Estructura incremental D7 sobre el repositori D1–D6

7.1 Del projecte funcional al release defensable

Un projecte de Machine Learning o d'IA generativa pot semblar correcte perquè una demo concreta surt bé. Però això no és suficient. En enginyeria, el valor no és només produir una sortida convincent, sinó conservar el camí que permet reproduir-la, diagnosticar-la i bloquejar-la si deixa de ser fiable.

D7 introdueix la idea de **release defensable**: una versió del sistema que té codi, configuració, mètriques, proves, riscos, informe i criteris d'acceptació. El canvi respecte als capítols anteriors és que ja no s'avalua un component aïllat. S'avalua la coherència del sistema complet.

Integrar no vol dir posar tots els scripts en una mateixa carpeta. Integrar vol dir que cada peça té una funció clara dins d'un contracte de qualitat:

- D1 defineix el primer baseline i el protocol experimental;
- D2 afegeix entrenament profund i checkpoint;
- D3 comprova equitat, explicabilitat i robustesa;
- D4 recupera evidència documental i aplica abstenció;
- D5 monitoritza regressió i drift;
- D6 controla accions, eines i traces;
- D7 decideix si el conjunt és defensable.

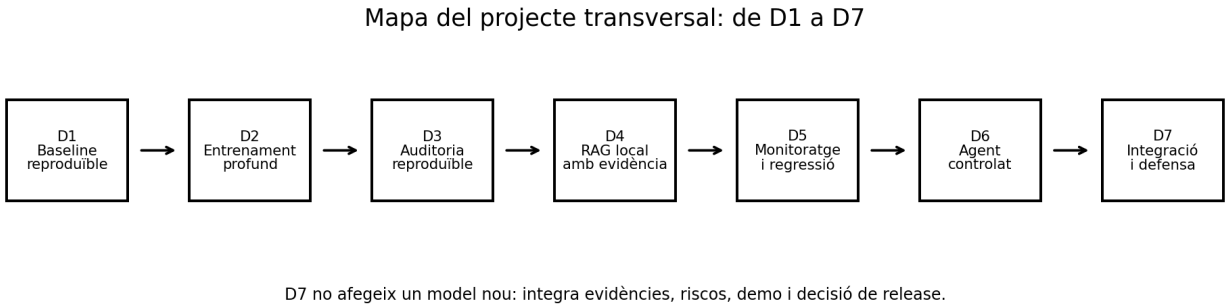

Figura 7.1: Mapa del projecte transversal D1–D7. El capítol final no afegeix un model nou: integra evidències i converteix el projecte en un release defensable.

Decisió d'arquitectura

Release acadèmic o release professional mínim? Un lliurable acadèmic pot funcionar en una demo. Un release professional mínim ha de ser reproduïble, auditable i revisable per algú que no ha escrit el codi. La diferència no és estètica: és la presència d'evidències, versions, riscos i criteris de bloqueig.

Decisió D7: no es tanca el projecte fins que cada decisió important apunta a una evidència D1–D6: mètriques, informes, tests, figures o registres.

Laboratori de construcció: Generar el primer release summary

Objectiu: generar l'informe final D7, la matriu de traçabilitat, el registre de riscos i la decisió de release a partir d'una configuració externa. El resultat no és un text escrit a mà: és un informe generat des de dades de projecte.

```
python -m lic_project.final_report \
    --config configs/d7_final.yaml \
    --output reports/d7_final_report.md
python -m lic_project.validate_release --config configs/d7_final.yaml
```

Listing 7.2: Execució del release final D7

Un bon release no és el que no té riscos. És el que declara els riscos, els associa a evidències i defineix què es farà si es materialitzen. La diferència professional entre una demo i un sistema defensable és la traçabilitat.
Per superar aquesta part, el fitxer `reports/final/d7_release_summary.json` ha d'indicar `PASS`, `WARN` o `FAIL`, i l'informe ha d'explicar els motius. Una conclusió manual que no surt del pipeline no és vàlida com a evidència D7.

7.2 Matriu de traçabilitat: requisits, evidències i buits

La matriu de traçabilitat és l'eina que evita que el projecte final sigui una narració selectiva. Obliga a respondre una pregunta simple: per a cada lliurable, quina evidència existeix i quina falta?

Si una dimensió important no té evidència, el problema no s'ha de maquillar. S'ha de declarar com a buit, risc o millora pendent. Aquesta disciplina és especialment important en sistemes amb IA generativa, on és fàcil confondre una resposta fluent amb un comportament controlat.

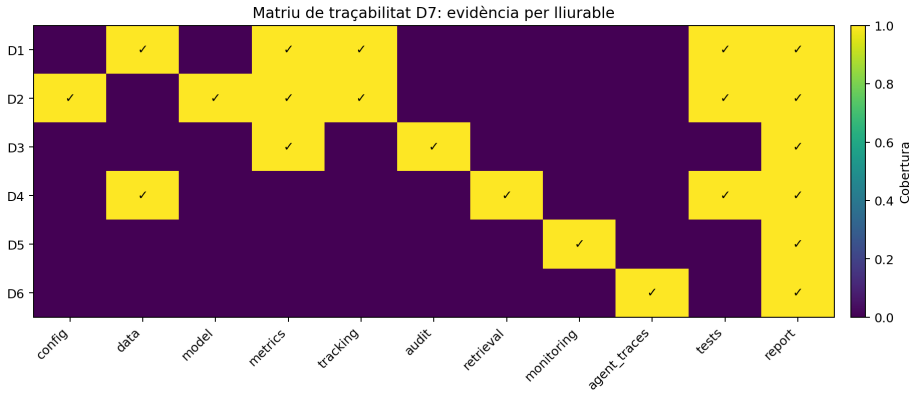

Figura 7.2: Matriu de traçabilitat D7. Cada marca indica que un lliurable aporta evidència en una dimensió del sistema: dades, model, mètriques, tracking, auditoria, recuperació, monitoratge, traces, tests o informe.

Taula 7.1: Lectura operativa de la matriu de traçabilitat

Dimensió	**Pregunta de defensa**	**Evidència mínima**
Configuració	Es pot repetir l'experiment?	Fitxers YAML versionats i seeds declarats.
Mètriques	La decisió està justificada?	JSON de mètriques i informe associat.
Auditoria	S'han mesurat riscos socials o tècnics?	Mètriques per subgrup, robustesa i audit gate.
Recuperació	Les respostes tenen suport documental?	Chunks, cites, scores i política d'abstenció.
Monitoratge	Es detecta degradació amb el temps?	Drift, regressió i gate D5.
Traces d'agent	Es pot reconstruir una acció?	Trajectòries, permisos, resultats i decisió HITL.
Tests	El contracte s'ha automatitzat?	Tests que fallen si el pipeline incompleix requisits.

Estratègia de validació

Test de matriu de traçabilitat. La matriu final ha de detectar requisits sense evidència. Un test simple pot llegir el registre de requisits i fallar si algun ítem obligatori no té artefacte associat o si l'artefacte referenciat no existeix.

Aquest test no substitueix la revisió docent, però evita el problema més habitual del portafoli: afirmar que s'ha fet una auditoria, un gate o un monitoratge sense deixar rastre verificable.

En una assignatura de laboratori, la rúbrica no ha de quedar separada del repositori. El repositori ha de contenir proves que responen a la rúbrica. Això transforma l'avaluació: l'alumnat no diu "he treballat molt", sinó que mostra evidències executables.

No ompliu la matriu amb optimisme. Una cel · la buida és útil si revela una limitació real. El que penalitza en D7 no és tenir limitacions; és no haver-les detectat o no saber explicar-les.

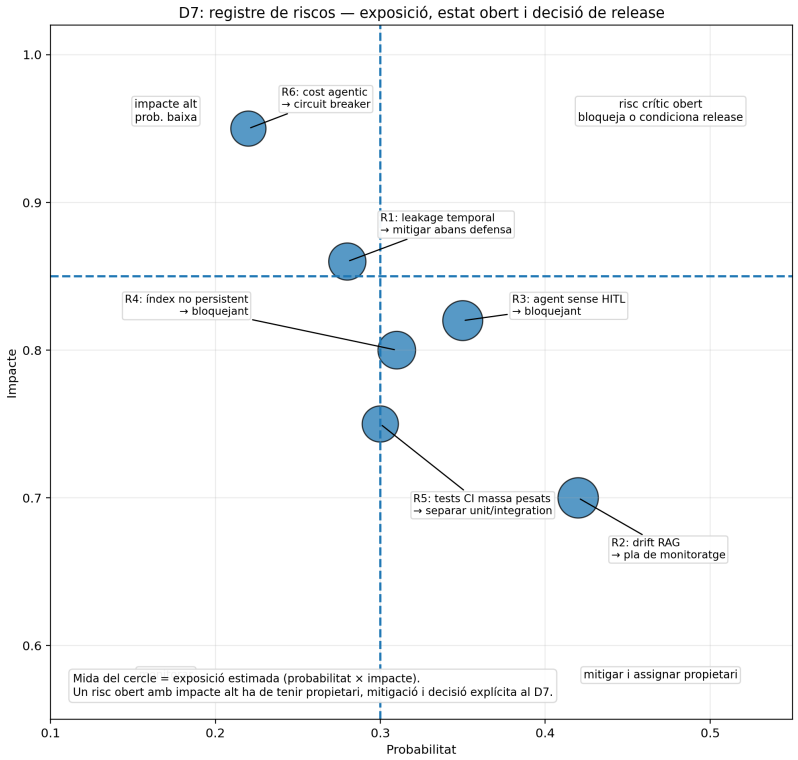

Figura 7.3: Registre de riscos D7. Els riscos amb impacte alt i estat obert han de bloquejar o condicionar la defensa del release.

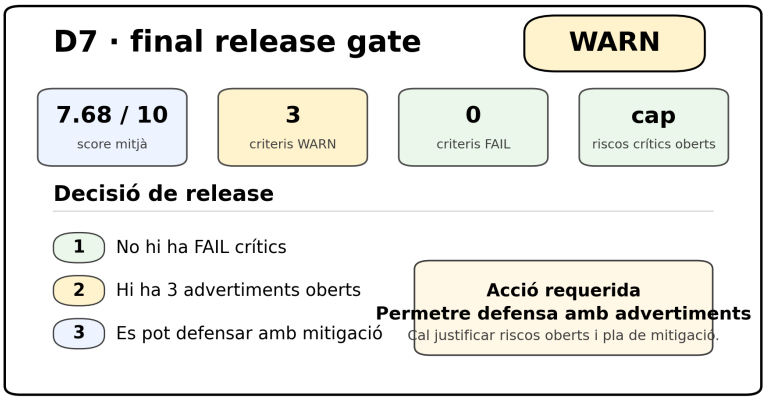

Figura 7.4: Final release gate D7. El resultat acceptable no és necessàriament `PASS`; també pot ser `WARN` si les limitacions són explícites i defensables.

7.3 Registre de riscos i decisió de release

El registre de riscos converteix les limitacions del projecte en decisions accionables. Cada risc ha de tenir identificador, probabilitat, impacte, estat, mitigació i propietari. Aquesta taula és la manera de mostrar que l'equip entén què pot fallar i què faria si passés.

Un release final no necessita fingir perfecció. Necessita una decisió honesta:

- **PASS**: el sistema és defensable amb els riscos actuals controlats;
- **WARN**: el sistema es pot defensar, però cal declarar riscos oberts i pla de mitigació;
- **FAIL**: hi ha riscos crítics oberts o lliurables bàsics que no funcionen.

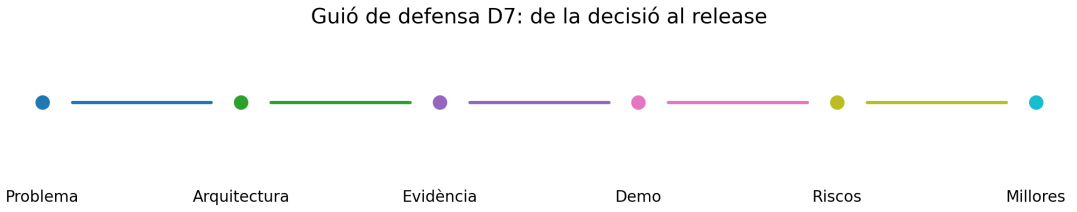

Figura 7.5: Guió recomanat de defensa D7: problema, arquitectura, evidència, demo, riscos i millores.

Troubleshooting i depuració

Incidència de laboratori: release amb riscos sense propietari. Símptoma: l'informe enumera riscos, però cap risc té responsable, severitat, evidència, mitigació o decisió. Això converteix el registre en una llista decorativa.

Acció d'enginyeria: cada risc ha de tenir estat, responsable tècnic, criteri de tancament i connexió amb un artefacte. Si un risc crític queda obert, el release no pot ser PASS sense justificació explícita.

```
python -m lic_project.validate_release --config configs/d7_final.yaml
```

Listing 7.3: Validació automàtica del release

Un `WARN` honest pot ser millor que un `PASS` fictici. En un projecte acadèmic, declarar una limitació crítica i proposar una mitigació raonable demostra més maduresa que ocultar-la darrere d'una demo fluida.

7.4 Scorecard final i defensa tècnica

La defensa final ha de ser una narració tècnica, no una visita turística pel codi. Ha de respondre cinc preguntes: quin problema resol el sistema, quina arquitectura heu triat, quina evidència suporta la decisió, quins riscos queden oberts i què faríeu en la següent iteració.

Pregunta de defensa

Defensa final: del resultat a la decisió de release.
La defensa no ha de seguir necessàriament l'ordre dels capítols, sinó l'ordre d'una decisió d'enginyeria: problema, arquitectura, evidència, demo, riscos i millores. Cada diapositiva o apartat ha de respondre tres preguntes: què heu decidit, amb quina evidència ho sosteniu i quin risc continua obert.

Abans de defensar el release, l'equip ha d'identificar els lliurables forts i els lliurables que requereixen justificació addicional. No n'hi ha prou amb tenir una puntuació global acceptable: cal explicar quins D tenen evidència sòlida, quins queden en zona `WARN` i quina acció de mitigació s'ha previst.

Preguntes mínimes de defensa:

- Quin lliurable és el punt més fort del projecte i quina evidència ho demostra?
- Quin lliurable és el més feble i quin risc introdueix al release?
- Quin criteri us faria bloquejar el release encara que la demo funcionés?
- Quina millora prioritzaríeu si tinguéssiu una setmana més?

Checklist de verificació

La defensa final ha de cobrir:

- **Problema**: quin risc o necessitat aborda el sistema;
- **Arquitectura**: com es connecten D1–D6;
- **Evidència**: mètriques, auditories, cites, traces i tests;
- **Demo**: una execució curta, reproduïble i preparada;
- **Limitacions**: què no funciona encara o què pot fallar;
- **Millores**: què faria l'equip amb una iteració més.

No convertiu la defensa en una llista de llibreries. Dir que heu usat PyTorch, Weights & Biases, FastAPI o Docker no explica cap decisió. La defensa ha d'explicar per què cada eina resol un risc o habilita una evidència.

7.5 Portafoli professional: del lliurable acadèmic a la prova de competència

El resultat de LIC no hauria de morir en una carpeta del campus virtual. Un bon D7 pot transformar-se en un portafoli professional: README clar, comandes reproduïbles, informe de riscos, captures mínimes de W&B, figures regenerables i decisions documentades.

Així, una persona externa no necessita llegir tots els notebooks, sinó entendre:

- quin problema resol el projecte;
- quina és l'arquitectura general;
- com s'executa el pipeline;
- quines mètriques s'han obtingut;
- quins riscos s'han detectat;
- què està automatitzat amb tests;
- quina decisió final pren l'equip.

Microcas o incidència de laboratori: Incidència de laboratori: demo brillant, release feble

Un equip prepara una demo visualment molt convincent: el RAG respon bé a tres preguntes, l'agent genera un esborrany i les figures són atractives. Però quan el professorat intenta executar el projecte, falta una dependència, el checkpoint no està versionat i l'informe no declara cap risc.
Acció d'enginyeria: bloquejar el release fins que el repositori tingui configuració, comandes reproduïbles, informe D7 i validació automàtica.
Lliçó per al D7: una demo pot mostrar potencial, però només el release mostra maduresa d'enginyeria.

7.6 Tancament del D7 i del llibre

Aquest llibre ha començat amb una decisió senzilla: no acceptar gràfics aïllats com a substitut d'enginyeria. D1 ha imposat baseline, split i tracking. D2 ha exigit entrenament profund reproduïble. D3 ha convertit l'ètica i la robustesa en auditories executables. D4 ha exigit evidència documental. D5 ha monitoritzat degradació. D6 ha controlat accions. D7 tanca el cercle: el sistema complet només és defensable si totes aquestes peces deixen rastre.

El missatge final és pràctic: l'enginyeria de sistemes intel · ligents no consisteix a confiar en models cada vegada més grans, sinó a construir sistemes que saben mostrar què han fet, amb quina evidència, sota quins límits i amb quins riscos oberts.

Decisió d'arquitectura

Decisió de release: què entra al CI i què queda com a prova d'integració?
Un release professional no executa sempre tots els tests pesats en cada commit. La decisió correcta és separar una bateria ràpida i obligatòria d'una més lenta i opcional o programada.
Contracte mínim D7:

- `pytest -m "not integration"` ha de passar en local i en CI sense GPU;
- `pytest -m integration` pot carregar models, índexs o serveis externs, però ha d'estar separat;
- l'API s'ha de poder testejar amb un model dummy;
- cap test unitari ha de dependre d'una clau d'API, d'un model de GBs o d'una base vectorial remota.

Risc si es fa malament. Si cada commit carrega un model pesat o reindexa un corpus, el CI esdevé lent, car i fràgil. Quan els tests són massa cars, l'equip deixa d'executar-los.

Microcas o incidència de laboratori: Lliçó final: el primer sistema defensable

Un equip no obté el model amb més rendiment de la classe, però entrega un projecte que es pot executar, auditar, monitoritzar i defensar. Declara dos riscos oberts, proposa mitigacions i mostra una demo curta que reprodueix el release. Aquest projecte és més valuós que un model aparentment millor que ningú pot reproduir.
Lliçó final del llibre: LIC no busca formar alumnat que només provin models, sinó que siguin capaços de construir sistemes intel · ligents amb evidència, límits i responsabilitat.

Tancament del D7

Artefacte lliurable

D7 — Integració final i defensa. Al final del capítol heu de lliurar:

- execució documentada de `python -m lic_project.final_report --config configs/d7_final.yaml`;
- matriu de traçabilitat `reports/final/d7_traceability_matrix.json`;
- registre de riscos `reports/final/d7_risk_register.json`;
- resum de release `reports/final/d7_release_summary.json`;
- informe `reports/d7_final_report.md`;
- figures D7 generades pel pipeline;
- validació amb `python -m lic_project.validate_release`;
- tests mínims executats amb `pytest`;
- guió de defensa i demo reproduïble.

Checklist de verificació

Abans de donar D7 per acabat, comproveu:

- el release s'executa amb comandes documentades;
- totes les figures es regeneren des del codi;
- la matriu de traçabilitat no amaga buits;
- els riscos crítics estan tancats o explícitament justificats;
- el validador de release passa o explica per què falla;
- la defensa té una demo curta i una narrativa tècnica coherent;
- l'informe final conté una decisió d'enginyeria, no només una conclusió optimista.

Criteris d'acceptació

Criteri	Punts
Release executable i sense dependència d'un notebook	2.0
Matriu de traçabilitat completa i honesta	1.5
Registre de riscos amb mitigació i decisió de gate	1.5
Informe final amb decisions d'enginyeria justificades	1.5
Figures i artefactes generats pel pipeline	1.0
Tests mínims i validador de release	1.0
Defensa tècnica clara: evidència, demo, limitacions i millores	1.5

Bibliografia i referències recomanades

Aquesta bibliografia està pensada com a suport del projecte transversal del Laboratori Integrat de Computació. Les referències no s'han seleccionat per acumular teoria, sinó per ajudar l'estudiant a construir, verificar i defensar els lliurables D0–D7: repositori, baseline, entrenament, auditoria, RAG, monitoratge, agents i release final.

Referències transversals per al projecte D0–D7

Enginyeria de sistemes d'IA i MLOps

- **Huyen, C. (2022).** *Designing Machine Learning Systems: An Iterative Process for Production-Ready Applications.* O'Reilly Media.
  Ús al LIC: referència central per entendre el pas de models a sistemes: dades, entrenament, avaluació, desplegament, monitoratge i iteració.
- **Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.-F., i Dennison, D. (2015).** *Hidden Technical Debt in Machine Learning Systems.* NeurIPS.
  Ús al LIC: lectura clau per justificar per què el projecte no pot acabar en un notebook: dependències ocultes, configuració, deute tècnic, validació i monitoratge.
- **Breck, E., Cai, S., Nielsen, E., Salib, M., i Sculley, D. (2017).** *The ML Test Score: A Rubric for ML Production Readiness and Technical Debt Reduction.* IEEE Big Data.
  Ús al LIC: base conceptual per convertir els lliurables en criteris de release: tests de dades, tests de model, contractes de pipeline i preparació per producció.
- **Kreuzberger, D., Kühl, N., i Hirschl, S. (2023).** *Machine Learning Operations (MLOps): Overview, Definition, and Architecture.* IEEE Access.
  Ús al LIC: visió de conjunt per connectar el repositori, el tracking, la integració, el desplegament i el monitoratge com un cicle únic.
- **Warden, P. i Situnayake, D. (2019).** *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers.* O'Reilly Media.
  Ús al LIC: lectura complementària per recordar que l'eficiència, la memòria i les restriccions de desplegament també formen part del disseny d'un sistema intel · ligent.

Documentació tècnica viva

- **Weights & Biases. (2026).** *Weights & Biases Documentation.* `https://docs.wandb.ai/`
  Ús al LIC: tracking d'experiments, registre d'hiperparàmetres, mètriques, artefactes, taules i comparació de runs.
- **PyTorch Contributors. (2026).** *PyTorch Documentation.* `https://docs.pytorch.org/`
  Ús al LIC: implementació de models, entrenament, checkpoints, tensors, optimitzadors i bucles d'entrenament reproduïbles.
- **scikit-learn Developers. (2026).** *scikit-learn User Guide.* `https://scikit-learn.org/stable/user_guide.html`
  Ús al LIC: baselines clàssics, particions de dades, pipelines, mètriques, calibració, validació

i models tabulars.

- **pytest-dev. (2026).** *pytest Documentation.* `https://docs.pytest.org/`
  Ús al LIC: tests mínims de dades, mètriques, contractes d'entrenament, contractes de RAG, seguretat d'agents i validació final.
- **Docker Inc. (2026).** *Docker Documentation.* `https://docs.docker.com/`
  Ús al LIC: empaquetament, entorns reproduïbles, Dockerfile, ports, imatges i execució fora del portàtil de l'estudiant.
- **FastAPI. (2026).** *FastAPI Documentation.* `https://fastapi.tiangolo.com/`
  Ús al LIC: servei d'inferència, contractes d'API, validació d'entrades amb tipus, endpoints i integració amb contenidors.
- **Hugging Face. (2026).** *Transformers Documentation.* `https://huggingface.co/docs/transformers/`
  Ús al LIC: models preentrenats, tokenització, inferència, linear probing, fine-tuning i ús responsable de models fundacionals.
- **Hugging Face. (2026).** *Datasets Documentation.* `https://huggingface.co/docs/datasets/`
  Ús al LIC: càrrega, versionat i preparació de datasets per a experiments reproduïbles.

Capítol 1: Fonaments de l'Aprenentatge Computacional: Del Símbol a la Dada

Lectures fonamentals

- **Hastie, T., Tibshirani, R., i Friedman, J. (2009).** *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Springer.
  Ús al LIC: base per entendre generalització, biaix-variança, regularització, classificació, mètriques i models clàssics. És una lectura d'aprofundiment, no un manual de laboratori.
- **James, G., Witten, D., Hastie, T., Tibshirani, R., i Taylor, J. (2023).** *An Introduction to Statistical Learning with Applications in Python.* Springer.
  Ús al LIC: alternativa més accessible i pràctica per reforçar baselines, validació, regressió, classificació, arbres i mètriques amb implementació propera al treball de grau.
- **Russell, S. i Norvig, P. (2020).** *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.
  Ús al LIC: marc general per situar el contrast entre sistemes simbòlics, cerca, inferència i aprenentatge estadístic.
- **Bishop, C. M. i Bishop, H. (2024).** *Deep Learning: Foundations and Concepts.* Springer.
  Ús al LIC: lectura moderna per connectar fonaments probabilístics, representacions i aprenentatge profund sense perdre el fil conceptual.

Articles i conceptes d'aprofundiment

- **Vapnik, V. N. (1999).** *An Overview of Statistical Learning Theory.* IEEE Transactions on Neural Networks.
  Ús al LIC: fonament teòric de generalització i risc estructural. Al cos principal només cal retenir la conseqüència operativa: sense validació independent no hi ha evidència de generalització.
- **Cover, T. M. (1965).** *Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition.* IEEE Transactions on Electronic

Computers.
Ús al LIC: suport conceptual per entendre per què canviar la representació pot fer un problema més separable.

- **Cybenko, G. (1989).** *Approximation by Superpositions of a Sigmoidal Function.* Mathematics of Control, Signals and Systems.
  Ús al LIC: fonament del poder aproximador de les xarxes neuronals. En grau interessa sobretot la lectura crítica: poder aproximar no vol dir entrenar bé ni generalitzar.
- **Domingos, P. (2012).** *A Few Useful Things to Know about Machine Learning.* Communications of the ACM.
  Ús al LIC: lectura curta i molt útil sobre dades, generalització, sobreajustament i errors habituals en projectes de ML.
- **Breiman, L. (2001).** *Random Forests.* Machine Learning.
  Ús al LIC: referència per entendre arbres, ensembles i baselines robustos abans d'introduir models profunds.

Capítol 2: Deep Learning i l'Arquitectura Transformer

Lectures fonamentals

- **Goodfellow, I., Bengio, Y., i Courville, A. (2016).** *Deep Learning.* MIT Press.
  Ús al LIC: referència de fons sobre optimització, regularització, CNN, representacions i entrenament profund.
- **Zhang, A., Lipton, Z. C., Li, M., i Smola, A. J. (2023).** *Dive into Deep Learning.* Cambridge University Press.
  Ús al LIC: molt adequada per a grau perquè combina explicació, codi i experiments executables.
- **Chollet, F. (2021).** *Deep Learning with Python* (2nd ed.). Manning.
  Ús al LIC: lectura pràctica per entendre regularització, bucles de treball i decisions d'arquitectura sense convertir el capítol en una demostració matemàtica.

Articles i conceptes d'aprofundiment

- **LeCun, Y., Bengio, Y., i Hinton, G. (2015).** *Deep Learning.* Nature.
  Ús al LIC: visió sintètica de per què l'aprenentatge profund aprèn representacions útils.
- **Vaswani, A., et al. (2017).** *Attention Is All You Need.* NeurIPS.
  Ús al LIC: referència fonamental de l'atenció i el Transformer. En D2 s'ha d'usar per entendre el mecanisme i el cost quadràtic, no per entrenar un LLM.
- **He, K., Zhang, X., Ren, S., i Sun, J. (2016).** *Deep Residual Learning for Image Recognition.* CVPR.
  Ús al LIC: lectura d'aprofundiment sobre arquitectures profundes i el paper de les connexions residuals.
- **Ronneberger, O., Fischer, P., i Brox, T. (2015).** *U-Net: Convolutional Networks for Biomedical Image Segmentation.* MICCAI.
  Ús al LIC: fonament de la pràctica satèl · lit de segmentació; cal llegir-la com a canvi de contracte de sortida i mètrica, no com a curs complet de visió mèdica.
- **Kingma, D. P. i Ba, J. (2015).** *Adam: A Method for Stochastic Optimization.* ICLR.
  Ús al LIC: base per entendre optimitzadors adaptatius i decisions de learning rate.
- **Loshchilov, I. i Hutter, F. (2019).** *Decoupled Weight Decay Regularization.* ICLR.
  Ús al LIC: justificació d'AdamW i de la regularització com a decisió observable en el

pipeline.

- **Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., i Salakhutdinov, R. (2014).** *Dropout: A Simple Way to Prevent Neural Networks from Overfitting.* Journal of Machine Learning Research.
  Ús al LIC: suport per comparar configuracions regularitzades amb runs registrats.
- **Hoffmann, J., et al. (2022).** *Training Compute-Optimal Large Language Models.* NeurIPS.
  Ús al LIC: lectura avançada sobre escala i pressupost. En D2 serveix per discutir recursos, no per exigir entrenaments massius.

Capítol 3: Verificació de Models: Robustesa, Explicabilitat i Equitat Algorísmica

Lectures fonamentals

- **Barocas, S., Hardt, M., i Narayanan, A. (2019).** *Fairness and Machine Learning: Limitations and Opportunities.* fairmlbook.org.
  Ús al LIC: base per entendre mètriques de grup, tensions entre criteris d'equitat i limitacions de les auditories automàtiques.
- **Molnar, C. (2022).** *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable* (2nd ed.). Leanpub.
  Ús al LIC: guia pràctica per explicar models, declarar limitacions i evitar confondre explicació amb causalitat.
- **Mitchell, M., et al. (2019).** *Model Cards for Model Reporting.* FAT*.
  Ús al LIC: referència per convertir l'auditoria D3 en evidència comunicable: rendiment, limitacions, context d'ús i riscos.
- **Gebru, T., et al. (2021).** *Datasheets for Datasets.* Communications of the ACM.
  Ús al LIC: suport per documentar dades, procedència, biaixos, cobertura i usos no previstos.

Articles i conceptes d'aprofundiment

- **Ribeiro, M. T., Singh, S., i Guestrin, C. (2016).** *"Why Should I Trust You?": Explaining the Predictions of Any Classifier.* KDD.
  Ús al LIC: base de les explicacions locals amb subrogats interpretables.
- **Lundberg, S. M. i Lee, S. I. (2017).** *A Unified Approach to Interpreting Model Predictions.* NeurIPS.
  Ús al LIC: referència SHAP per atribució de prediccions. En grau s'ha de tractar com a eina d'auditoria, no com a prova causal.
- **Hardt, M., Price, E., i Srebro, N. (2016).** *Equality of Opportunity in Supervised Learning.* NeurIPS.
  Ús al LIC: referència directa per a equal opportunity i equalized odds.
- **Dwork, C., Hardt, M., Pitassi, T., Reingold, O., i Zemel, R. (2012).** *Fairness through Awareness.* ITCS.
  Ús al LIC: lectura d'aprofundiment sobre la idea que tractar igual pot requerir entendre similituds rellevants entre individus.
- **Goodfellow, I., Shlens, J., i Szegedy, C. (2015).** *Explaining and Harnessing Adversarial Examples.* ICLR.
  Ús al LIC: base per introduir robustesa i proves de tensió sense convertir D3 en un capítol

d'atacs adversarials avançats.

- **Guo, C., Pleiss, G., Sun, Y., i Weinberger, K. Q. (2017).** *On Calibration of Modern Neural Networks.* ICML.
  Ús al LIC: lectura útil per entendre per què una probabilitat alta no equival necessàriament a confiança fiable.

Capítol 4: Sistemes amb Recuperació: RAG local, evidència i traçabilitat

Lectures fonamentals

- **Manning, C. D., Raghavan, P., i Schütze, H. (2008).** *Introduction to Information Retrieval.* Cambridge University Press.
  Ús al LIC: base per entendre recuperació, ranquing, consultes, índexs i mètriques de cerca.
- **Jurafsky, D. i Martin, J. H. (2025).** *Speech and Language Processing.* Draft 3rd edition.
  Ús al LIC: suport general per entendre embeddings, recuperació semàntica, llenguatge natural i generació.

Articles i conceptes d'aprofundiment

- **Lewis, P., et al. (2020).** *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.* NeurIPS.
  Ús al LIC: referència central per al patró RAG: recuperar evidència abans de generar.
- **Karpukhin, V., et al. (2020).** *Dense Passage Retrieval for Open-Domain Question Answering.* EMNLP.
  Ús al LIC: lectura per entendre recuperació densa i contrastar-la amb cerca lèxica.
- **Johnson, J., Douze, M., i Jégou, H. (2019).** *Billion-Scale Similarity Search with GPUs.* IEEE Transactions on Big Data.
  Ús al LIC: referència tècnica de FAISS i cerca vectorial eficient.
- **Malkov, Y. A. i Yashunin, D. A. (2020).** *Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs.* IEEE Transactions on Pattern Analysis and Machine Intelligence.
  Ús al LIC: fonament d'HNSW i índexs aproximats de veïns més propers.
- **Carbonell, J. i Goldstein, J. (1998).** *The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries.* SIGIR.
  Ús al LIC: base del criteri MMR per reduir redundància en el context recuperat.
- **Hu, E. J., et al. (2022).** *LoRA: Low-Rank Adaptation of Large Language Models.* ICLR.
  Ús al LIC: lectura avançada. Serveix per decidir quan adaptar un generador, però no substitueix un RAG que recupera bé.
- **Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., i Finn, C. (2023).** *Direct Preference Optimization: Your Language Model is Secretly a Reward Model.* NeurIPS.
  Ús al LIC: lectura avançada per entendre alineament per preferències un cop el sistema ja té evidència, mètriques i política d'abstenció.

Documentació tècnica específica

- **FAISS Contributors. (2026).** *FAISS Documentation.* `https://faiss.ai/`

Ús al LIC: índexs locals, cerca vectorial i comparació de recuperació.

- **Chroma. (2026).** *Chroma Documentation.* `https://docs.trychroma.com/`
  Ús al LIC: base de dades vectorial local per a prototips RAG, metadades i col · leccions.

Capítol 5: Monitoratge i Avaluació Contínua: Drift, regressió i qualitat en producció

Lectures fonamentals

- **Huyen, C. (2022).** *Designing Machine Learning Systems: An Iterative Process for Production-Ready Applications.* O'Reilly Media.
  Ús al LIC: referència transversal per al monitoratge, la iteració i la detecció de degradació després de la demo inicial.
- **Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., i Bouchachia, A. (2014).** *A Survey on Concept Drift Adaptation.* ACM Computing Surveys.
  Ús al LIC: base per entendre deriva de dades, canvis de distribució i adaptació de models.
- **Settles, B. (2009).** *Active Learning Literature Survey.* University of Wisconsin–Madison.
  Ús al LIC: fonament per convertir incertesa, novetat i desacord en una cua de revisió humana.

Articles i conceptes d'aprofundiment

- **Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., i Smola, A. (2012).** *A Kernel Two-Sample Test.* Journal of Machine Learning Research.
  Ús al LIC: referència de MMD per comparar distribucions, especialment en embeddings.
- **Kull, M., Silva Filho, T. M., i Flach, P. (2017).** *Beta Calibration: A Well-Founded and Easily Implemented Improvement on Logistic Calibration for Binary Classifiers.* AISTATS.
  Ús al LIC: lectura d'aprofundiment sobre calibratge i fiabilitat de probabilitats.
- **Zheng, L., Chiang, W.-L., Sheng, Y., et al. (2023).** *Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena.* NeurIPS.
  Ús al LIC: lectura clau per usar jutges automàtics amb cautela, calibratge i mostra humana de control.
- **Sculley, D., et al. (2015).** *Hidden Technical Debt in Machine Learning Systems.* NeurIPS.
  Ús al LIC: reforça que monitoratge, tests i gates són part del sistema, no decoració posterior.

Capítol 6: Sistemes amb Acció: agents, eines, guardrails i traçabilitat

Lectures fonamentals

- **Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., i Cao, Y. (2023).** *ReAct: Synergizing Reasoning and Acting in Language Models.* ICLR.
  Ús al LIC: referència central per entendre agents com a seqüència de raonament, acció, observació i verificació.
- **Schick, T., et al. (2023).** *Toolformer: Language Models Can Teach Themselves to Use Tools.* NeurIPS.
  Ús al LIC: lectura per entendre l'ús d'eines, però el LIC n'agafa una versió controlada: eines tancades, permisos, dry-run i logs.

- **Karpas, E., et al. (2022).** *MRKL Systems: A Modular, Neuro-Symbolic Architecture That Combines Large Language Models, External Knowledge Sources and Discrete Reasoning.* arXiv.
  Ús al LIC: suport per entendre sistemes modulars amb eines externes i separació de responsabilitats.

Seguretat, guardrails i governança d'accions

- **OWASP Foundation. (2025).** *OWASP Top 10 for Large Language Model Applications.* enllaç oficial.
  Ús al LIC: referència pràctica per discutir riscos de prompt injection, dades sensibles, permisos, supply chain i accions no autoritzades.
- **Bai, Y., et al. (2022).** *Constitutional AI: Harmlessness from AI Feedback.* arXiv.
  Ús al LIC: lectura avançada sobre restriccions normatives i polítiques de comportament en models generatius.
- **Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., i Yao, S. (2023).** *Reflexion: Language Agents with Verbal Reinforcement Learning.* NeurIPS.
  Ús al LIC: lectura d'aprofundiment sobre agents que revisen trajectòries. En D6 interessa sobretot el registre i l'avaluació de trajectòries, no l'autonomia sense límits.
- **Sutton, R. S. i Barto, A. G. (2018).** *Reinforcement Learning: An Introduction* (2nd ed.). MIT Press.
  Ús al LIC: lectura avançada per entendre MDP, política i recompensa. En el cos del capítol queda com a context, no com a nucli pràctic del lliurable D6.
- **Altman, E. (1999).** *Constrained Markov Decision Processes.* CRC Press.
  Ús al LIC: lectura avançada per connectar accions, restriccions i polítiques de seguretat quan es vulgui aprofundir més en control formal.

Capítol 7: Integració Final: release, defensa i portafoli d'evidències

Lectures fonamentals

- **Breck, E., Cai, S., Nielsen, E., Salib, M., i Sculley, D. (2017).** *The ML Test Score: A Rubric for ML Production Readiness and Technical Debt Reduction.* IEEE Big Data.
  Ús al LIC: referència directa per al release gate final: el projecte no es defensa només amb resultats, sinó amb proves, contractes i evidència.
- **Mitchell, M., et al. (2019).** *Model Cards for Model Reporting.* FAT*.
  Ús al LIC: base per presentar el model final amb ús previst, limitacions, mètriques, riscos i condicions de no ús.
- **Gebru, T., et al. (2021).** *Datasheets for Datasets.* Communications of the ACM.
  Ús al LIC: suport per al portafoli final de dades, cobertura, decisions de preparació i limitacions.
- **NIST. (2023).** *Artificial Intelligence Risk Management Framework (AI RMF 1.0).* National Institute of Standards and Technology.
  Ús al LIC: marc de governança per connectar evidència tècnica, riscos, controls i responsabilitat.
- **NIST. (2024).** *Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile.* National Institute of Standards and Technology.
  Ús al LIC: complement útil per sistemes generatius, RAG, agents, monitoratge i riscos

específics de generació.

Release, traçabilitat i defensa professional

- **Sculley, D., et al. (2015).** *Hidden Technical Debt in Machine Learning Systems.* NeurIPS.
  Ús al LIC: lectura final per justificar per què la defensa ha d'incloure deute tècnic, riscos pendents i decisions no resoltes.
- **Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., i Mané, D. (2016).** *Concrete Problems in AI Safety.* arXiv.
  Ús al LIC: lectura per situar reward hacking, errors d'especificació, supervisió, robustesa i accions insegures com a problemes d'enginyeria.
- **European Parliament and Council of the European Union. (2024).** *Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence.* Official Journal of the European Union.
  Ús al LIC: lectura de context normatiu per entendre que traçabilitat, gestió del risc, documentació i supervisió humana no són només bones pràctiques tècniques.
- **ISO/IEC. (2023).** *ISO/IEC 42001: Artificial intelligence management system.* International Organization for Standardization.
  Ús al LIC: referència de context per entendre sistemes de gestió d'IA, responsabilitats, controls i millora contínua.
- **GitHub. (2026).** *GitHub Actions Documentation.* `https://docs.github.com/actions`
  Ús al LIC: suport pràctic per automatitzar tests, validació de release i evidències executables en integració contínua.

Colofó

Composició: LaTeX
Edició: Grau en Enginyeria Informàtica, UAB
Versió: 1 de juny de 2026