

Entrenament d'un gran model de llenguatge per a la generació automatitzada d'aplicacions

Quer Masagué Deu

Grau d'Enginyeria Informàtica

Àrea d'Intel·ligència Artificial

Jonathan Ferrer Mestres

Xavier Baró Solé

20/6/2023

Quer Masagué Deu



Aquesta obra està subjecta a una llicència de Reconeixement-NoComercial-SenseObraDerivada [3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FITXA DEL TREBALL FINAL

Títol del treball:	<i>Entrenament d'un gran model de llenguatge per a la generació automatitzada d'aplicacions.</i>
Nom de l'autor:	<i>Quer Masagué Deu</i>
Nom del consultor/a:	<i>Jonathan Ferrer Mestres</i>
Nom del PRA:	<i>Xavier Baró Solé</i>
Data de lliurament:	06/2023
Titulació:	<i>Grau d'Enginyeria Informàtica</i>
Àrea del Treball Final:	<i>Intel·ligència artificial</i>
Idioma del treball:	Català
Nombre de crèdits:	12
Paraules clau	<i>Machine Learning (ML), Natural Language Processing (NLP), Large Language Model (LLM)</i>

Resum del Treball (màxim 250 paraules): *Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball.*

Es vol definir una estratègia que aprofiti la potència dels grans models de llenguatge per a la generació automatitzada d'aplicacions. Amb aquesta finalitat es dissenya i entrena un model de llenguatge generatiu a partir de fonts pròpies.

La utilització d'aquests models en el camp del desenvolupament d'aplicacions es troba limitat al context de l'assistència, tot i que l'evolució de la seva qualitat comença a fer-los adients per automatitzar tasques d'aquest tipus. Malgrat que l'accés als grans models preentrenats s'ha privatitzat existeix una gran comunitat que treballa en versions obertes.

Es proposa fer servir una d'aquestes arquitectures obertes, la nanoGPT, per entrenar un model amb aquesta finalitat. Degut a l'elevat cost de computació i els grans volums de dades requerits, s'ha hagut de multiplicar els datasets originals mitjançant plantilles. En un procés iteratiu s'han entrenat i comparat diferents configuracions dels models cercant la millora de la qualitat dels seus resultats. Gràcies a això i a l'aplicació de tècniques d'enginyeria d'indicacions s'ha arribat a assolir l'objectiu de generar petites aplicacions de forma automatitzada amb les funcionalitats i paràmetres requerits.

Aplicant aquestes resultats es podria entrenar un model a partir de codi productiu per a oferir suport a una aplicació que faciliti la creació automatitzada

d'aplicacions. El cost de la infraestructura necessària porta a valorar la utilització d'un model preentrenat refinat amb el codi propi resultant a més una eina que pot ser consultada directament amb llenguatge natural per a l'obtenció del programa necessari sense necessitat d'aplicació mitjancera.

Abstract (in English, 250 words or less):

This work aims to define a strategy to take advantage of the large language models for automated application generation. To this end, a generative language model is designed and trained using proprietary sources.

While the use of these models in application development is currently limited to the context of assistance, their improving quality is making them increasingly suitable for automating tasks of this type. Although the privatization of access to pre-trained large models there is a large community working on open versions.

It is proposed to use one of these open architectures, the NanoGPT, to train a model for this purpose. Due to the high computational cost and the large volumes of data required, the original datasets had to be multiplied using templates. In an iterative process, different configurations of the models have been trained and compared, seeking to improve the quality of their results. Through this approach and the application of prompt engineering techniques, the goal of generating small applications in an automated way with the required functionalities and parameters has been achieved.

Applying these results, it becomes possible to train a model based on production code to provide support for an application that facilitates the automated creation of applications. Considering the cost of the necessary infrastructure, using a pre-trained model refined with custom code becomes an attractive option. This tool can be directly queried using natural language to obtain the required program without the need for an intermediary application.

Índex

1. Resum	1
2. Introducció	2
2.1. Context i justificació del Treball	2
2.2. Objectius del Treball	3
2.3. Enfocament i mètode seguit	4
2.4. Planificació del Treball	5
2.5. Breu sumari de contribucions i productes obtinguts	6
2.6. Breu descripció dels altres capítols de la memòria	7
3. Estat de l'art	7
3.1. Descripció general	7
3.2. L'arquitectura del models	9
3.3. Detall funcionament algorisme nanoGPT	13
4. Metodologia	15
5. Resultats	23
6. Discussió	29
7. Valoració econòmica	29
8. Conclusions	31
8.1. Conclusions	31
8.2. Línies de futur	32
8.3. Seguiment de la planificació	33
9. Glossari	34
10. Bibliografia	35
11. Annexos	38

Llista de figures

Figura 1 Diagrama de Gannt de la planificació del treball	6
Figura 2 Estructura d'una neurona (biològica)	8
Figura 3 Neurona d'una xarxa neural	8
Figura 4 Esquema de xarxa neural artificial	8
Figura 5 Evolució Grans Models de Llenguatge	9
Figura 6 Transformer architecture	10
Figura 7 GPT2 architecture	12
Figura 8 NanoGPT architecture	13
Figura 9. Tauler wandb amb comparativa d'entrenaments	21
Figura 10 Corba de pèrdua	22
Figura 11 Diagrama de Gantt del detall d'iteracions d'entrenaments	33

Llista de taules

Taula 1 Planificació del treball	5
Taula 2. Recursos de màquina utilitzats	20
Taula 3 Resultats dels entrenaments	24
Taula 4 Cost energètic dels entrenaments	29

1. Resum

Amb aquest treball final de grau es preté aprofundir en l'àmbit de la intel·ligència artificial orientat al camp del processament del llenguatge natural i analitzar els seus possibles usos a l'àmbit de la generació automàtica de codi aplicat a l'àmbit laboral per donar resposta a la necessitat de crear d'aplicacions de forma més eficient.

Es centra en els mètodes i tècniques utilitzats en aquest camp com són els algorismes d'aprenentatge computacional no supervisat per a crear grans models de llenguatge, així com les seves arquitectures. Concretament estudia els tipus *Generative Pre-Trained models* (GPT) que són els que han revolucionat aquest camp en els últims anys. Per ampliar el coneixement sobre aquests models es treballen els conceptes de xarxa neuronal recurrent, *Transformer*, entrenament, parametrització i refinament, entre d'altres.

L'estudi emprà el model d'entrenament nanoGPT d'Andrej Karpathy (2023b) que reproduïx l'arquitectura i algorismes utilitzats per l'entrenament del GPT2 al 2017. S'utilitza un conjunt de dades privat generat i diversificat automàticament de forma aleatòria. Un cop entrenats els models se n'analitza l'efectivitat obtenint indicis de millora. També es recorreix a l'enginyeria de les indicacions (*prompt engineering*) per tal de millorar la qualitat dels resultats. A més es treballa de forma iterativa entrenant diferents models canviant aspectes clau del conjunt de dades d'entrenament, parametritzacions, tècniques d'entrenament i comparant-ne els resultats. L'objectiu d'aquest procés és el d'aconseguir una aplicació a l'àmbit empresarial de models privats especialitzats en tasques concretes, com la d'ajudar en la producció de codi similar a l'existent sota demanda. A més s'ha valorat la viabilitat de l'entrenament d'un model propi que eviti dependre dels models privats oferint una opció confidencial lliure de fuites d'informació sensible.

Els resultats obtinguts han estat significatius tot i la limitació en la quantitat de dades emprades als entrenaments així com la limitació en els recursos de computació gràcies a la utilització de tècniques de reducció del sobreajustament així com amb enginyeria d'indicacions.

S'ha aconseguit entrenar un model capaç de generar codi nou a partir de descripcions on s'especifiquen les funcionalitats i parametritzacions desitjades. Tot i que aquesta funcionalitat és similar a la que podria oferir un sistema de plantilles, l'utilització d'un model així, entrenat a partir dels repositoris privats podria evitar la feina d'aplanillar cada tipus d'aplicació existent. Tanmateix es proposen alternatives utilitzant models avançats amb les que es podrien crear les plantilles des del propi model.

2. Introducció

2.1. Context i justificació del Treball

Dins el camp de la intel·ligència artificial, els últims mesos s'està parlant molt del ChatGPT i de les seves potencialitats. Aquest xatbot utilitza una família de models de llenguatges entrenats amb la tècnica d'aprenentatge de màquina per reforç. Aquests models permeten entendre el llenguatge natural i processar respostes coherents en base al corpus utilitzat durant l'entrenament. L'actual *hype* en l'ús d'aquestes tècniques ve donat per la gran quantitat de recursos utilitzats durant l'entrenament que ha permès l'èxit a l'hora de respondre correctament les qüestions dels usuaris en diferents temàtiques. Tot i això en el camp de l'automatització de creació d'aplicacions informàtiques no ha tingut encara prou recorregut.

Actualment encara té molt pes l'acció manual dels desenvolupadors. Tot i que van apareixent opcions que no requereixen desenvolupadors experimentats (com les “*no code*” o “*low code*”) que simplifiquen la creació d'aplicacions en determinats àmbits, solen requerir un treball previ sobre “plantilles” i els seus resultats solen ser simples o homogenis i no cobreixen totes les necessitats que poden tenir les empreses. Són conegudes opcions que permeten l'aplantillament o arquetipus dels diferents programes de tal manera que quan se'n necessita un de nou. Aquestes eines de substitució o generació de codi automàtic (com Maven), però, requereixen un sobre esforç que les empreses no sempre estan disposades a acceptar. Tanmateix els assistents dels entorns de treball dels desenvolupadors han millorat molt les seves tasques d'automatització aplicant tècniques d'assistència per tal d'augmentar la seva productivitat. A més, recentment s'hi han incorporat tècniques d'escriptura automàtica que proposen la “següent” paraula o paràgraf com ho fa el Github Copilot. Tot i que han suposat un gran avanç es mantenen en la línia de l'assistència supervisada i encara no han demostrat ser de gran ajuda en la creació de noves classes. A més tenen un cost elevat per usuari.

Tanmateix amb l'aparició del ChatGPT s'ha generat molt interès per trobar noves maneres d'utilitzar-lo per a millorar la productivitat a diferents tasques però també s'han fet palesos problemes de privacitat i confidencialitat, de propietat intel·lectual, de biaixos que s'han d'afrontar (Gartner 2023). Per últim existeix incertesa en la regulació estatal (Pacho 2023) i supra-estatal (Noticias del Parlamento Europeo 2023) que pot acabar afectant a la disponibilitat d'aquestes eines.

Quer Masagué Deu

Dins l'àmbit laboral personal es proposa una col·laboració per a investigar com es podrien emprar els grans models de llenguatge (en endavant LLM de l'anglès) per a generar codi de forma automàtica per a la automatització de tasques feixugues i repetitives com són la creació d'aplicacions senzilles de les que no es disposen plantilles. L'opció que es planteja és estudiar la viabilitat d'entrenar models privats a partir de repositoris interns així com el refinament de models públics mitjançant codi propi per generar un model privatiu. Aquesta col·laboració amb l'empresa que inicialment havia de proporcionar un entorn de computació al núvol ha quedat reduïda a la cessió d'un conjunt limitat d'aplicacions del repositori. Per qüestions de privacitat i confidencialitat de dades els entrenaments realitzats amb aquestes fonts no es comparteixen al treball.

Per a simplificar la tasca es limita el conjunt de dades d'entrenament a un sol tipus d'aplicacions que utilitzen tecnologia Camel Blueprint escrits en llenguatge XML (veure [Annex 1](#)).

Cal remarcar que un model amb aquesta finalitat es troba limitat a la generació de funcionalitats conegudes, és a dir que apareixen al conjunt d'entrenament i es poden diferenciar bé les unes de les altres. Això és degut a la complexitat mateixa del programari que sovint cada implementació concreta que fa servir un número de paràmetres concret, amb una tipologia específica a cada cas (com una cadena de text o un tipus numèric). Tot i això la consecució de l'objectiu final a partir d'un mínim d'entrenament pot resultar molt útil en el camp de la creació automàtica d'aplicacions.

Amb la combinació d'aquests models propis amb models de reconeixement de llenguatge natural (oberts o comercials) s'obre la possibilitat d'accedir als models en mode xat bot.

2.2. Objectius del Treball

- Disseny i entrenament d'un model LLM, concretament nanoGPT per a la generació de fitxers XML a partir de consultes.
- Definir una estratègia utilitzant els LLM per a la generació automatitzada d'aplicacions.

Objectius específics:

- 1) Contextualització dels conceptes de LLM, GPT i Transformers.

- 2) Aprofundiment en les tècniques que s'utilitzaran en el treball: nanoGPT, PyTorch, tècniques de consulta (*prompt engineering*).
- 3) Definició de l'arquitectura del model i dels algorismes que s'utilitzen durant l'entrenament
- 4) Preparació de l'entorn d'entrenament.
- 5) Preparació de les dades d'entrenament del model:
 - a) Elaboració d'un script per extreure dels repositoris els fitxers amb els quals es vol alimentar el model. Com que resultaran ser insuficients, s'elaborarà un altre script que creï aleatòriament nous fitxers amb el format necessari.
 - b) Revisió dels fitxers i eliminació de dades confidencials si s'escau
- 6) Entrenament iteratiu del model
 - a) Definició del conjunt d'entrenament
 - i) Preparació de les dades, si s'escau
 - ii) Generació de dades d'entrenament
 - iii) Separació dels conjunts d'entrenament i test
 - b) Entrenament del model
 - c) Refinament del model, si s'escau
 - d) Avaluació del model
 - e) Desplegament del model
- 7) Valoració de l'efectivitat dels resultats obtinguts per assolir l'objectiu

2.3. Enfocament i mètode seguit

L'estratègia plantejada inicialment com a solució és la d'utilitzar un model generatiu preentrenat (per exemple el GPT de OpenAI) i refinar-lo amb els conjunt de programes propis (sense aplantillar). Aquesta opció genera dubtes: tant de privacitat de les dades alhora d'enviar-les als servidors dels propietaris del model, com de viabilitat en quant a costos, com de possibles bloquejos legals en quant a regulació pública.

Com a alternativa es proposa l'entrenament d'un model propi a partir del repositori de programes intern. S'ha fet recerca sobre diferents mètodes d'entrenament que es podrien emprar com GPT, H Neural Architecture Search - NAS, optant pel primer del que es disposa una implementació en obert i molt accessible, anomenada NanoGPT. S'ha investigat com implementar-ne la càrrega de dades i com optimitzar l'entrenament per a la millora dels resultats obtinguts mitjançant el correcte etiquetatge dels conjunts de dades.

S'han adoptat diferents estratègies per a fer front als recursos limitats que disposàvem, com la quantitat de dades del conjunt d'entrenament, o els recursos de hardware. Per exemple, el conjunt d'entrenament inicialment

Quer Masagué Deu

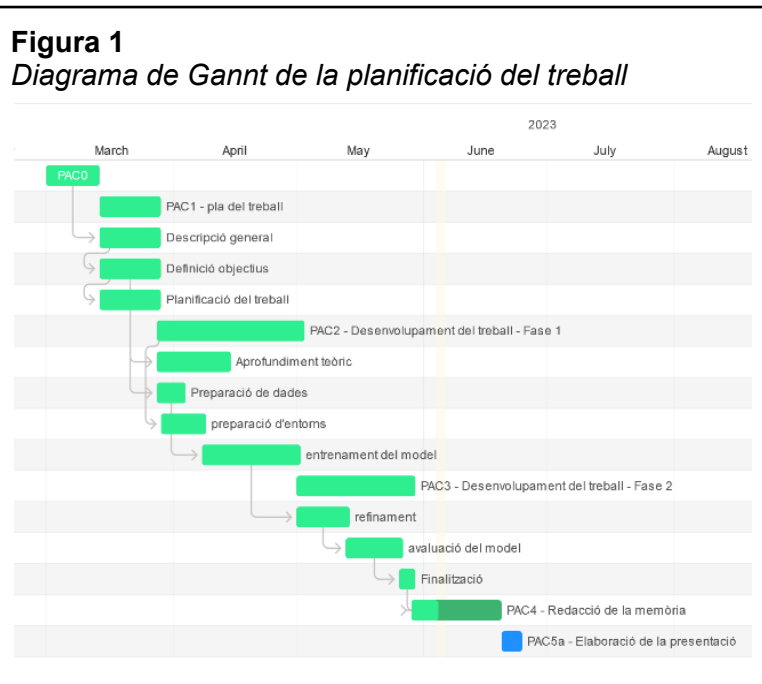
només estava format per 30 programes Camel Blueprint en XML reals facilitats per l'empresa col·laboradora, i per a entrenar un LLM se'n necessiten milers. Per solventar-ho s'ha hagut de crear un programa en Python per tal de produir còpies a partir de plantilles que s'han dissenyat a partir dels programes més significatius. Finalment, s'ha creat també un conjunt de programes encara més simplificats on es fan desaparèixer parts no fonamentals del programa per millorar l'eficiència dels entrenaments.

2.4. Planificació del Treball

Taula 1

Planificació del treball

PAC0	3/1/2023	3/13/2023
PAC1 - pla del treball	3/14/2023	3/28/2023
Descripció general	3/14/2023	3/28/2023
Definició objectius	3/14/2023	3/28/2023
Planificació del treball	3/14/2023	3/28/2023
PAC2 - Desenvolupament del treball - Fase 1	3/29/2023	5/3/2023
Aprofundiment teòric	3/29/2023	15/4/2023
Preparació de dades	3/29/2023	4/4/2023
preparació d'entorns	3/30/2023	4/9/2023
entrenament del model	4/9/2023	5/2/2023
PAC3 - Desenvolupament del treball - Fase 2	5/2/2023	5/30/2023
refinament	5/2/2023	5/14/2023
avaluació del model	5/14/2023	5/27/2023
Finalització	5/27/2023	5/30/2023
PAC4 - Redacció de la memòria	5/30/2023	6/20/2023
PAC5a - Elaboració de la presentació	6/21/2023	6/25/2023



2.5. Breu sumari de contribucions i productes obtinguts

S'han obtingut diferents models. Aquesta és la relació dels que es poden considerar vàlids:

- El model 10-BIS aconsegueix realitzar la substitució de paràmetres dels programes aplantillats.
- El model 13 aconsegueix realitzar la substitució de paràmetres dels programes aplantillats.
- El model 14 aconsegueix la generació de programes nous a partir de la menció de les funcionalitats requerides realitzant la substitució de paràmetres de cadascuna d'elles.

No es publiquen degut al seu gran pes, al voltant d'1GB. Per això es publica al github tot el programari i configuracions per a reproduir-los.

<https://github.com/quermd/TFC>

- Programes python per a generar XMLs a partir de plantilles ([xmlAutomationV6](#) en diferents versions).
- Jocs de plantilles utilitzades per a generar els diferents datasets: [short_template.xml](#) , [short_template_pro.xml](#),
- Configuracions del NanoGPT per a entrenar el model ([config](#))
- Programes utilitzats per interactuar amb el model ([from_prompttxt_sample.py](#), [q_sample_Interactive.py](#))

Quer Masagué Deu

- Textos utilitzats per a consultar al model ([test10.txt](#), [prompt3.txt](#))
- Utilitats addicionals ([print_gpu_utilization.py](#), [joinXMLs.py](#))

2.6. Breu descripció dels altres capítols de la memòria

El capítol 3 conté el rerefons de les tecnologies utilitzades. El capítol 4 explica la metodologia dels entrenaments i el 5 els seus resultats. Al 7 es proposa la valoració econòmica del projecte i les conclusions es troben al capítol 8.

L'annex 1 conté un exemple d'aplicació, l'annex 2 el detall de configuració d'un entrenament i el 3 conté la taula comparativa dels entrenaments realitzats.

Als annexos 4, 5 s'entrega el codi de les aplicacions 11, 12 , 13 i 14.

L'annex 6 conté el resultat de l'entrenaments 10 i l'annex 7 el de les indicacions contra el ChatGPT.

3. Estat de l'art

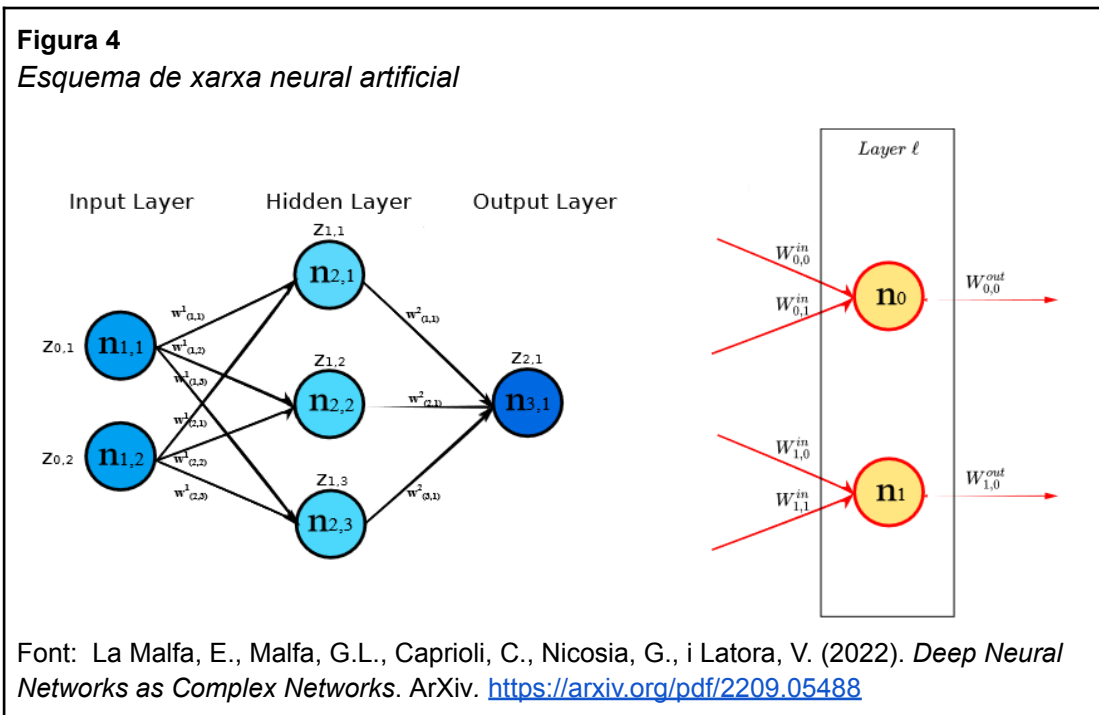
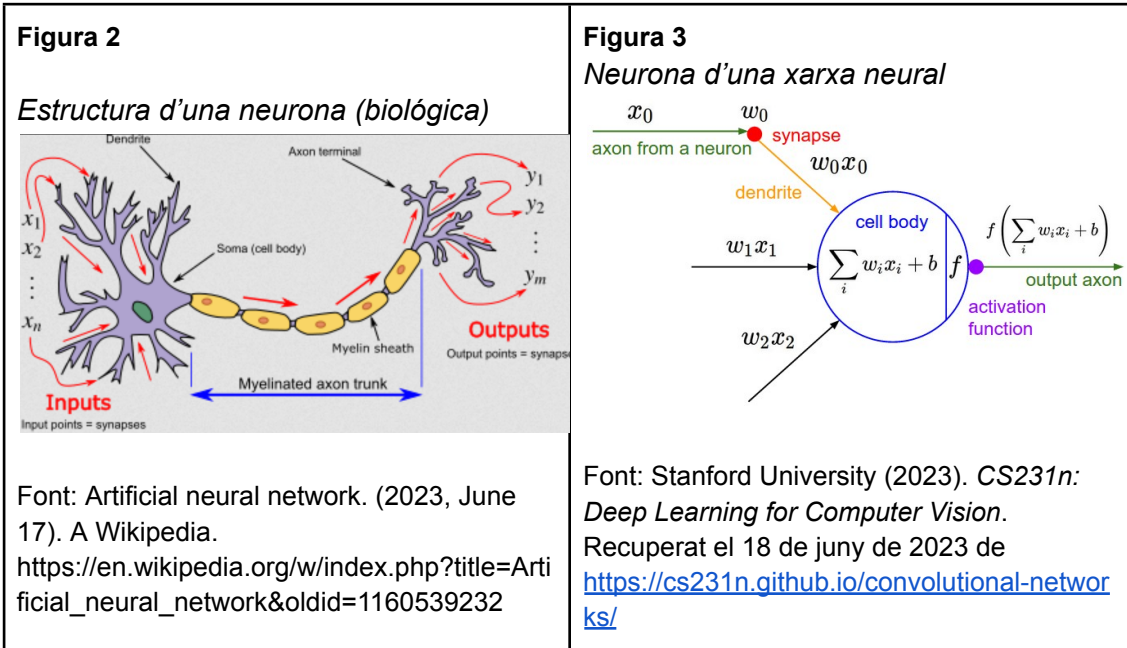
3.1. Descripció general

Actualment l'automatització del codi es centra en l'ús d'assistents als entorns de desenvolupament integrats, així com eines de prototipatge o bastiment (*scaffolding*) a partir d'apantillaments previs (o arquetipus), i per últim d'assistents IA generatius.

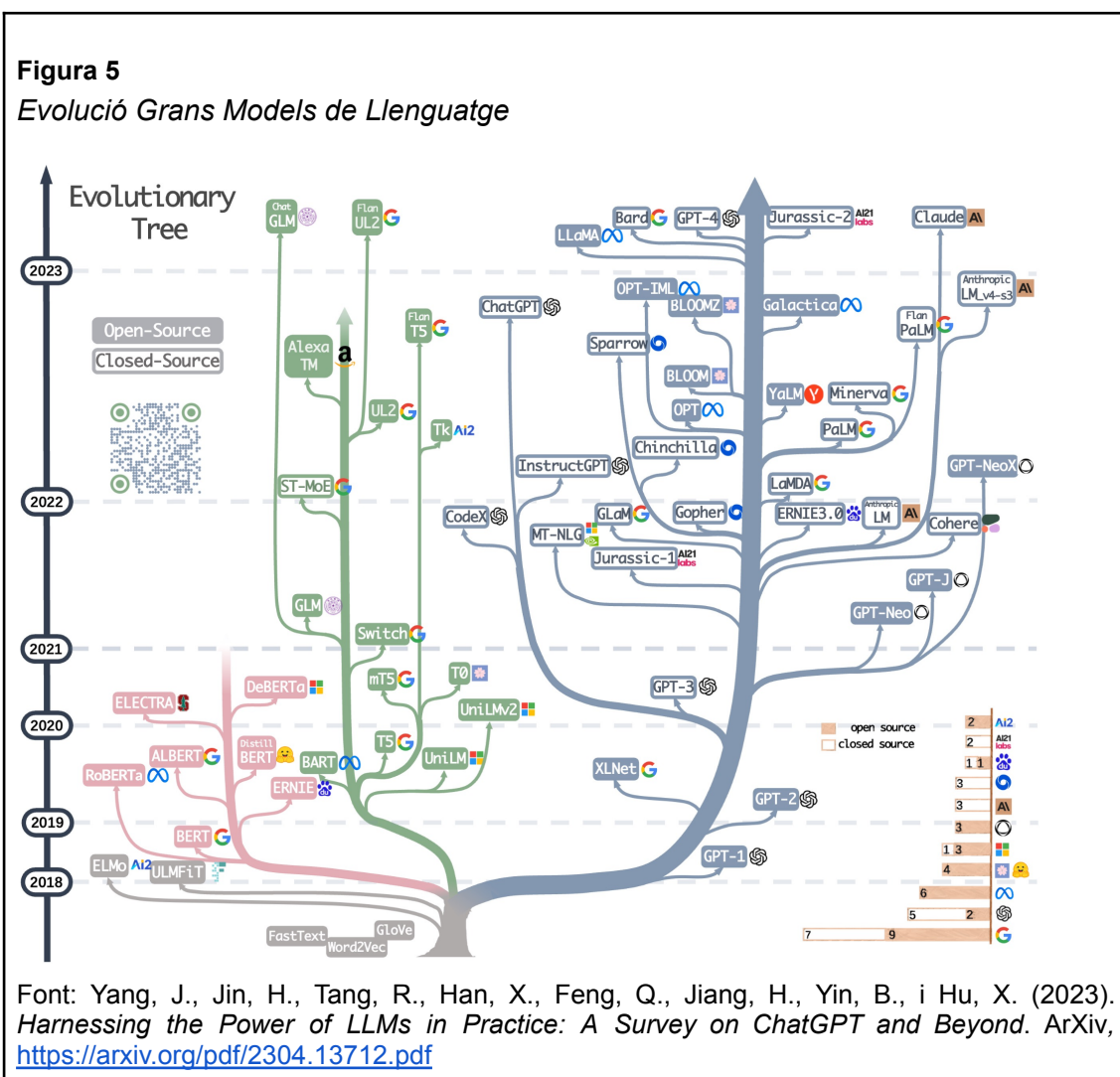
El present treball es centra en aquests últims: la Intel·ligència Artificial Generativa, que és un subcamp de la intel·ligència artificial centrat en la creació d'algorismes i xarxes neuronals artificials capaços de generar contingut original i creatiu (Wikipedia 2023 a). Utilitzant models probabilístics per generar dades que s'assemblen a les dades d'entrenament, però que no són meres còpies. La IA generativa ha estat utilitzada amb èxit en àmbits com la generació d'imatges, la música, el text i fins i tot el disseny de nous fàrmacs..

Les xarxes neuronals artificials (XNA o ANN en anglés), tal i com indica Torra i Reventós (2019), són estructures emprades en els algorismes d'aprenentatge automàtic inspirats en el funcionament del cervell humà (figura 2). Aquestes xarxes s'implementen mitjançant múltiples capes de neurones

artificials (figura 3) interconnectades (figura 4). Cadascuna d'aquestes processa i transmet la informació a través de pesos i funcions d'activació. Les xarxes neuronals són capaces d'aprendre patrons i relacions complexes a partir d'un conjunt de dades mitjançant un procés iteratiu anomenat entrenament, durant el qual s'ajusten els pesos de les "neurons" per minimitzar l'error en la predicció de les dades (Kalchbrenner et al 2016).



Els LLMs són un tipus especialitzat de xarxes neuronals que s'entrenen per comprendre i generar llenguatge humà (*natural language processing*, NLP) utilitzant una gran quantitat de dades textuais com a conjunt d'entrenament per tal d'aprendre les estructures i les regles del llenguatge. Amb aquest coneixement adquirit poden: generar text coherent, respondre preguntes, traduir idiomes i fins i tot, simular el to i l'estil d'un autor en particular. El desenvolupament dels LLMs té una ràpida evolució però també una tendència a la privatització dels models resultants degut als alts costos en els seus entrenaments tal i com s'observa a la figura 5 on es mostra una cronologia dels LLMs que actualment lideren els rankings de qualitat així com la seva naturalesa oberta o privada.

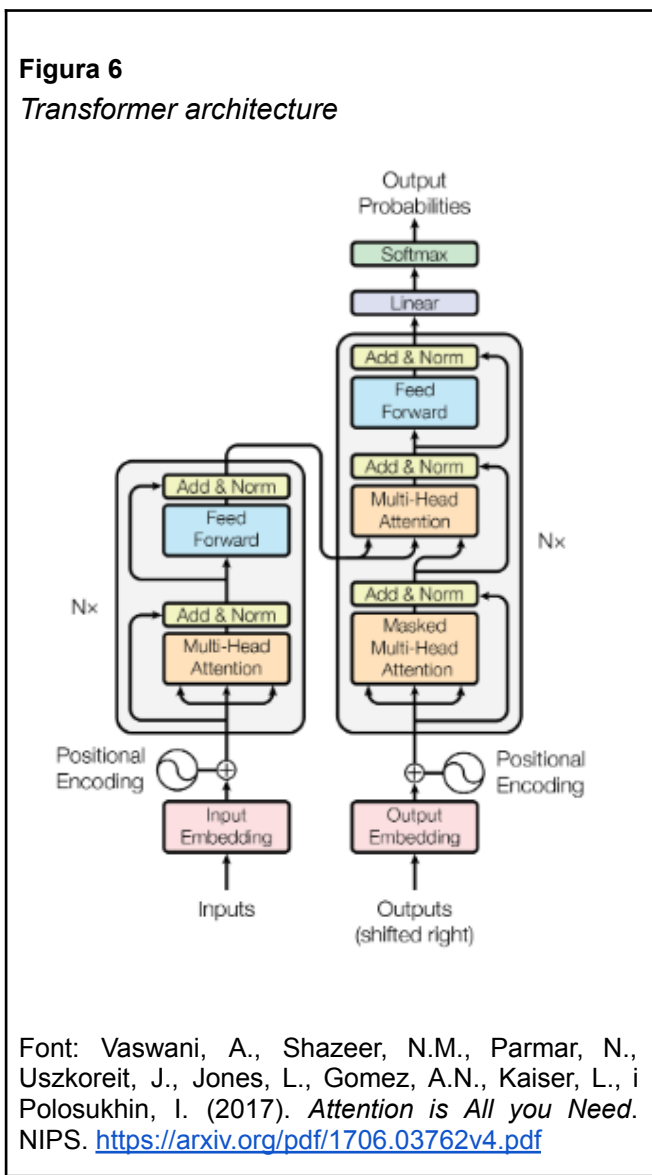


3.2. L'arquitectura del models

Els LLMs més recents es basen en mètodes d'atenció, implementas per Bahdanau et al. (2015), que permeten als models de llenguatge assignar

pesos als diferents trets o paraules d'una seqüència en funció de la seva importància relativa dins el context. Gràcies a aquest mecanisme es poden captar relacions complexes i de llarg abast entre les paraules i ajudar a millorar la comprensió i generació de llenguatge. L'obra de Vaswani et al. (2017), amb la presentació dels Transformadors, van optimitzar i popularitzar aquest tipus de models. Aquests últims implementen una arquitectura caracteritzada per utilitzar múltiples capes d'atenció (attention) i, a diferència de les arquitectures recurrents tradicionals, permetent un processament en paral·lel i més eficient de les dades, cosa que els fa adequats per a models de llenguatge amb una gran quantitat d'informació.

Figura 6
Transformer architecture



Font: Vaswani, A., Shazeer, N.M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., i Polosukhin, I. (2017). *Attention is All you Need*. NIPS. <https://arxiv.org/pdf/1706.03762v4.pdf>

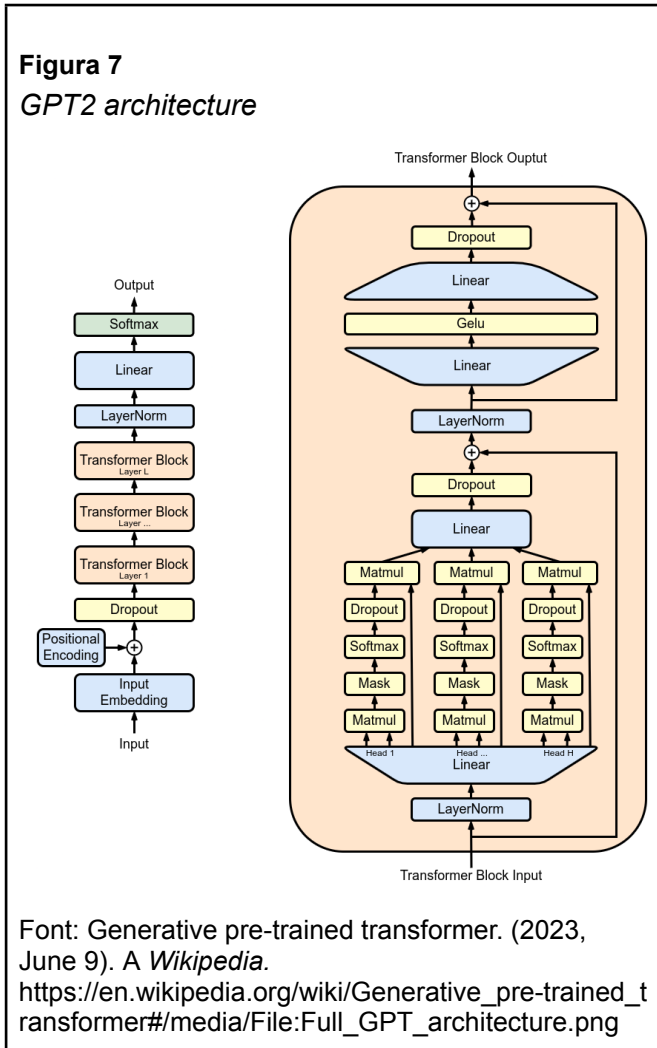
Els transformadors consten de dues parts diferenciades: el Codificador (*Encoder*) i el Decodificador (*Decoder*). El primer (a l'esquerra a la figura 6) s'encarrega de processar l'entrada aplicant varies passades del mecanisme

d'auto-atenció per a capturar la relació semàntica i contextual de les paraules creant una sortida vectorial. El Decodificador, en canvi, processa el contingut de l'entrada i el de la sortida del codificador de forma iterativa i incremental per a cada posició per obtenir la predicció del següent símbol de la seqüència.

Per a preparar l'entrada dels transformadors s'utilitza el mètode de tokenització fent ús de la incrustació de mots (word embedding) per al mapatge dels *tokens* (divisió dels mots en sub parts). Aquest mapatge es combina amb el "*position embedding*" que assigna un valor a cada *token* en funció de la seva posició dins la sentència obtenint una representació en un espai vectorial continu on es capturen i preserven les relacions semàntiques entre els *tokens* permetent a més una de les característiques dels transformadors: el processament en paral·lel de tota la sentència d'entrada. Un cop processat pels blocs dels transformadors s'aplica una capa per tornar a representar els *tokens* com a paraules (anomenada "*Linear*" que bàsicament aplica la multiplicació de matrius) i per últim gràcies a la capa "*softmax*" es transformen els vectors en una distribució probabilística.

L'arquitectura del model de GPT2 (figura 7) varia lleugerament respecte als transformadors originals al ser una arquitectura que només utilitza decodificador (Radford et al 2019). A part dels blocs de transformadors s'hi troben altres components com són els mecanismes de normalització de capes, de *dropout* (ideat per Srivastava et al 2014) que permeten uns resultats menys sobreajustats (*overfitting*) a base d'ignorar aleatòriament un percentatge dels pesos nous de cada passada.

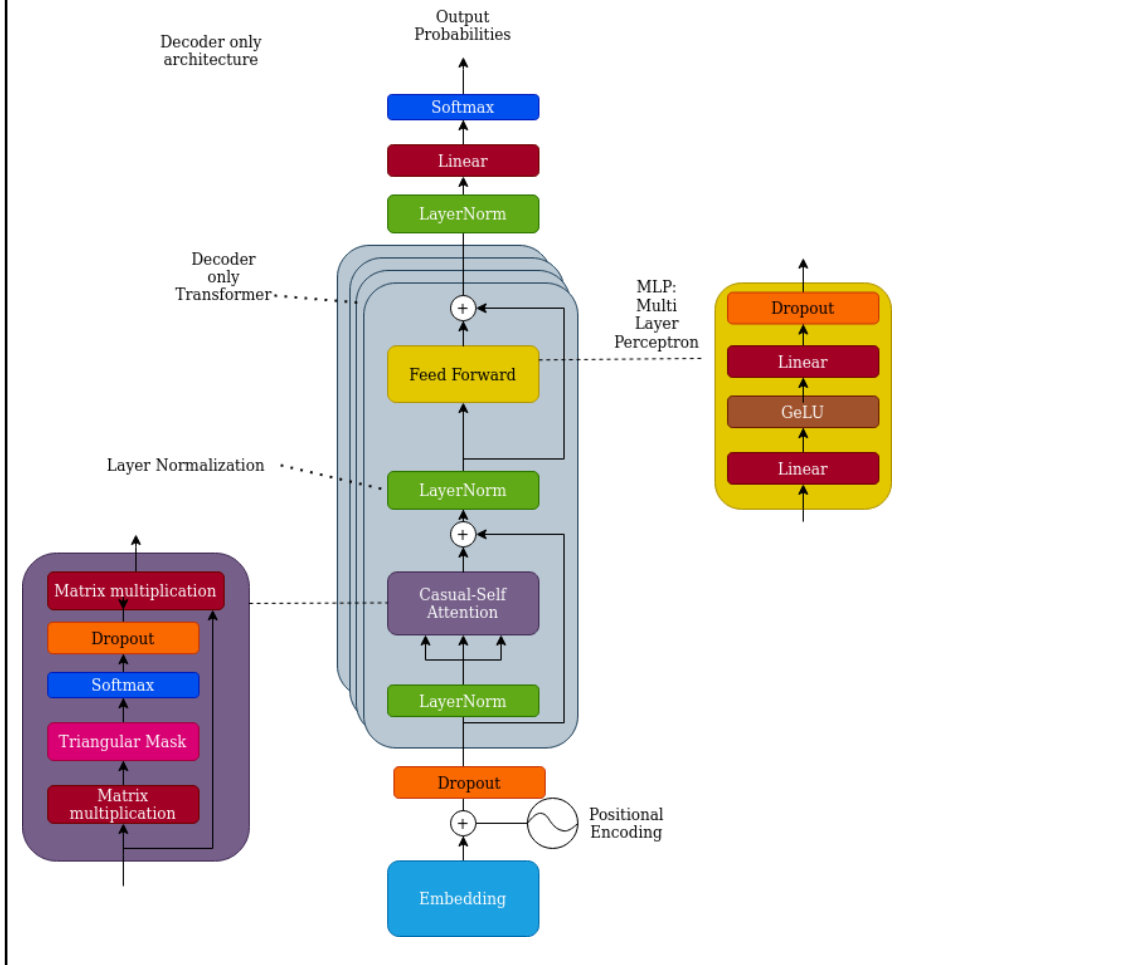
Figura 7
GPT2 architecture



Font: Generative pre-trained transformer. (2023, June 9). A *Wikipedia*.
https://en.wikipedia.org/wiki/Generative_pre-trained_transformer#/media/File:Full_GPT_architecture.png

Per últim es descriu l'arquitectura del model NanoGPT (figura 8) que és pràcticament idèntica a la del GPT2 en la versió on es fa ús del mecanisme "casual self-attention" (Karpathy 2023) al aplicar un emmascarament de la matriu (multiplicant-la per una matriu triangular on la meitat està marcada com a infinits) per tal d'impedir que el transformador pugui aprendre de la informació posicional dels *tokens* que hi ha per davant de la sentència del *token* "analitzat" (evitant l'aprenentatge del context futur). Aquesta particularitat és opcional als transformadors existents variants que permeten el seu ús, per exemple en casos on el que es vol obtenir és "el sentiment" de la frase (i no una predicció de la següent paraula).

Figura 8
NanoGPT architecture



3.3. Detall funcionament algorisme nanoGPT

L'entrada de l'algorisme és el conjunt de dades (al que també ens referim per el seu terme en anglès "*dataset*") agrupat en un únic fitxer que es prepara amb una funció complementària del repositori de Karpathy (2023b). En aquesta el text es transforma en *tokens* mitjançant la llibreria TikToken de OpenAI, i es divideix en dues parts, una d'entrenament (desada al fitxer "train" amb un 90% de *tokens* respecte el total) i l'altre de validació (desada al fitxer "val" amb un 10% de *tokens*).

Durant l'entrenament, a cada iteració es recullen diferents conjunts (blocs) de *tokens* de forma aleatòria del llistat de *tokens* que es troba a "train". El nombre de conjunts es regula amb l'hyper-paràmetre "*batch_size*". Aquests conjunts es carreguen a la GPU i es passen pel *transformer* de forma simultània. Un cop allí es processen per totes les capes indicades a l'arquitectura del model. El model és una matriu multidimensional amb els pesos de les probabilitats del

“següent” *token* segons els *tokens* del seu context (els que hi estan “a prop”). L’objectiu de cada iteració és la de sumar a aquesta matriu el resultat de cada conjunt nou de *tokens* analitzat. Això s’aconsegueix gràcies al mecanisme de retropropagació (*backpropagation*) que, donat un conjunt de *tokens* i el resultat esperat (*token* següent), calcula el valor resultant de travessar la xarxa neural (feed forward layer) “cap enrere” (des del resultat fins a l’entrada) mitjançant la derivació de les fórmules matemàtiques conformant els valors a afegir a la matriu de pesos. Per a perfeccionar aquest procés s’utilitzen diferents tècniques com el *positional encoding* (Kosar 2022 i Wang et al. 2019), el mecanisme de *cross attention* (Kosar 2021), la normalització dels valors i s’apliquen mecanismes d’optimització dels resultats com *dropout*.

Un dels paràmetres amb més rellevància és la llargada del context: la mida dels blocs de *tokens* (*block_size*) que es fan servir en cada iteració de l’entrenament. Per últim, el nombre d’iteracions (*max_iters*) determina, juntament amb el *batch size* i el *block size*, el total de dades que es tractaran del conjunt d’entrenament¹. Cal notar que a diferència de les Xarxes Neurals Recurrents aquest model no realitza diferents passades sobre el mateix conjunt de dades (epochs), si no que realitza moltes passades sobre parts aleatòries d’un conjunt molt gran, podent arribar a repetir-ne algunes i no utilitzar-ne d’altres. El que sí que fa a nivell de “bloc” és realitzar tantes passades com la mida del *block_size*, partint del primer *token*, per preveure’n el següent, continuant processant els dos primers, per predir el tercer; i així successivament fins a haver processat tots els *tokens* per predir el de la posició el *block_size* més un.

El model avalua el resultat de cada passada utilitzant la funció de pèrdua (*loss function*) basada en l’entropia creuada al mesurar la divergència entre els resultats predits i la distribució correcta. Això es fa sobre cada posició de la seqüència completa, assignant valors més alts quan més s’apropen a mots correctes. Mitjançant l’algorisme d’optimització “descens del gradient” (*gradient descent*) el procés busca de minimitzar el resultat de la funció de pèrdua modificant els paràmetres de la matriu en cada iteració. Un dels perills d’aquesta pràctica és la ràpida adaptació de la matriu de pesos al les particularitats del conjunt d’entrenament (sobreajustament) provocant rigidesa al model impeding la predicció de res que no s’hagi entrenat. Per la banda contrària, el poc entrenament provoca resultats aleatoris en les prediccions. Per assolir una bona curva d’aprenentatge es requereix un equilibri de la mida del conjunt d’entrenament, el nombre d’iteracions, i la taxa d’aprenentatge. També es poden aplicar mecanismes de “regularització” com el L1, que penalitza la funció de pèrdua, el L2 que distribueix l’impacte de les funcionalitats, així com

¹ Més detall sobre els hiper-paràmetres del NanoGPT a l'[annex](#)

altres mètodes com la reducció d'iteracions, l'aturada prematura, l'ús de dades fragmentàries (*sparse*), el *dropout*.

Per últim, una de les estratègies a seguir és l'optimització d'hiperparàmetres (*Hyperparameter tuning*) utilitzant eines per automatitzar la consecució de diferents entrenaments sobre un mateix conjunt de dades amb diferents hiperparàmetres per tal de comparar-ne els resultats i trobar la millor combinació dels mateixos.

4. Metodologia

S'han realitzat diversos entrenaments de models amb enfocaments diferents tan en la morfologia del conjunt de dades com en la parametrització de l'entrenament, comparant-ne els resultats. A continuació es mostra un exemple de les aplicacions que conformen el conjunt de dades descrit des d'un punt de vista a alt nivell:

- Aplicació 1:

```
<!--"Descripció aplicació 1 amb funcionalitat 1 (paràmetre A),  
funcionalitat 2 (paràmetres B i C) i funcionalitat 3" -->  
<Aplicació 1>  
<Funcionalitat 1 "paràmetre A">  
<Funcionalitat 2 "paràmetres B" "paràmetre C">  
<Funcionalitat 3>
```

- Aplicació 2:

```
<!--"Descripció aplicació 2 amb funcionalitat 1 (paràmetre A) i  
funcionalitat 3"-->  
<Aplicació 2>  
<Funcionalitat 1 "paràmetre A">  
<Funcionalitat 3>
```

- Aplicació 3:

```
<!--"Descripció aplicació 3 amb funcionalitat 4"-->  
<Aplicació 3>  
<Funcionalitat 4>
```

Quer Masagué Deu

Cadascuna consta de dues parts diferenciades:

- Una descripció (delimitada per <!-- i -->) on es fa menció a les seves funcionalitats, així com als paràmetres que corresponen a cadascuna. Aquesta és la clau que es vol fer servir per indicar al model com es requereix que sigui l'aplicació generada.
- El codi de l'aplicació que implementa cada funcionalitat descrita a al punt anterior amb els seus paràmetres.

Aquests en són alguns exemples reals del conjunt de dades 14 que es marquen amb diferents colors per indicar les correlacions entre conceptes. Aquest conjunt està format per aplicacions simulades per moure fitxers des d'una carpeta d'origen a una de destí. Aquesta és la funcionalitat base anomenada ruta. Primer s'indica el tipus d'aplicació (**simple**, **doble** o triple) segons si gestiona una ruta o varies. A continuació hi ha les funcionalitats: **cap**, la **funcionalitat 1** i/o la **funcionalitat 2**. Aquestes funcionalitats fan referència a blocs de codi que afegeixen algun valor afegit a la ruta (com filtrat de noms de fitxer, o persistència a base de dades). Solen utilitzar algún paràmetre adicional que en aquest cas s'ha simplificat a una combinació formada per l'**origen** i el **destí**. A continuació la funcionalitat opcional **log** (que mostra traces de les operacions). Per últim s'indiquen els paràmetres de les rutes dels fitxers (**origen** i **destí**).

Aplicació tipus "simple" (que "mou fitxers" desde l'origen **FGZOM** al destí **ZUEDRNZGO**) amb **funcionalitat 1** (paràmetre **FGZOM-ZUEDRNZGO**):

nom fitxer	mover_archivos_simple_fun1_application_FGZOM_ZUEDRNZGO.xml
descripció	<!-- Mover simple de FGZOM a ZUEDRNZGO--> <!-- Con funcionalidad particular1-->
cos aplicació	<xml> <!-- Funcionalidad particular1--> <funcionalidad particular1="FGZOM-ZUEDRNZGO"> <source>FGZOM</source> <target>ZUEDRNZGO</target> </xml>

Aplicació tipus "simple" (que "mou fitxers" desde l'origen **POBJKTG** al destí **EQHA**) amb **funcionalitat 2** (paràmetre **POBJKTG-EQHA**):

nom fitxer	mover_archivos_simple_fun2_application_POBJKTG_EQHA.xml
descripció	<!-- Mover simple de POBJKTG a EQHA--> <!-- Con funcionalidad particular2-->

cos aplicació	<pre><xml> <!-- Funcionalidad particular2--> <funcionalidad particular2="POBJKTG-EQHA"> <source>POBJKTG</source> <target>EQHA</target> </xml></pre>
---------------	---

Aplicació tipus “doble”(que “mou fitxers” desde l’origen JSLJFI A al destí RBNJOIA i “recull fitxers” de RBNJOIB i els porta a JSLJFIB) amb el tipus de funcionalitat log :

nom fitxer	mover_archivos_doble_log_application_JSLJFI_RBNJOI.xml
descripció	<pre><!-- Mover doble de JSLJFI a RBNJOI--> <!-- Con log--></pre>
cos aplicació	<pre><xml> <!-- Ida de JSLJFIA a RBNJOIA--> <source>JSLJFIA</source> <target>RBNJOIA</target> <!-- Vuelta de RBNJOIB a JSLJFIB--> <source>RBNJOIB</source> <target>JSLJFIB</target> <!-- log--> <log message="[JSLJFI-RBNJOI-connect] Sent file to JSLJFIB and obtained file. Recovered file from RBNJOIB"/> </xml></pre>

S’acompanyen més exemples d’altres tipus d’entrenaments així com més combinacions a l’[Annex 4](#).

L’objectiu principal és poder generar aplicacions noves utilitzant com a indicació al model una descripció amb el nom i les funcionalitats volgudes. Per a assolir aquest objectiu es planteja un sistema d’avaluació de la qualitat del model segons si presenta una aplicació ben formatada (marcant-lo com a invàlid en cas contrari), amb les funcions que s’esperen per a l’entrada indicada, així com de la capacitat de substitució del paràmetres de les funcions, i per últim del seu grau de capacitat creativa, és a dir, de si consta d’habilitat per generar noves combinacions de funcionalitats que no es troben al conjunt d’entrenament. A mode d’exemple:

Es reconeixerà un assoliment de grau baix l’obtenir un model vàlid amb capacitat creativa limitada a estructures existents. Per exemple:

Consulta:	“Descripció aplicació 4 amb funcionalitat 1 (paràmetre A), funcionalitat 2 (paràmetres B i C) i funcionalitat 3”
-----------	---

Quer Masagué Deu

Resposta del model:	<!--"Descripció aplicació 4 amb funcionalitat 1 (paràmetre A), funcionalitat 2 (paràmetres B i C) i funcionalitat 3"--> <Aplicació 4> <Funcionalitat 1 "paràmetre X"> <Funcionalitat 2 "paràmetre X" "paràmetre X"> <Funcionalitat 3>
---------------------	---

El grau mitjà es considera l'obtenir un model amb certa capacitat creativa en casos senzills de combinació: per exemple al sol·licitar la funcionalitat 3 i 4 combinació no existent al conjunt d'entrenament:

Consulta:	"Descripció aplicació 5 amb funcionalitat 3 i funcionalitat 4"
Resposta del model:	<!--"Descripció aplicació 5 amb funcionalitat 3 i funcionalitat 4"--> <Aplicació 5> <Funcionalitat 3> <Funcionalitat 4>

Per últim l'assoliment de l'objectiu final es reconeix quan a part dels punts anteriors, el model demostra capacitat creativa al generar aplicacions a partir de qualsevol tipus de combinació i a més és capaç de substituir els paràmetres indicats correctament:

Consulta:	"Descripció aplicació 6 amb funcionalitat 2 (paràmetres B i C) i funcionalitat 4"
Resposta del model:	<!--"Descripció aplicació 6 amb funcionalitat 2 (paràmetres B i C) i funcionalitat 4"--> <Aplicació 6> <Funcionalitat 2 "paràmetre B" "paràmetre C"> <Funcionalitat 4>

Aquest últim nivell s'ha hagut de descomposar de graus de compliment obtinguts, en el cas de fer bé tots els passos però cometre algun tipus d'error en la substitució d'alguns dels paràmetres.

Quer Masagué Deu

Els primers conjunts de dades utilitzats² s'han preparat a partir dels XMLs originals cedits per l'empresa col·laboradora (amb informació confidencial suprimida). El segon *dataset* s'han emprat XMLs "retallats" amb només informació essencial per tal de reduir la mida del context informació no necessària per l'experiment ja que per problemes de memòria i rendiment s'ha hagut de limitar el `block_size` a 1024 KB. El tercer *dataset* s'ha aplantillat un dels XML i se n'han generat còpies amb variables aleatoritzats utilitzant el programa dissenyat per a aquesta finalitat. Per al quart *dataset* s'han emprat 3 XMLs aplantillats. El cinquè i sisè són variacions del tipus d'aleatorització emprada, al primer s'ha utilitzat un diccionari, i al segon s'ha simplificat els símbols aleatoris a xifres i lletres.

Tota la varietat dels scripts dissenyats per a la generació automatitzada de XML es comparteix al repositori públic:

<https://github.com/quermd/TFC>

A part dels XMLs amb informació propietària s'ha creat també un conjunt de dades nou (amb molt poca relació amb els fitxers originals) a partir d'XMLs molt simplificats amb variables curtes aleatoritzades a partir de lletres minúscules. Les variants emprades a diferents entrenaments També es troben disponibles a les carpetes corresponents del repositori (xmlAutomationV3, xmlAutomationV5).

Per últim s'han millorat aquests XMLs reduïts afegint "funcionalitats" amb atributs parametrizables com es pot veure a continuació (generats amb l'script xmlAutomationV6) :

```
<!-- Mover triple de ${ORIGEN}A a ${DESTINO}A y de ${ORIGEN}B a ${DESTINO}B-->
<!-- Con funcionalidad particular1-->
<!-- Con Log-->
<xml>
  <!-- Funcionalidad particular1-->
  <funcionalidad particular1="${ORIGEN}${DESTINO}">
    <!-- Ida de ${ORIGEN}A a ${DESTINO}A-->
    <source>${ORIGEN}A</source>
    <target>${DESTINO}A</target>
    <!-- Vuelta de ${DESTINO}A a ${ORIGEN}A-->
    <source>${DESTINO}A</source>
    <target>${ORIGEN}A</target>
    <!-- Ida de ${ORIGEN}B a ${DESTINO}B-->
    <source>${ORIGEN}B</source>
    <target>${DESTINO}B</target>
    <!-- Vuelta de ${DESTINO}B a ${ORIGEN}B-->
    <source>${DESTINO}B</source>
```

² Aquests conjunts de dades no es comparteixen públicament.

Quer Masagué Deu

```
<target>${ORIGEN}B</target>
<!-- Log-->
<log message="[${ORIGEN}-${DESTINO}-connect] Sent file to ${DESTINO}A and obtained
file. Recovered file from ${DESTINO}B"/>
</xml>
```

Amb aquests scripts es generen de forma aleatoria noms que s'assignen tant als paràmetres de les plantilles com al nom del fitxer. A continuació es presenta una mostra dels noms aleatoris de fitxers i variables creats:

- mover_archivos_triple_application_qPAkg5IY_Kn3K6HP.xml
- mover_archivos_simple_application_pG_tVCXeIFPfiB.xml
- mover_archivos_doble_application_Dwq6hQ_vtxBg4S1IYCnzC6.xml
- mover_archivos_multiple_application_I9E8_eoJT5ogma.xml

Marcats en blau es mostra el nom de l'aplicació (amb una sèrie de funcionalitats particulars). Marcat en verd el primer dels paràmetres, i en groc el segon. Els scripts també s'encarreguen de substituir les variables dins contingut de cada fitxer (a partir de la plantilla corresponent a simple, doble, triple o múltiple) amb els mateixos valors que s'han informat al nom del fitxer al les variables marcades amb \${ORIGEN} i \${DESTINO}.

Per a la realització dels entrenaments s'ha utilitzat un equip personal amb les característiques que es mostren a la taula 2. S'ha utilitzat un sistema operatiu Ubuntu 23.4 amb dependències següents: python 3.8, Pytorch 2.0, transformers 4.28, tiktoken 0.3, wandb 0.15, tqdm 4.65, nvidia-ml-py3 7.352, datasets 2.12³.

Taula 2.

Recursos de màquina utilitzats

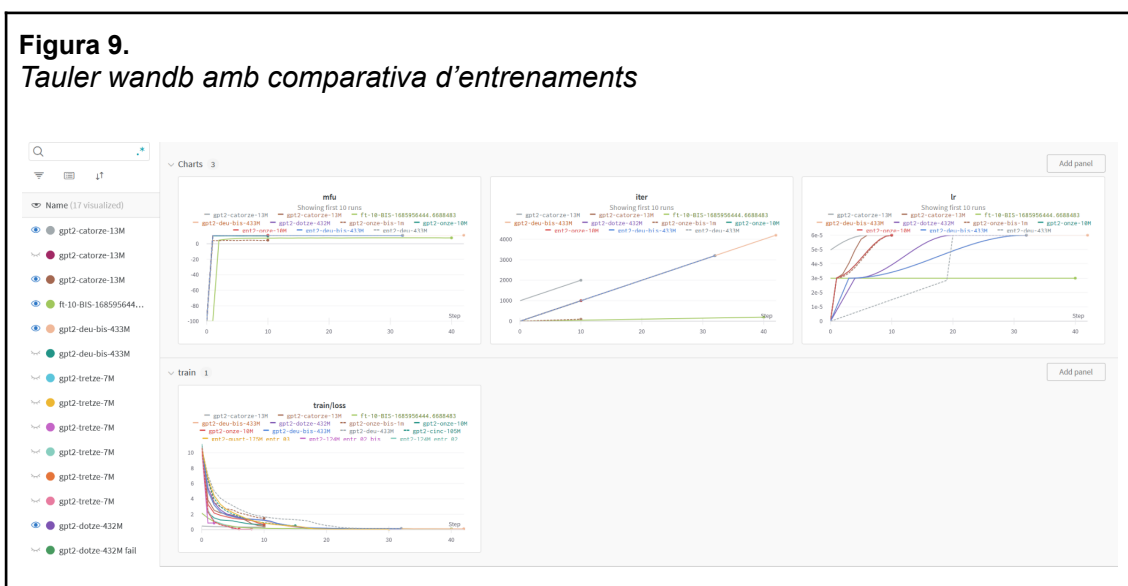
Processador	AMD Ryzen 5 3600 (4.2 GHz 65W)
Memòria Ram	32 Gb Corsair Vengeance LPX DDR4 3200MHz PC4-25600 2x16GB CL16)
Disc dur	Corsair MP600 Gen4 PCIe 1TB amb una velocitat de lectura de fins a 4950MB/s
Tarjeta gràfica	Zotac Gaming GeForce RTX 4070 Twin Edge OC 12GB GDDR6X DLSS3
Placa base	MSI X570-A Pro

³ La resta de les dependències es poden veure al fitxer requeriments.txt del repositori públic del treball

Font d'alimentació	Seasonic FOCUS Gx-550
--------------------	-----------------------

Respecte al configuració dels entrenaments s'ha mantingut invariable una part de la parametrització de l'algorisme NanoGPT original i s'ha treballat en un grup reduït de hiperparàmetres cercant els millors resultats. Els principals paràmetres amb els que s'ha treballat són el *block_size*, el *batch_size* i el *max_iters* velocitat d'entrenament, corba d'aprenentatge, dropout⁴.

Tots els entrenaments es registren amb l'eina al núvol wandb⁵ que permet visualitzar les mètriques en forma de gràfiques i comparar els resultats dels diferents entrenaments (figura 9). A més en manté un registre de les configuracions, traces i recursos de sistema.



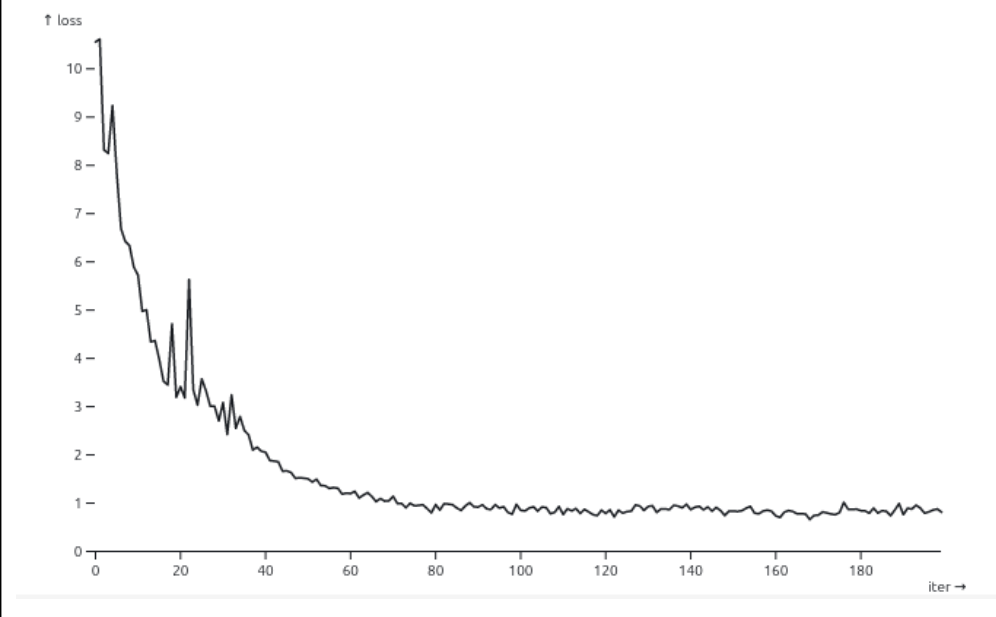
Un cop acabat un entrenament s'analitzen en els valors de *train\loss* i *val\loss*, que indiquen, respectivament, els valors de la funció de pèrdua contra el conjunt de dades d'entrenament i contra el conjunt de dades d'avaluació.

D'aquests valors se n'observa la corba resultant que indica l'efectivitat dels hiperparàmetres com "gradient descent" o de la taxa d'aprenentatge (figura 10). Una corba que no segueix aquesta forma logarítmica denota una mala configuració de l'entrenament. Arribat al punt on s'aplana la cua de la corba es considera que l'entrenament no pot trobar noves configuracions i que per tant s'està sobreajustant.

⁴ Més informació a [l'annex Detall configuració completa d'un entrenament](#)

⁵ El repositori personal publicat a <https://wandb.ai/quer/owt> amb accés privat.

Figura 10
Corba de pèrdua



La última part de l'avaluació del model és qualitativa i manual: es verifica la qualitat del model utilitzant combinacions d'indicacions (*prompts*) i observant-ne els resultats. El mètode de combinar les indicacions influeix notablement sobre els resultats i per tant s'ha buscat optimitzar els resultats al interactuar amb aquests tipus de models fent el que es diu "*prompt engineering*". Una d'aquestes tècniques es basa en passar com a part de la consulta un o més exemples de com es vol que el model respongui, tal i com Brown et al (2020) expliquen a la seva obra. S'ha creat un programa en python⁶ que permet interactuar amb el model poden consultar de forma consecutiva. Aquest programa també permet la consulta a partir del text en un fitxer per permetre consultes llargues.

Amb les conclusions extretes dels resultats de l'entrenament es planteja un mètode de millora i es repeteix el procés. Els canvis que s'han fet entre els entrenaments amb més impacte sobre els resultats són les millores en el tipus de dades del conjunt d'entrenament. Aquestes van des d'afegir descripcions a les funcionalitats, afegir els noms de fitxer entre els XML, a canviar el mètode de generació dels noms. Un exemple d'aquest seria:

(...)
persistent-id="edu.company.DeFiGo.PaToWe.app"

⁶ Disponible al repositori públic de l'autor:
https://github.com/quermd/TFC/blob/main/q_sample_interactive.py

```
<!-- Esta aplicación mover archivos simple application DeFiGo PaToWe sirve para  
mover archivos de DeFiGo a PaToWe y obtener una respuesta de PaToWe a  
DeFiGo -->  
(...)
```

Per a la generació aleatoritzada dels noms de les variables s'han enfocat diferents estratègies: des d'utilitzar un conjunt aleatori de síl·labes, utilitzant un diccionari de paraules, o directament restringir-lo a un conjunt de caràcters determinat. Tanmateix, la modificació dels hiperparàmetres també ha resultat en canvis significatius en els resultats dels entrenaments.

Una altra estratègia utilitzada ha sigut la d'analitzar els resultats d'utilitzar models pre-entrenats de GPT sotmesos a un procés de refinament amb el nostre repositori de programes o amb el conjunt de dades preparat tal i com es documenta a Lim et al (2021). Aquests s'avaluen de la mateixa manera i a més se n'avalua la capacitat d'entendre el llenguatge natural tal i com ho fa el model preentrenat original.

Per últim, s'ha utilitzat el ChatGPT d'OpenAI per a indicar-li que generi aplicacions tal i com ho s'ha fet a l'estudi i adjuntant com a context els últims conjunts d'aplicacions (simulades) i s'e n'avalua el resultat amb el mateix criteri.

5. Resultats

A diferència d'altres exemples d'entrenament de LLMs els resultats d'aquest estudi han obtingut ratios molt baixes de les funcions de pèrdua (generalment per sota del 0.2) però tot i semblar un bon resultat això denota una clara tendència al sobreajustament deguda a la poca diversitat de fonts del dataset (taula 3). Això ha resultat en la limitació clara de la seva capacitat creativa a tasques de rèplica exacta dels XMLs amb substitució de paràmetres i funcionalitats en els millors dels casos tal i com s'havia previst inicialment.

Taula 3

Resultats dels entrenaments

Entrenament	<i>train loss</i>	temps d'entrenament	XML ben format	Substitució de Funcionalitats	Substitució de variables
1	0.082	44min	No	No	No
1 bis	0.067	1h 20min	Sí	No	No
2	0.159	50min	Sí	No	No
refinament	0.063	4min	Sí	No	No
3	0.05	3h	Sí	No	Grau baix
5	0.5	1h30min	Sí	No	No
6	0.049	3h	Sí	No	Grau mig
refinament	0.563	2min	No	No	No
10	0.14	3h 12 min	Sí	no	Grau alt 90%-100%
11	0.132	1h	Sí	No aplica	Grau alt 100%
14	0.333	1h 30 min	Sí	Sí	Grau mig 50%
GPT 3.5	-	-	Sí	Sí	Grau alt 100%

Els entrenaments de l'u al sis s'han realitzat amb els tres primers datasets, i han reportat mals resultats, no obtenint un model vàlid amb capacitat de substitució dels paràmetres. Només en el cas de l'entrenament tres s'ha observat un resultat correcte i amb certa creativitat sota determinades condicions. Es fan proves amb tècniques per reduir el sobreajustament com el *dropout* obtenint una clara millora en els resultats. S'observa l'estreta relació entre el mètode escollit per aleatoritzar els noms de variables durant la generació dels XMLs i la capacitat creativa del model. La possibilitat de substitució de paràmetres es veu restringida al conjunt de caràcters utilitzat durant les aleatoritzacions. A més l'excessiu sobreajustament impedeix utilitzar paraules similars a d'altres del propi codi del XML (parts no variables de la plantilla). S'ha referit aquesta última observació com a "paraules reservades".

Als entrenaments dels datasets 10⁷ i 11 s'han observat importants millores en els resultats. Arribant a obtenir els models amb capacitat de substituir els paràmetres correctament i flexibilitat a l'hora de construir nous XMLs no existents al conjunt d'entrenament (a l'entrenament del dataset 11). Tal i com s'havia observat a la primera part dels entrenaments l'elecció dels noms de les variables és determinant per a l'obtenció de bones substitucions, obtenint millors resultats emprant variables curtes. Tot i això aquest dataset es constitueix per XMLs on les funcionalitats es limiten al tipus d'aplicació però no en la combinació de funcionalitats diferents dins el XML.

Tot i utilitzar tècniques de "few shot prompting" no s'aconsegueix millorar els resultats. Els resultats correctes s'obtenen introduïnt el nom del fitxer, o bé la línia de la descripció.

La substitució sense errades (grau alt d'assoliment 100% correcte) de paràmetres s'ha obtingut amb l'entrenament 11 bis. Com es pot observar la especificació del tipus d'aplicació (mover archivos doble) així com els paràmetres (origen és Z8 i destí és X9) es substitueixen correctament per a totes les rutes presents al programa doble:

Exemple de resultat de l'entrenament 11:

Consulta

```
<!-- Mover de ab a zu-->
```

Resposta

```
<!-- Mover de ab a zu-->
<xml>
<source>ab</source>
<target>zu</target>
</xml>
```

Consulta

```
<!-- Mover de abc a vxyz-->
```

Resposta

```
<!-- Mover de abc a vxyz-->
<xml>
<source>abc</source>
<target>vxyz</target>
</xml>
```

Exemple de resultat de l'entrenament 10:

Consulta

```
mover_archivos_doble_Z8_X9.xml
```

Resposta

```
mover_archivos_doble_Z8_X9.xml
<?xml version="1.0" encoding="UTF-8"?>
```

⁷ Veure annex [Detall entrenament 10 bis](#)

Quer Masagué Deu

```
<blueprint xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0"
  xmlns:cm="http://aries.apache.org/blueprint/xmlns/blueprint-cm/v1.1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <!-- mover_archivos_doble_application Z8 X9 -->
  <!-- nombre corto de aplicación: Z8-X9-connect -->
  <!-- paquete de aplicación: net.company.doble.Z8.X9.connect -->
  <!-- Esta aplicación mover archivos doble application Z8 sirve para mover de Z8B1
  a X9B1IN y obtener una respuesta de X9B1OUT a Z8B1IN
  y además para mover de Z8B2 a X9B2 y obtener una respuesta de
  X9B2OUT a Z8B2IN -->

  (...)
  <!--camelContext de mover archivos doble de Z8 a X9 -->
  <camelContext id="Z8-X9-connect-camelContext"

  (...)

    <!--ruta para mover archivos de Z8B1 a X9B1IN -->
    <!-- Route1: Z8B1 to X9B1IN -->
    <route id="Z8-X9-connect-fuse-routeX9B1IN">
      <from
uri="master:Z8-X9-connect-fuse-routeX9B1IN:{{input.sftpRoute.X9B1IN}}"/>
      <log message="[Z8-X9-connect] Received file from SFTP /interface:
${file:name}"/>
      (...)
      <to uri="{{output.fileFolder.X9B1IN}}"/>
      <log message="[Z8-X9-connect] Sent file ${file:name} to remote
folder Interface route {{output.fileFolder.X9B1IN}}"/>
      (...)
      <log loggingLevel="ERROR" message="[Z8-X9-connect] [routeX9B1IN]
ERROR: ${exception.stacktrace}"/>
      <setHeader headerName="routeMailSubject">
        <simple>Z8B1-X9B1IN</simple>
      </setHeader>
      <setHeader headerName="description">
        <simple>Error processing file ${file:name} from Z8B1 to X9B1IN
</simple>
      (...)
    <!--ruta para mover archivos de Z8B2 a X9B2IN -->
    <!-- Route3: Z8B2 to X9B2IN -->
    <route id="Z8-X9-connect-fuse-routeX9B2IN">
      <from
uri="master:Z8-X9-connect-fuse-routeX9B2IN:{{input.sftpRoute.X9B2IN}}"/>
      <log message="[Z8-X9-connect] Received file from SFTP /interface:
${file:name}"/>
      (...)
      <to uri="{{output.fileFolder.X9B2IN}}"/>
      <log message="[Z8-X9-connect] Sent file ${file:name} to remote
folder Interface route {{output.fileFolder.X9B2IN}}"/>
      <onException redeliveryPolicyRef="Redeliveries">
        <exception>java.lang.Exception</exception>
        <log loggingLevel="ERROR" message="[Z8-X9-connect]
[routeX9B2IN] ERROR: ${exception.stacktrace}"/>
        <setHeader headerName="routeMailSubject">
```

Quer Masagué Deu

```
<simple>Z8B2-X9B2IN</simple>
</setHeader>
<setHeader headerName="description">
<simple>Error processing file ${file:name} from Z8B2 to X9B2IN
</simple>
(...)

<!--ruta para mover archivos de Z8B3 a X9B3IN -->
<!-- Route6: Z8B3 to X9B3IN -->
<route id="Z8-X9-connect-fuse-routeX9B3IN">
  <from
uri="master:Z8-X9-connect-fuse-routeX9B3IN:{{input.sftpRoute.X9B3IN}}"/>
  <log message="[Z8-X9-connect] Received file from SFTP /interface:
${file:name}"/>
  (...)
  <to uri="{{output.fileFolder.X9B3IN}}"/>
  <log message="[Z8-X9-connect] Sent file ${file:name} to remote
folder Interface route {{output.fileFolder.X9B3IN}}"/>
  <onException redeliveryPolicyRef="Redeliveries">
  <exception>java.lang.Exception</exception>
  <log loggingLevel="ERROR" message="[Z8-X9-connect]
[routeX9B3IN] ERROR: ${exception.stacktrace}"/>
  <setHeader headerName="routeMailSubject">
  <simple>Z8B3-X9B3IN</simple>
  </setHeader>
  <setHeader headerName="description">
  <simple>Error processing file ${file:name} from Z8B3 to X9B3IN
</simple>
(...)

<!--ruta para mover archivos respuesta de X9B1OUT a Z8B1IN -->
<!-- Route7: X9B1OUT to Z
```

S'ha intentat combinar funcionalitats provinents de diferents tipus d'aplicacions sense obtenir bons resultats.

Finalment amb l'entrenament 14 per falta de capacitat d'entrenament (temps i potència) no s'ha pogut entrenar suficient com per a obtenir resultats 100% vàlids. Un exemple d'interacció s'acompanya a continuació. Un dels paràmetres (ABC) no ha sigut substituït correctament essent substituït per un altre paràmetre "inventat" (EDPI) i a més cometent algunes errades afegint un caràcter de més (B) en algunes ocasions:

Consulta

```
mover_archivos_triple_func2_ABC_XYZ.xml
```

Resposta

```
mover_archivos_triple_func2_ABC_XYZ.xml
<!-- Mover triple de EDPIA a XYZA y de EDPBIB a XYZB-->
<!-- Con funcionalidad particular2-->
<xml>
  <!-- Funcionalidad particular2-->
  <funcionalidad particular2="XYZ">
    <!-- Ida de EDPIA a XYZA-->
    <source>EDPIA</source>
    <target>XYZA</target>
    <!-- Vuelta de XYZA a EDPIA-->
    <source>XYZA</source>
    <target>EDPIA</target>
    <!-- Ida de EDPBIB a XYZB-->
    <source>EDPBIB</source>
    <target>XYZB</target>
    <!-- Vuelta de XYZB a EDPBIB-->
    <source>XYZB</source>
    <target>EDPBIB</target>
    <!-- log-->
    <log message="[EDPIB-XYZ-connect] Sent file to XYZA and obtained file
and Sent file to XYZB and obtained file "/>
  </xml>
```

L'anàlisi de la capacitat de substituir funcionalitats i paràmetres en aquest exemple es mostra com es demana el tipus tripe, amb la funcionalitat 2 i els paràmetres ABC i XYZ i sobté un resultat parcialment correcte, però el model hi afegeix el "log" que no se li ha demanat.

En les diferents interaccions amb aquest últim model (14) s'ha pogut observar capacitat generativa alhora d'encaixar les diferents funcionalitats ("funcionalidad particular1", "funcionalidad particular2" i el log) sota demanda, tot i que s'han observat problemes en la substitució de les variables així com en la introducció d'elements no sol·licitats.

Els resultats de les proves al refinar models de GPT2 amb els repositoris propis han sigut insatisfactoris, obtenint resultats incorrectes. El problema radica en la limitació dels recursos no podent carregar el model GPT2 a la escassa memòria de la GPU (12GB), quedat aquest exercici sempre limitat a una versió reduïda del mateix ("gpt2-medium"). Cal mencionar que per entrenar el model GPT 2 van ser necessàries 8 gràfiques A100 de 40GB cadascuna i durant un temps de 4 dies.

Per salvar l'impediment de dur a terme un refinament adequat del model GPT2, s'ha optat per consultar el CHAT GPT d'OpenAI (versió del model GPT 3.5) obtenint resultats perfectes tant en substitució de variables com de combinació funcionalitats (veure [annex Detall xat amb GPT](#)).

6. Discussió

Existeix un debat obert sobre la limitació d'aquest tipus de models a convertir-se en mers "lloros" que només repeteixen el que se'ls diu sense cap capacitat creativa (Wikipedia 2023 b). Els primers models que s'han entrenat efectivament eren aquest tipus. Però amb els últims entrenaments s'ha vist el potencial creatiu i generatiu que poden arribar a tenir, tot i partir d'entrenaments limitats. Els LLMs com el GPT4, el Bard o LLaMA superen amb escreix aquesta creativitat essent capaços de generar contingut nou a partir de senzilles instruccions per a manipular textos, com també s'ha pogut demostrar.

7. Valoració econòmica

Els major cost ve donat pel valor de la targeta gràfica ha costat 640 €. Els consums elèctrics són molt petits: en els moments de l'entrenament l'equip informàtic consumeix de mitjana 500W/h i el preu de l'energia és de 0,16kW/h resultant un cost de 8 cèntims la hora d'entrenament. En total s'han realitzat unes 20 hores d'entrenament el que resulta a un cost total en electricitat de 1,6€.

Taula 4

Cost energètic dels entrenaments

entrenament de 1 hora	500W x 1h x 0,16 kW/h	0,08 €
entrenament de 3 hores	500W x 3h x 0,16 kW/h	0,24 €
total entrenaments	500W x 20h x 0,16 kW/h	1,60 €

Aquests costos poden arribar a ser viables si es comparen amb els cost dels recursos en programadors de les alternatives: ja sigui per crear les plantilles dels programes més característics, i desenvolupar o configurar un programa per a poder generar noves aplicacions a partir d'aquestes plantilles. O bé per desenvolupa cada nova aplicació a mà.

L'opció de comprar hardware per entrenar un model propi pots suposar un cost molt elevat sobretot en targetes gràfiques (una NVIDIA A100 costa 14.000€). Les grans empreses com OpenAI han invertit molts milions de dòlars en l'entrenament d'aquests models i han necessitat utilitzar fins a milers d'aquests tipus de targetes durant setmanes. Tot i que existeixen models preentrenats reduïts opensource (com gpt4all⁸) que es poden refinar amb GPUs de la gama dels 600€-2000€.

Aquesta opció al presentar un gran cost de capital (CAPEX) es pot considerar inviable aquesta opció si no és per força major, com la impossibilitat de fer ús de models externalitzats.

Es pot considerar viable la contractació dels serveis oferts per les diferents empreses d'aquest mercat, com OpenAI, Google o Amazon entre d'altres amb costos molt baixos tant d'entrada com, per al cas que es planteja en aquest estudi de manteniment (OPEX) ja que no se'n faria un us intens.

Per exemple a OpenAI l'ús del model GPT4 amb un context de 8KB suposa un cost de 0.03\$ cada 1000K tokens d'indicacions, i 0.06€ cada 1K tokens de les respostes. El total dels 30 XMLs corresponen a 50.000 tokens. Un sol XML són uns 7325 tokens. Si s'opta per fer un refinament complet del seu model amb el repositori propi, per exemple amb el seu model Davinci suposa tan sols un cost de refinament de \$0.03 cada 1K tokens i un cost de \$0.12 / 1K les indicacions⁹. Això suposa un total de 0,2€ del refinament amb els 30 XML i un cost inferior a 1 € per cada consulta d'un XML nou.

Aquests preus tan reduïts, fan viable la utilització d'aquests models sempre que no es topi amb impediments legals, de privacitat o confidencialitat.

⁸ <https://gpt4all.io/index.html>

⁹ <https://openai.com/pricing>

8. Conclusions

8.1. Conclusions

S'han trobat moltes limitacions sobre l'objectiu original, havent de reduir la complexitat dels programes, així com havent de crear plantilles per generar còpies per a poder entrenar els models. Això és contraproductiu ja que el que es pretenia era eliminar la necessitat de crear plantilles. Tot i així no hi ha hagut més remei al disposar d'una variabilitat de poc més de 4 KB de codi. Com s'ha pogut observar al comparar els resultats dels entrenaments 11 i 11 bis, el primer amb 300mil mostres i el segon només amb 300, són necessaris gran nombre de dades per a obtenir uns bons resultats. Els conjunts de dades que ens han funcionat millor eren d' 1 GB.

Tanmateix s'han emprat temps d'entrenament molt reduïts, d'entre una i tres hores, i també s'ha pogut observar la millora dels resultats amb l'augment dels temps. Un dels problemes més significatius ha vingut determinat per la limitació del hardware de la mida del context entrenable, inferior a la mida mitjana dels XMLs. Això pot arribar a ser un problema per a que el model pugui recollir tota el context dels XMLs. En el moment que s'ha reduït la mida dels XMLs (eliminant les parts no essencials i simplificant-los) s'han començat a obtenir els resultats esperats. Això denota que seria possible obtenir bons resultats a base d'escalar la infraestructura. Finalment, les proves realitzades amb XMLs simulats (no originaris de l'empresa col·laboradora) com a context contra model GPT3.5 d'OpenAI sol·licitant els nous programes han obtingut resultats de gran qualitat¹⁰.

S'ha valorat l'opció de utilitzar algun model preentrenat de domini públic i refinar-lo. Per això és necessita infraestructura amb més poder de computació o pagar serveis al núvol d'aprenentatge automàtic. La primera opció es descarta per el seu elevat cost. La segona s'ha plantejat i provat l'èxit amb els serveis d'OpenAI. Aquesta tipus d'opció degut al seu baix cost, i les millores en les condicions de privacitat que les empreses ofereixen als seus clients empresarials, es considera pot ser vàlida per a la generació automatitzada d'aplicacions. Caldria, això sí, implementar el client a la API del servei, però aquests són molt simples d'implementar i a més pot ser compartit per a molts usos dins l'empresa, i obtenir més funcionalitats a part de la generació d'aplicacions. Fins i tot se li pot demanar a un model amb reconeixement de llenguatge natural que realitzi l'apantillament d'aplicacions.

¹⁰ Veure ANNEX [Conversa amb ChatGPT](#)

Es pot considerar que l'ús d'aquest tipus de models per a la realització de tasques com aquestes, que es podrien resoldre amb plantilles, pot semblar una obra de “sobre-enginyeria”, no sempre es disposa d'aquestes quan se les necessita i les empreses no acostumen a veure amb bons ulls un augment del CAPEX per una futura possible reducció en l'OFEX. El seu baix cost i risc poden animar a la seva aprovació i les possibilitats que obren són moltes: per exemple integrant-ne l'ús en d'aplicacions que, a través de l'API dels models, els sollicitin indicacions parametritzades per a generar resultats es puguin executar directament per línia de comandes.

Cal recordar però que de moment totes aquestes tasques “automatitzades” requereixen una supervisió per verificar el correcte acompliment de les indicacions i resoldre possibles problemes. Tanmateix és necessària una correcta formulació de les indicacions que requereix habilitat o coneixements tècnics del funcionament dels models. Aquest handicap es podria solventar utilitzant programari que “decori” les consultes dels usuaris aplicant les tècniques conegudes d'enginyeria d'indicacions. És important també, en aquest sentit permetre el diàleg amb el model, és a dir que a les indicacions que se li fan se li afegixi el context previ per a poder matisar les respostes, sol·licitar transformacions i utilitzar-les en successives interaccions.

El coneixement adquirit respecte el funcionament dels LLM així com les lliçons apreses sobre l'enginyeria de les indicacions ha servit per obrir iniciatives dins la pròpia empresa per a facilitar l'ús d'aquests models de forma segura, confidencial i senzilla per a tot tipus d'usuaris.

8.2. Línies de futur

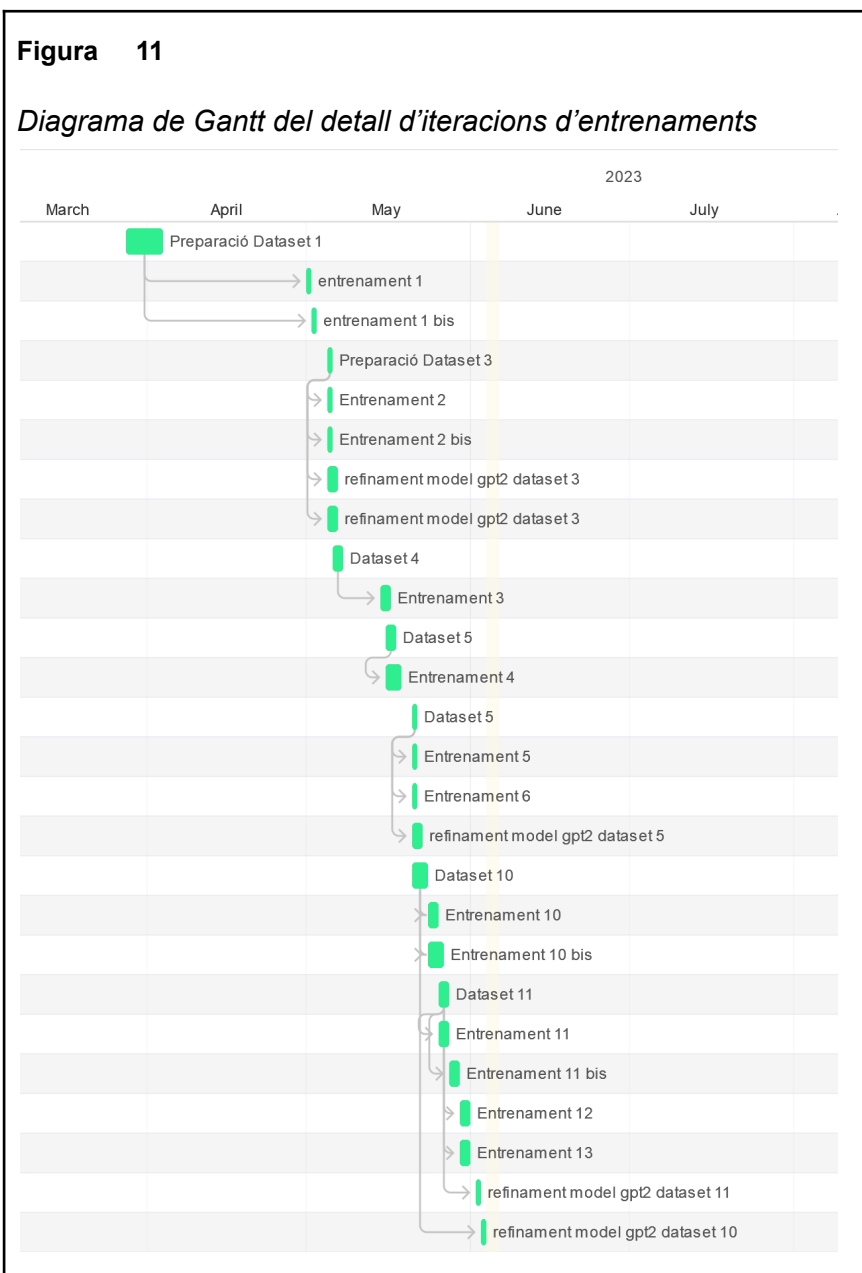
Crear una interfície per a interactuar amb models per a poder indicar la creació de noves aplicacions, a partir de funcionalitats conegudes del conjunt d'aplicacions propi entrenament.

Estudiar els possibles usos i riscos de la utilització d'interfícies màquina-màquina (b2b) per a que diferents aplicacions s'integrin amb les APIs dels models i puguin executar indicacions per a obtenir resultats. Estudiar la “decoració” necessària que s'ha de sol·licitar al model per a aquests resultats per que es puguin executar directament a l'entorn de l'aplicació original.

8.3. Seguiment de la planificació

Els gran impediment en la planificació ha estat el no disposar dels usuaris proporcionat per l'empresa per a dur a terme la computació al núvol. Això ha provocat que es prenguessin mesures de mitigació com la compra de la tarjeta gràfica (l'anterior, AMD, va suposar grans complicacions per a la computació de degut a la poca compatibilitat entre de les llibreries ROCm i CUDA-NVIDIA).

Les tasques “preparació de dades”, “entrenament del model”, “d'avaluació”, que inicialment s'havien planificat com a una de sola, ha calgut anar-les repetint de forma iterativa, tal i com es mostra a la figura 11.



9. Glossari

Algorisme del gradient descendent. Mètode d'optimització que cerca de forma iterativa trobar el mínim d'una funció.

Attention: L'atenció és un mecanisme clau dels Transformers. Permet als models de llenguatge assignar pesos als diferents trets o paraules d'una seqüència en funció de la seva importància relativa per a la tasca. L'atenció permet captar relacions complexes i de llarg abast entre les paraules i ajuda a millorar la comprensió i generació de llenguatge (Lin et al 2017).

Embedding: L'*embedding* és el procés de representar les paraules o els trets en un espai vectorial continu. Aquest procés té l'objectiu de capturar i preservar les relacions semàntiques entre les paraules. Els *embeddings* són generats mitjançant tècniques com ara Word2Vec o GloVe, i són utilitzats pels LLMs per entendre i generar el llenguatge de manera més eficient.

Enginyeria de les indicacions : De l'anglès *Prompt Engineering* és la tècnica mitjançant la qual aplicant coneixement tècnics del funcionament dels LLMs s'adeqüen les indicacions transmises al model per tal d'obtenir els millors resultats.

Feed-Forward: El *feed-forward* és una part clau de l'arquitectura dels LLMs. Consisteix en capes totalment connectades que apliquen transformacions lineals i no lineals a les representacions de les paraules. Aquestes capes *feed-forward* permeten als models aprendre relacions no lineals entre les paraules i millorar la seva capacitat per capturar i generar informació complexa.

Hiper-paràmetre: Aquells paràmetres emprats per controlar el procés d'entrenament i que no vairen durant l'execució d'aquest. Es diferencien dels paràmetres que es gestionen dins el procés d'entrenament i que poden anar durant les iteracions.

Layer normalization: Mètode d'estandardització (o normalització estadística) d'una capa de la xarxa neural per a millorar l'eficiència dels entrenaments. Es diferencia de la normalització de lots (batch normalization) precisament al aplicar-se només sobre una capa i no sobre tot el lot de processament (Ba et al 2016).

Transformers: Els Transformers són la base fonamental dels LLMs. Aquesta arquitectura es caracteritza per utilitzar múltiples capes d'atenció (*attention*) i eliminar la dependència seqüencial en el processament del llenguatge. En lloc

de les arquitectures recurrents tradicionals, els Transformers permeten un processament paral·lel i més eficient de la informació, cosa que els fa adequats per a models de llenguatge amb una gran quantitat de dades.

10. Bibliografia

Ba, J., Kiros, J.R., i Hinton, G.E. (2016). *Layer Normalization*. ArXiv. <https://arxiv.org/pdf/1607.06450.pdf>.

Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T.J., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., i Amodei, D. (2020). *Language Models are Few-Shot Learners*. ArXiv. <https://arxiv.org/pdf/2005.14165.pdf>.

Gartner. (2023, Maig 18). *Gartner Identifies Six ChatGPT Risks Legal and Compliance Leaders Must Evaluate*. <https://www.gartner.com/en/newsroom/press-releases/2023-05-18-gartner-identifies-six-chatgpt-risks-legal-and-compliance-must-evaluate>.

Kalchbrenner, N., Espeholt, L., Simonyan, K., Oord, A.V., Graves, A., i Kavukcuoglu, K. (2016). *Neural Machine Translation in Linear Time*. ArXiv. <https://arxiv.org/pdf/1610.10099.pdf>

Karpathy, A. [Neural Networks: Zero to Hero] (17 de gener 2023). *Let's build GPT: from scratch, in code, spelled out*. [Vídeo]. Youtube. <https://youtu.be/kCc8FmEb1nY>

Karpathy, A. (2023). *nanoGPT*. [Software]. Dins *GitHub*. Recuperat el 19 de juliol de 2023 de: <https://github.com/karpathy/nanoGPT>

Kosar, V. (2021, desembre 28). *Cross-Attention in Transformer Architecture*. [entrada de blog] *Vaclavkosar* <https://vaclavkosar.com/ml/cross-attention-in-transformer-architecture>

Kosar, V. (2022, juny 5). *Transformer Positional Embeddings and Encodings*. [entrada de blog] *Vaclavkosar* <https://vaclavkosar.com/ml/transformer-positional-embeddings-and-encodings#fourier-sinusoid-positional-encodings-in-bert>

La Malfa, E., Malfa, G.L., Caprioli, C., Nicosia, G., i Latora, V. (2022). *Deep Neural Networks as Complex Networks*. ArXiv. <https://arxiv.org/pdf/2209.05488>

Lin, Z., Feng, M., Santos, C.N., Yu, M., Xiang, B., Zhou, B., i Bengio, Y. (2017). *A Structured Self-attentive Sentence Embedding*. ArXiv.
<https://arxiv.org/pdf/1703.03130.pdf>.

Lim, R., Wu, M. i Miller, L. (2021, desembre 14). *Customizing GPT-3 for your application. Fine-tune with a single command*. [entrada de blog] OpenAI.
<https://openai.com/blog/customizing-gpt-3>

Noticias del Parlamento Europeo. (2023, juny 12). *Ley de IA de la UE: primera normativa sobre inteligencia artificial*.
<https://www.europarl.europa.eu/news/es/headlines/society/20230601STO93804/ley-de-ia-de-la-ue-primera-normativa-sobre-inteligencia-artificial>

Pacho, L. (2023, març 31). Italia bloquea el uso de ChatGPT por incumplir la normativa de protección de datos. *El País*.
<https://elpais.com/tecnologia/2023-03-31/italia-bloquea-el-uso-de-chatgpt-por-incumplir-la-normativa-de-proteccion-de-datos.html>

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., i Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners.
<https://d4mucfpksyww.cloudfront.net/better-language-models/language-models.pdf>

Russell, S.J., i Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*.

Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., i Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15, 1929-1958.

Stanford University (2023). *CS231n: Deep Learning for Computer Vision*. Recuperat el 18 de juny de 2023 de
<https://cs231n.github.io/convolutional-networks/>

Torra i Reventós, V. (2019) *Què és la intel·ligència artificial*. [Recurs d'aprenentatge textual]. Primera edició. Barcelona: Fundació per a la Universitat Oberta de Catalunya (FUOC).

Vaswani, A., Shazeer, N.M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., i Polosukhin, I. (2017). *Attention is All you Need*. NIPS.
<https://arxiv.org/pdf/1706.03762v4.pdf>

Wang, B., Zhao, D., Lioma, C., Li, Q., Zhang, P., i Simonsen, J.G. (2019). *Encoding word order in complex embeddings*. ArXiv,
<https://arxiv.org/pdf/1912.12333.pdf>

Quer Masagué Deu

Stochastic parrot. (2023, Juliol 16). A *Wikipedia*.

https://en.wikipedia.org/w/index.php?title=Stochastic_parrot&oldid=1160483300

Inteligencia artificial generativa. (2023, Juliol 6). A *Wikipedia*

https://es.wikipedia.org/w/index.php?title=Inteligencia_artificial_generativa&oldid=151395764

Yang, J., Jin, H., Tang, R., Han, X., Feng, Q., Jiang, H., Yin, B., i Hu, X. (2023). *Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond*. ArXiv, <https://arxiv.org/pdf/2304.13712.pdf>

11. Annexos

1. Exemple aplicació Camel Blueprint	39
2. Detall configuració complerta d'un entrenament	40
3. Comparativa d'entrenaments realitzats	41
4. Exemples dels conjunts d'entrenament	45
Entrenament 11:	45
Entrenament 12 (s'eliminen fragments):	46
Entrenament 13:	48
Entrenament 14:	49
5. Detall plantilles conjunt entrenament 14	51
6. Detall Entrenament 10 bis	54
7. Proves contra ChatGPT d'OpenAI	59

1. Exemple aplicació Camel Blueprint¹¹

```
<?xml version="1.0" encoding="UTF-8"?>
<blueprint xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:camel="http://camel.apache.org/schema/blueprint"
  xsi:schemaLocation="
    http://www.osgi.org/xmlns/blueprint/v1.0.0
    http://www.osgi.org/xmlns/blueprint/v1.0.0/blueprint.xsd
    http://camel.apache.org/schema/blueprint
    http://camel.apache.org/schema/blueprint/camel-blueprint.xsd">

  <camelContext id="test-ahc" streamCache="true"
    xmlns="http://camel.apache.org/schema/blueprint">

    <route>
      <from uri="timer://foo?repeatCount=1"/>
      <to uri="direct:start"/>
    </route>
    <route>
      <from uri="direct:start"/>
      <setHeader headerName="Exchange.HTTP_METHOD">
        <constant>HEAD</constant>
      </setHeader>
      <setHeader headerName="Exchange.HTTP_URI">
        <constant>http://t.co/R2GE6IbrQL</constant>
      </setHeader>
      <to
uri="ahc:http://changeme?throwExceptionOnFailure=false&clientConfig.max
RequestRetry=3&clientConfig.followRedirect=false" />
        <log message="header location: ${headers.location}" />
        <log message="header Content-Type: ${headers.Content-Type}" />
        <log message="header ${headers}" />
        <to uri="file:///tmp/" />
      </route>
    </camelContext>
  </blueprint>
```

<https://github.com/matteoredaelli/apache-camel-blueprint-samples/blob/master/http-ahc-to-file.xml>

¹¹ Documentació: <https://camel.apache.org/components/2.x/others/blueprint.html>

2. Detall configuració complerta d'un entrenament

```
always_save_checkpoint=true
backend="nccl"
batch_size=6
beta1=0.9
beta2=0.95
bias=false
block_size=1,024
compile=true
dataset="catorze"
decay_lr=true
device="cuda"
dropout=0.2
dtype="bfloat16"
eval_interval=100
eval_iters=600
eval_only=false
grad_clip=1
gradient_accumulation_steps=32
init_from="resume"
learning_rate=0.00003
log_interval=1
lr_decay_iters=1,600
max_iters=2,000
min_lr=0.00006
n_embd=768
n_head=12
n_layer=12
out_dir="out-14"
wandb_log=true
wandb_project="owt"
wandb_run_name="gpt2-catorze-13M"
warmup_iters=100
weight_decay=0.1
```

Els detalls de cada entrenament es troben a:

<https://github.com/quermd/TFC/tree/main/config>

3. Comparativa d'entrenaments realitzats

La informació comperta de cada entrenament es troba al <https://wandb.ai/quer/owt>

Títol	Preparació de dades	Entrenament	Resultats
entrenament 01	1er dataset una sola plantilla amb variables aleatorietzades a partit de conjut de sílabes 10k xmls 33,281,603 tokens al conjunt d'entrenament	Entrenament fet per iteracions=600 gpt2-124M <hr/> 44min train loss=0.082 val loss=0.083	Tot i l'aparent correctesa del XML la validació demostra que apareixen dos tags tancats per duplicat i que per tant no és un XML correcte . Incapacitat de substituir paràmetres per les variables del prompt
entrenament 01 bis	mateix que a entrenament 01	batch_size=6 block_size=1,024 iteracions=600 gpt2-124M <hr/> 1h 20min train loss=0.067 val loss=0.067	Reproducció exacta del que se li demana. XML idèntics als de les plantilles , per tant vàlids. El mateix amb el codi de l'aplicació Es verifica que no té capacitat creativa. Incapacitat de substituir paràmetres per les variables del prompt Això es deu que el model és extremadament " overfitted ".
entrenament 02	3er dataset una sola plantilla amb descripcions i amb variables aleatorietzades a partit de conjut de sílabes 10k xmls 33,281,603 tokens al conjunt d'entrenament	iteracions=500 gpt2-124M <hr/> 50min train loss=0.159 val loss=0.163	No substitueix el text. S'observa que inventa els noms dels paràmetres. En aquest cas ha inventat la cadena CoUAnGoOU (que no es troba en el conjunt d'entrenament).
entrenament 2 Bis	mateix que a entrenament 02	meitat de capes: n_head=6 n_layer=6 max_iters=500 Learning rate més gran:	Es tornen a trobar xml molt malt formats . Amb aparença de xml i codi d'aplicació correcte. Incapacitat de substituir paràmetres per les variables del prompt

		<p>learning_rate=0.1</p> <hr/> <p>gpt2-124M</p> <hr/> <p>10min train loss=0.86 val loss=0.87</p>	
<p>intent de refinament a partir del model preentrenat de gpt</p>	<p>mateix que a entrenament 02 com a base més el model preentrenat gpt2-medium</p>	<p>batch size=1 20 iteracions</p> <hr/> <p>loss 1.0723</p>	<p>Sortida incorrecte. No té ni forma de xml. Repeticions de cadenes sense sentit S'observa que no té la capacitat del model original de reproduir cadenes de text del <i>openwebtext</i> .</p>
<p>intent de refinament a partir del model preentrenat de gpt bis</p>	<p>mateix que a entrenament 02 com a base més el model preentrenat gpt2-medium</p>	<p>batch size=200 20 iteracions</p> <hr/> <p>loss 0.0629</p>	<p>S'observa que a part de tornar el xml original de forma perfecte, quan acaba i continua mostrant altres xmls (i això ho fa en totes les respostes) s'inventa combinacions de síl·labes dels noms de l'aplicació. S'inventa noms però manté la coherència amb els noms que ha inventat dins el XML Incapacitat de substituir paràmetres per les variables del prompt</p> <p>El model ha perdut la seva capacitat de GPT-2 al estar tant sobreajustat amb el nostre dataset</p>
<p>entrenament 03</p>	<p>3 plantilles amb variables aleatoritzades a partir de conjut de sílabes</p> <p>20k de cada plantilla total 60k xmls</p> <p>232,851,034 tokens al conjut d'entrenament</p>	<p>batch size=4 block size= 1024 gradient acc=32 131072 tokens/iteració iteracions=3000 gpt2-quart-175M</p> <hr/> <p>3h train loss = 0.0506 val loss= 0.0504</p>	<p>XML=ben organitzat, formalment vàlid APP=aparentment correcte Substitució de Variables= és homogènia segons prompts. Si es sollicita AAAAAA per a la substitució, substitueix per una altra paraula inventada a partir de les síl·labes que coneix. S'observa que si s'utilitza una paraula inventada a partir de les síl·labes del conjut, sí que fa la substitució correctament</p>

			És normal en una mostra de 20.000 aleatoritzacions amb un conjunt de 25 síl·labes. Aquesta particularitat fa notar que cal millorar la forma d'aleatoritzar les construccions de noms al observar-se que juguen un paper important.
entrenament 04	3 plantilles amb variables aleatorietzades a partir de diccionari RandomWords	batch size=4 block size= 1024 gradient acc=32 131072 tokens/iteració iteracions=1500 gpt2-quart-105M gpt2-cinc-105M	Resultats similars en tots 2 entrenaments. En el quart es va cometre un error entrenant el dataset anterior (el que es va utilitzar a l'entrenament 03 de 60k). Tot i que els XMLs segueixen sent vàlids i la APP sembla correcta , els resultats en la substitució de variables no són correctes ja que la substitució de variables ràpidament es torna erràtica
entrenament 05	10k de cada plantilla total 30k xmls 107,656,226 tokens conjunt d'entrenament	<hr/> 1h 31min train loss = 0.5 val loss= 0.5	
entrenament 06	3 plantilles amb variables aleatorietzades a partir de diccionari RandomWords	batch size=4 block size= 1024 gradient acc=32 131072 tokens/iteració iteracions=3000 gpt2-sis-105M	S'aconsegueix una certa millora respecte el model anterior, però no s'arriba a una coherència en la substitució de variables, en poques línies perd la relació amb la variable introduïda convertint-la en una altra paraula aleatoritzada, o amb certa similitud a la paraula introduïda.
	10k de cada plantilla total 30k xmls 107,656,226 tokens conjunt d'entrenament	<hr/> 2h 59min train loss = 0.049 val loss= 0.053	
2on intent de finetuning del GPT2 amb el nostr dataset	mateix dataset que als entrenaments 5 i 6 Més com a base, el model preentrenat gpt2-medium	batch_size = 1 gradient_accumulati on_steps = 32 max_iters = 20 learning_rate = 3e-5 decay_lr = False	Els resultats obtinguts son un invàlids . Ni el xml és vàlid, ni té coherència, ni segueix les línies de la plantilla. Entra en bucles dels que no surt.
		<hr/> loss 0.5627	
entrenament 10	Reformes dataset: 4 plantilles noms: mix aleatori d'entre 2 i 15 xifres i lletres Increment de	batch size=4 block size= 1024 gpt2-deu-433M max_iters = 3200	Amb petites errades si s'empra variables més llargues. Gairebé perfectes si es limita els noms de les variables a substituir a una lletra i una xifra.

	comentaris Desendregament dels xmls per evitar conjunt de validació sobre-representat 100.000 XMLs 25k de cada plantilla train has 433M tokens val has 47M tokens	3h 12m 39s Train loss 0.1397 Val loss 0.1402	La substitució de funcionalitats es troba limitada a l'existència de plantilles. No combina ni genera noves distribucions de funcionalitats
Entrenament deu bis	Mateix que a l'entrenament 10	mateix que a entrenament 10 expte: dropout=0.2 max_iters = 3200 <hr/> 3h 19m 48s Train loss 0.132 Val loss 0.133	Resultats perfectes si es limita els noms de les variables a substituir a una lletra i una xifra. Amb petites errades si empra variables més llargues La substitució de funcionalitats es troba limitada a l'existència de plantilles. No combina ni genera noves distribucions de funcionalitats
Entrenament 11	Conjunt d'xmls super-reduïts (5 línies) amb variables d'entre 2 i 4 lletres minúscules (no números). 300k versió V3	Batch size=16 block_size = 256 <hr/> 56min	Resultats perfectes amb diferents proves si mantenim les variables entre 2 i 4 lletres minúscules no números. La limitació es troba en l'ús de paraules "prohibides" (xml, target o source). XML tant reduït que no dona lloc a substitució de funcionalitats
Entrenament 11 bis	Mateix conjunt però només 300 mostres	només 100 iteracions	Model no suficientment entrenat. Voldria trobar quin és el número mínim de mostres que necessita per treure uns resultats correctes.
12 gpt2-dotze-4 32M	Conjunt d'xmls complexos amb molta descripció a les funcionalitats versió V4 (5 línies) amb variables d'entre 2 i 4 lletres minúscules (no números). 300k	batch_size 4 block_size 1,024 max_iters 3,200 dropout 0.2 gradient_accumulati on_steps 32 <hr/> 3h 15m 37s Train loss 0.123 Val loss 0.127	Resultats perfectes amb diferents proves si mantenim les variables entre 2 i 4 lletres minúscules no números. La limitació es troba en l'ús de paraules "prohibides" (xml, target o source) La substitució de funcionalitats es troba limitada a l'existència de plantilles. No combina ni genera noves distribucions de funcionalitats
13	Conjunt d'xmls	batch_size 6	Resultats perfectes amb diferents

<p>gpt2-tretze-7M</p>	<p>anterior al que es simplifica el codi al màxim versió short-template(5 linies) amb variables d'entre 2 i 4 lletres minúscules (no números). 300k</p>	<p>block_size 1,024 max_iters 780 warmup_iters 100 gradient_accumulati on_steps 16 dropout 0.2</p> <hr/> <p>39m 33s Train loss 1.778 Val loss 1.794</p>	<p>proves si mantenim les variables entre 2 i 4 lletres minúscules no números. La limitació es troba en l'ús de paraules "prohibides" (xml, target o source) La substitució de funcionalitats es troba limitada a l'existència de plantilles. No combina ni genera noves distribucions de funcionalitats</p>
<p>14 gpt2-catorze-13M</p>	<p>Conjunt d'xmles super-reduïts amb funcionalitat diverses dins el codi. versió V6 (5 linies) amb variables d'entre 2 i 4 lletres minúscules (no números). 300k S'han emprat caràcters de final de fitxer</p>	<p>block_size 1,024 batch_size 6 max_iters 2,000 dropout 0.2</p> <hr/> <p>1h 31m 48s Train loss 0.333 Val loss 0.321</p>	<p>XMLs ben formatats Substitució de variables imperfecte genera alguns errors 50% Substitució de funcionalitats perfecte</p>

4. Exemples dels conjunts d'entrenament

Entrenament 11:

```

<!-- Mover de ncsb a rub-->
<xml>
  <source>ncsb</source>
  <target>rub</target>
</xml>
<!-- Mover de mlg a rm-->
<xml>
  <source>mlg</source>
  <target>rm</target>
</xml>
<!-- Mover de bpvm a yc-->
<xml>
  <source>bpvm</source>
  <target>yc</target>
</xml>

```

Entrenament 12 (s'eliminen fragments):

```
mover_archivos_simple_application_mUWvigwbcSmKz_OIDZ.xml
<?xml version="1.0" encoding="UTF-8"?>
<blueprint xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0"
  xmlns:cm="http://aries.apache.org/blueprint/xmlns/blueprint-cm/v1.1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <!-- mover_archivos_simple_application_mUWvigwbcSmKz_OIDZ -->
  <!-- nombre corto de aplicación: mUWvigwbcSmKz-OIDZ-connect -->
  <!-- paquete de aplicación: net.company.mUWvigwbcSmKz.OIDZ.connect
-->
  <!-- Esta aplicación mover archivos simple application mUWvigwbcSmKz
OIDZ sirve para mover archivos de mUWvigwbcSmKz a OIDZ y obtener una
respuesta de OIDZ a mUWvigwbcSmKz -->

  <cm:property-placeholder id="configurationProperties"
    persistent-id="net.company.mUWvigwbcSmKz.OIDZ.connect"
    update-strategy="reload"/>

  <!--camelContext de mover archivos simple de mUWvigwbcSmKz a OIDZ
-->
  <camelContext id="mUWvigwbcSmKz-OIDZ-connect-camelContext"
    xmlns="http://camel.apache.org/schema/blueprint">
    <propertyPlaceholder id="properties"
    location="blueprint:configurationProperties"/>
    (...)

    <!--ruta para mover archivos de mUWvigwbcSmKz a OIDZ -->
    <!-- OIDZ a mUWvigwbcSmKz (PREGUNTA) -->
    <route id="mUWvigwbcSmKz-OIDZ-route">

      <!--uri mover archivos de mUWvigwbcSmKz a OIDZ -->
      <from
uri="master:mUWvigwbcSmKz-OIDZ-route:{{input.mUWvigwbcSmKz.OIDZ}}"/
>
        <log loggingLevel="INFO"
message="[mUWvigwbcSmKz-OIDZ-connect] Recibido fichero ${file:name}"/>
        <log loggingLevel="INFO"
message="[mUWvigwbcSmKz-OIDZ-connect] Contenido del fichero: ${body}
"/>

        <to
uri="master:mUWvigwbcSmKz-OIDZ-route:{{output.mUWvigwbcSmKz.OIDZ}}
" />
        (...)
        <setHeader headerName="routeMailSubject">
        <simple>mUWvigwbcSmKz-OIDZ-connect (mUWvigwbcSmKz a
OIDZ)</simple>
```

```

        </setHeader>
        <setHeader headerName="description">
        <simple>Error procesando fichero ${file:name} de
mUWvigwbcSmKz a OIDZ </simple>
        </setHeader>
        <setHeader headerName="errorException">
        <simple>${exception.stacktrace}</simple>
        </setHeader>
        </onException>
    </route>

    <!--ruta para mover archivos de OIDZ a mUWvigwbcSmKz -->
    <!-- OIDZ a mUWvigwbcSmKz (RESPUESTA) -->
    <route id="OIDZ-mUWvigwbcSmKz-route">
        <!--from uri input mover archivos de OIDZ a mUWvigwbcSmKz
-->
        <from
uri="master:OIDZ-mUWvigwbcSmKz-route:{{input.OIDZ.mUWvigwbcSmKz}}"/
>
            <log loggingLevel="INFO"
message="[mUWvigwbcSmKz-OIDZ-connect] Recibido fichero ${file:name}
de OIDZ"/>
            <log loggingLevel="INFO"
message="[mUWvigwbcSmKz-OIDZ-connect] Contenido del fichero:
${body}"/>
            (...)

            <!--to uri output mover archivos de OIDZ a mUWvigwbcSmKz
-->
            <to
uri="master:OIDZ-mUWvigwbcSmKz-route:{{output.OIDZ.mUWvigwbcSmKz}}
" />
            (...)
            <setHeader headerName="routeMailSubject">
            <simple>mUWvigwbcSmKz-OIDZ-connect (OIDZ a
mUWvigwbcSmKz)</simple>
            </setHeader>
            <setHeader headerName="description">
            <simple>Error procesando fichero ${file:name} de OIDZ a
mUWvigwbcSmKz </simple>
            </setHeader>
            <setHeader headerName="errorException">
            <simple>${exception.stacktrace}</simple>
            </setHeader>
            </onException>
        </route>
    </camelContext>
</blueprint>
mover_archivos_doble_application_Hj7_4wPa3.xml

```

```
<?xml version="1.0" encoding="UTF-8"?>
<blueprint xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0"
  xmlns:cm="http://aries.apache.org/blueprint/xmlns/blueprint-cm/v1.1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <!-- mover_archivos_doble_application_Hj7_4wPa3 -->
  <!-- nombre corto de aplicación: Hj7-4wPa3-connect -->
  <!-- paquete de aplicación: net.company.doble.Hj7.4wPa3.connect -->
  <!-- Esta aplicación mover archivos doble application Hj7 4wPa3 sirve para
  mover de Hj7B1 a 4wPa3B1IN y obtener una respuesta de 4wPa3B1OUT a
  Hj7B1IN
  y además para mover de Hj7B2 a 4wPa3B2 y obtener una respuesta
  de 4wPa3B2OUT a Hj7B2IN -->

  <cm:property-placeholder id="configurationProperties"
    persistent-id="net.company.doble.Hj7.4wPa3.connect"
    update-strategy="reload"/>

  <!--camelContext de mover archivos doble de Hj7 a 4wPa3 -->
  <camelContext id="Hj7-4wPa3-connect-camelContext"
    xmlns="http://camel.apache.org/schema/blueprint">
```

Entrenament 13:

```
mover_archivos_doble_application_KNDMAQ_YPEVEPQH.xml
<!-- Mover doble de KNDMAQ a YPEVEPQH-->
<xml>
  <!-- Ida de KNDMAQ a YPEVEPQH-->
  <source>KNDMAQ</source>
  <target>YPEVEPQH</target>
  <!-- Vuelta de YPEVEPQH a KNDMAQ-->
  <source>YPEVEPQH</source>
  <target>KNDMAQ</target>
</xml>
mover_archivos_simple_application_LUUORTJ_RQMBMG.xml
<!-- Mover simple de LUUORTJ a RQMBMG-->
<xml>
  <source>LUUORTJ</source>
  <target>RQMBMG</target>
</xml>
mover_archivos_triple_application_JFMRUG_QMSQMU.xml
<!-- Mover triple de JFMRUGA a QMSQMUA y de JFMRUGB a
QMSQMUB-->
<xml>
  <!-- Ida de JFMRUGA a QMSQMUA-->
  <source>JFMRUGA</source>
```

```
<target>QMSQMUA</target>
<!-- Vuelta de QMSQMUA a JFMRUGA-->
<source>QMSQMUA</source>
<target>JFMRUGA</target>
<!-- Ida de JFMRUGB a QMSQMUB-->
<source>JFMRUGB</source>
<target>QMSQMUB</target>
<!-- Vuelta de QMSQMUB a JFMRUGB-->
<source>QMSQMUB</source>
<target>JFMRUGB</target>
</xml>
mover_archivos_simple_application_BFUFBVJL_XBWXRO.xml
<!-- Mover simple de BFUFBVJL a XBWXRO-->
<xml>
  <source>BFUFBVJL</source>
  <target>XBWXRO</target>
</xml>
```

Entrenament 14:

```
mover_archivos_simple_fun1_log_application_WWYYIH_HCPMLGSO.xml
<!-- Mover simple de WWYYIH a HCPMLGSO-->
<!-- Con funcionalidad particular1-->
<!-- Con log-->
<xml>
  <!-- Funcionalidad particular1-->
  <funcionalidad particular1="WWYYIHHCPMLGSO">
    <source>WWYYIH</source>
    <target>HCPMLGSO</target>
    <!-- log-->
    <log message="[WWYYIH-HCPMLGSO-connect] Sent file to
HCPMLGSO"/>
  </xml>
<|endoftext|>
mover_archivos_simple_fun1_application_FGZOM_ZUEDRNZGO.xml
<!-- Mover simple de FGZOM a ZUEDRNZGO-->
<!-- Con funcionalidad particular1-->
<xml>
  <!-- Funcionalidad particular1-->
  <funcionalidad particular1="FGZOMZUEDRNZGO">
    <source>FGZOM</source>
    <target>ZUEDRNZGO</target>
  </xml>
<|endoftext|>
mover_archivos_simple_fun2_application_POBJKTG_EQHA.xml
<!-- Mover simple de POBJKTG a EQHA-->
```



```
<!-- Con funcionalidad particular2-->
<xml>
  <!-- Funcionalidad particular2-->
  <funcionalidad particular2="EQHA">
    <source>POBJKTG</source>
    <target>EQHA</target>
  </xml>
<|endoftext|>
mover_archivos_doble_log_application_JSLJFI_RBNJOI.xml
<!-- Mover doble de JSLJFI a RBNJOI-->
<!-- Con log-->
<xml>
  <!-- Ida de JSLJFI a RBNJOI-->
  <source>JSLJFI</source>
  <target>RBNJOI</target>
  <!-- Vuelta de RBNJOI a JSLJFI-->
  <source>RBNJOI</source>
  <target>JSLJFI</target>
  <!-- log-->
  <log message="[JSLJFI-RBNJOI-connect] Sent file to RBNJOI and obtained
file "/>
</xml>
<|endoftext|>
mover_archivos_triple_fun1_log_application_OHYCDCJW_WLFDXF.xml
<!-- Mover triple de OHYCDCJWA a WLFDXFA y de OHYCDCJWB a
WLFDXFB-->
<!-- Con funcionalidad particular1-->
<!-- Con log-->
<xml>
  <!-- Funcionalidad particular1-->
  <funcionalidad particular1="OHYCDCJWWLFDXF">
    <!-- Ida de OHYCDCJWA a WLFDXFA-->
    <source>OHYCDCJWA</source>
    <target>WLFDXFA</target>
    <!-- Vuelta de WLFDXFA a OHYCDCJWA-->
    <source>WLFDXFA</source>
    <target>OHYCDCJWA</target>
    <!-- Ida de OHYCDCJWB a WLFDXFB-->
    <source>OHYCDCJWB</source>
    <target>WLFDXFB</target>
    <!-- Vuelta de WLFDXFB a OHYCDCJWB-->
    <source>WLFDXFB</source>
    <target>OHYCDCJWB</target>
    <!-- log-->
    <log message="[OHYCDCJW-WLFDXF-connect] Sent file to WLFDXFA and
obtained file and Sent file to WLFDXFB and obtained file "/>
  </xml>
mover_archivos_doble_log_application_FLZRNIB_UWPLMH.xml
<!-- Mover doble de FLZRNIB a UWPLMH-->
```

```
<!-- Con log-->
<xml>
  <!-- Ida de FLZRNIB a UWPLMH-->
  <source>FLZRNIB</source>
  <target>UWPLMH</target>
  <!-- Vuelta de UWPLMH a FLZRNIB-->
  <source>UWPLMH</source>
  <target>FLZRNIB</target>
  <!-- log-->
  <log message="[FLZRNIB-UWPLMH-connect] Sent file to UWPLMH and
obtained file "/>
</xml>
<|endoftext|>
mover_archivos_simple_log_application_VGSO_YMFXB.xml
<!-- Mover simple de VGSO a YMFXB-->
<!-- Con log-->
<xml>
  <source>VGSO</source>
  <target>YMFXB</target>
  <!-- log-->
  <log message="[VGSO-YMFXB-connect] Sent file to YMFXB"/>
</xml>
<|endoftext|>
```

5. Detall plantilles conjunt entrenament 14

short_template_pro.xml

```
<!-- Mover simple de ${ORIGEN} a ${DESTINO}-->
<!-- Con funcionalidad particular1-->
<!-- Con log-->
<xml>
  <!-- Funcionalidad particular1-->
  <funcionalidad particular1="${ORIGEN}${DESTINO}">
  <source>${ORIGEN}</source>
  <target>${DESTINO}</target>
  <!-- log-->
  <log message="[${ORIGEN}-${DESTINO}-connect] Sent file to
${DESTINO}"/>
</xml>
<|endoftext|>
```

short_template_pro1.xml

```
<!-- Mover simple de ${ORIGEN} a ${DESTINO}-->
<!-- Con log-->
<xml>
```

```
<source>${ORIGEN}</source>
<target>${DESTINO}</target>
<!-- log-->
<log message="[${ORIGEN}-${DESTINO}-connect] Sent file to
${DESTINO}"/>
</xml>
<|endoftext|>
```

short_template_pro2.xml

```
<!-- Mover simple de ${ORIGEN} a ${DESTINO}-->
<!-- Con funcionalidad particular1-->
<xml>
  <!-- Funcionalidad particular1-->
  <funcionalidad particular1="${ORIGEN}${DESTINO}">
  <source>${ORIGEN}</source>
  <target>${DESTINO}</target>
</xml>
<|endoftext|>
```

short_template_pro3.xml

```
<!-- Mover simple de ${ORIGEN} a ${DESTINO}-->
<!-- Con funcionalidad particular2-->
<xml>
  <!-- Funcionalidad particular2-->
  <funcionalidad particular2="${DESTINO}">
  <source>${ORIGEN}</source>
  <target>${DESTINO}</target>
</xml>
<|endoftext|>
```

short_template_doble_pro1.xml

```
<!-- Mover doble de ${ORIGEN} a ${DESTINO}-->
<!-- Con log-->
<xml>
  <!-- Ida de ${ORIGEN} a ${DESTINO}-->
  <source>${ORIGEN}</source>
  <target>${DESTINO}</target>
  <!-- Vuelta de ${DESTINO} a ${ORIGEN}-->
  <source>${DESTINO}</source>
  <target>${ORIGEN}</target>
  <!-- log-->
  <log message="[${ORIGEN}-${DESTINO}-connect] Sent file to ${DESTINO}
and obtained file "/>
</xml>
<|endoftext|>
```

short_template_doble_pro2.xml

```
<!-- Mover doble de ${ORIGEN} a ${DESTINO}-->
<!-- Con funcionalidad particular1-->
<xml>
  <!-- Funcionalidad particular1-->
  <funcionalidad particular1="${ORIGEN}${DESTINO}">
    <!-- Ida de ${ORIGEN} a ${DESTINO}-->
    <source>${ORIGEN}</source>
    <target>${DESTINO}</target>
    <!-- Vuelta de ${DESTINO} a ${ORIGEN}-->
    <source>${DESTINO}</source>
    <target>${ORIGEN}</target>
  </xml>
</endoftext>
```

short_template3_pro3.xml

```
<!-- Mover triple de ${ORIGEN}A a ${DESTINO}A y de ${ORIGEN}B a
${DESTINO}B-->
<!-- Con funcionalidad particular1-->
<!-- Con log-->
<xml>
  <!-- Funcionalidad particular1-->
  <funcionalidad particular1="${ORIGEN}${DESTINO}">
    <!-- Ida de ${ORIGEN}A a ${DESTINO}A-->
    <source>${ORIGEN}A</source>
    <target>${DESTINO}A</target>
    <!-- Vuelta de ${DESTINO}A a ${ORIGEN}A-->
    <source>${DESTINO}A</source>
    <target>${ORIGEN}A</target>
    <!-- Ida de ${ORIGEN}B a ${DESTINO}B-->
    <source>${ORIGEN}B</source>
    <target>${DESTINO}B</target>
    <!-- Vuelta de ${DESTINO}B a ${ORIGEN}B-->
    <source>${DESTINO}B</source>
    <target>${ORIGEN}B</target>
    <!-- log-->
    <log message="[${ORIGEN}-${DESTINO}-connect] Sent file to
${DESTINO}A and obtained file and Sent file to ${DESTINO}B and obtained
file "/>
  </xml>
```

6. Detail Entrenament 10 bis

Consulta
mover_archivos_doble_Z8_X9.xml
Resposta
<pre> mover_archivos_doble_Z8_X9.xml <?xml version="1.0" encoding="UTF-8"?> <blueprint xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0" xmlns:cm="http://aries.apache.org/blueprint/xmlns/blueprint-cm/v1.1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <!-- mover_archivos_doble_application_Z8_X9 --> <!-- nombre corto de aplicación: Z8-X9-connect --> <!-- paquete de aplicación: net.company.doble.Z8.X9.connect --> <!-- Esta aplicación mover archivos doble application Z8 sirve para mover de Z8B1 a X9B1IN y obtener una respuesta de X9B1OUT a Z8B1IN y además para mover de Z8B2 a X9B2 y obtener una respuesta de X9B2OUT a Z8B2IN --> <cm:property-placeholder id="configurationProperties" persistent-id="net.company.doble.Z8.X9.connect" update-strategy="reload"/> <!--camelContext de mover archivos doble de Z8 a X9 --> <camelContext id="Z8-X9-connect-camelContext" xmlns="http://camel.apache.org/schema/blueprint"> <propertyPlaceholder id="properties" location="blueprint:configurationProperties"/> <redeliveryPolicyProfile allowRedeliveryWhileStopping="false" id="Redeliveries" logRetryStackTrace="true" maximumRedeliveries="{{local.maximumRedeliveries}}" redeliveryDelay="{{local.redeliveryDelay}}" retryAttemptedLogLevel="WARN"/> <!--ruta para mover archivos de Z8B1 a X9B1IN --> <!-- Route1: Z8B1 to X9B1IN --> <route id="Z8-X9-connect-fuse-routeX9B1IN"> <from uri="master:Z8-X9-connect-fuse-routeX9B1IN:{{input.sftpRoute.X9B1IN}}"/> <log message="[Z8-X9-connect] Received file from SFTP /interface: \${file:name}"/> <setHeader headerName="bodySize"> <constant>1</constant> </setHeader> <setHeader headerName="description"> <simple>Process end successfully for file</simple> </setHeader> <to uri="{{output.fileFolder.X9B1IN}}"/> <log message="[Z8-X9-connect] Sent file \${file:name} to remote folder Interface route {{output.fileFolder.X9B1IN}}"/> <onException redeliveryPolicyRef="Redeliveries"> </pre>

```

    <exception>java.lang.Exception</exception>
    <log loggingLevel="ERROR" message="[Z8-X9-connect] [routeX9B1IN]
ERROR: ${exception.stacktrace}"/>
    <setHeader headerName="routeMailSubject">
        <simple>Z8B1-X9B1IN</simple>
    </setHeader>
    <setHeader headerName="description">
        <simple>Error processing file ${file:name} from Z8B1 to X9B1IN
</simple>
    </setHeader>
    <setHeader headerName="errorException">
        <simple>${exception.stacktrace}</simple>
    </setHeader>
</onException>
</route>

    <!--ruta para mover archivos de Z8B2 a X9B2IN -->
    <!-- Route3: Z8B2 to X9B2IN -->
    <route id="Z8-X9-connect-fuse-routeX9B2IN">
        <from
uri="master:Z8-X9-connect-fuse-routeX9B2IN:{{input.sftpRoute.X9B2IN}}"/>
        <log message="[Z8-X9-connect] Received file from SFTP /interface:
${file:name}"/>
        <setHeader headerName="bodySize">
            <constant>1</constant>
        </setHeader>
        <setHeader headerName="description">
            <simple>Process end successfully for file</simple>
        </setHeader>
        <to uri="{{output.fileFolder.X9B2IN}}"/>
        <log message="[Z8-X9-connect] Sent file ${file:name} to remote
folder Interface route {{output.fileFolder.X9B2IN}}"/>
        <onException redeliveryPolicyRef="Redeliveries">
            <exception>java.lang.Exception</exception>
            <log loggingLevel="ERROR" message="[Z8-X9-connect]
[routeX9B2IN] ERROR: ${exception.stacktrace}"/>
            <setHeader headerName="routeMailSubject">
                <simple>Z8B2-X9B2IN</simple>
            </setHeader>
            <setHeader headerName="description">
                <simple>Error processing file ${file:name} from Z8B2 to X9B2IN
</simple>
        </setHeader>
        <setHeader headerName="errorException">
            <simple>${exception.stacktrace}</simple>
        </setHeader>
</onException>
</route>

    <!--ruta para mover archivos de Z8B3 a X9B3IN -->
    <!-- Route6: Z8B3 to X9B3IN -->
    <route id="Z8-X9-connect-fuse-routeX9B3IN">
        <from
uri="master:Z8-X9-connect-fuse-routeX9B3IN:{{input.sftpRoute.X9B3IN}}"/>

```

```

        <log message="[Z8-X9-connect] Received file from SFTP /interface:
        ${file:name}"/>
        <setHeader headerName="bodySize">
            <constant>1</constant>
        </setHeader>
        <setHeader headerName="description">
            <simple>Process end successfully for file</simple>
        </setHeader>
        <to uri="{{output.fileFolder.X9B3IN}}"/>
        <log message="[Z8-X9-connect] Sent file ${file:name} to remote
        folder Interface route {{output.fileFolder.X9B3IN}}"/>
        <onException redeliveryPolicyRef="Redeliveries">
            <exception>java.lang.Exception</exception>
            <log loggingLevel="ERROR" message="[Z8-X9-connect]
            [routeX9B3IN] ERROR: ${exception.stacktrace}"/>
            <setHeader headerName="routeMailSubject">
                <simple>Z8B3-X9B3IN</simple>
            </setHeader>
            <setHeader headerName="description">
                <simple>Error processing file ${file:name} from Z8B3 to X9B3IN
            </simple>
            </setHeader>
            <setHeader headerName="errorException">
                <simple>${exception.stacktrace}</simple>
            </setHeader>
        </onException>
        </route>

        <!--ruta para mover archivos respuesta de X9B1OUT a Z8B1IN -->
        <!-- Route7: X9B1OUT to Z
    
```

Consulta
mover_archivos_triple_X8_Z9.xml
Resposta
<pre> mover_archivos_triple_X8_Z9.xml <?xml version="1.0" encoding="UTF-8"?> <blueprint xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0" xmlns:cm="http://aries.apache.org/blueprint/xmlns/blueprint-cm/v1.1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <!-- mover_archivos_triple_application_X8_Z9 --> <!-- nombre corto de aplicación: X8-Z9-connect --> <!-- paquete de aplicación: net.company.triple.X8.Z9.connect --> <!-- Esta aplicación mover archivos triple application X8 k9 sirve para mover de X8B1 a k9B1 y obtener una respuesta de k9B1OUT a X8B1IN y además para mover de X8B2 a k9B2 y obtener una respuesta de k9B2OUT a X8B2IN y además para mover de X8B3 a k9B3 y obtener una respuesta de k9B3OUT a X8B3IN --> </pre>

```

<cm:property-placeholder id="configurationProperties"
  persistent-id="net.company.triple.X8.Z9.connect" update-strategy="reload"/>

<!--camelContext de mover archivos triple de X8 a k9 -->
<camelContext id="X8-Z9-connect-camelContext"
xmlns="http://camel.apache.org/schema/blueprint">
  <propertyPlaceholder id="properties"
location="blueprint:configurationProperties"/>
  <redeliveryPolicyProfile allowRedeliveryWhileStopping="false"
id="Redeliveries" logRetryStackTrace="true"
maximumRedeliveries="{{local.maximumRedeliveries}}"
redeliveryDelay="{{local.redeliveryDelay}}"
retryAttemptedLogLevel="WARN"/>

  <!--ruta para mover archivos de X8B1 a k9B1IN -->
  <!-- Route4: X8B1 to k9B1IN -->
  <route id="X8-Z9-connect-fuse-routek9B1IN">
    <from
uri="master:X8-Z9-connect-fuse-routek9B1IN:{{input.sftpRoute.Z9B1IN}}"/>
    <log message="[X8-Z9-connect] Received file from SFTP /interface:
${file:name}"/>
    <setHeader headerName="bodySize">
      <constant>1</constant>
    </setHeader>
    <setHeader headerName="description">
      <simple>Process end successfully for file</simple>
    </setHeader>
    <to uri="{{output.fileFolder.Z9B1IN}}"/>
    <log message="[X8-Z9-connect] Sent file ${file:name} to remote
folder Interface route {{output.fileFolder.Z9B1IN}}"/>
    <onException redeliveryPolicyRef="Redeliveries">
      <exception>java.lang.Exception</exception>
      <log loggingLevel="ERROR" message="[X8-Z9-connect] [routek9B1IN]
ERROR: ${exception.stacktrace}"/>
      <setHeader headerName="routeMailSubject">
        <simple>X8B1-Z9B1IN</simple>
      </setHeader>
      <setHeader headerName="description">
        <simple>Error processing file ${file:name} from X8B1 to k9B1IN
</simple>
      </setHeader>
      <setHeader headerName="errorException">
        <simple>${exception.stacktrace}</simple>
      </setHeader>
    </onException>
  </route>

  <!--ruta para mover archivos respuesta de k9B2OUT a X8B2IN -->
  <!-- Route8: k9B2OUT to X8B2IN (Vueltak9B2OUT) -->
  <route id="X8-Z9-connect-Z9B2OUTX8B2IN-route2">
    <from
uri="master:X8-Z9-connect-Z9B2OUTX8B2IN-route2:{{input.route.Z9B2OUT}}"/>
    <log loggingLevel="INFO"

```



```
message="[X8-Z9-connect][Z9B2OUTX8B2IN-route2] Fichero ${file:name}.
Moviendo a X8B2IN..."/>
  <to uri="{{output.sftpRoute.Z9B2OUT}}"/>
  <log loggingLevel="INFO"
message="[X8-Z9-connect][Z9B2OUTX8B2IN-route2] Fichero ${file:name}. Movido a
X8B2IN."/>
  <onException redeliveryPolicyRef="Redeliveries">
  <exception>java.lang.Exception</exception>
  <log loggingLevel="ERROR" message="[X8-Z9-connect]
[Z9B2OUTX8B2-route2] ERROR: ${exception.stacktrace}"/>
  <setHeader headerName="routeMailSubject">
    <simple>Z9B2OUT-X8B2IN</simple>
  </setHeader>
  <setHeader headerName="description">
    <simple>Error processing file ${file:name} from k9B2OUT to X8B2IN
</simple>
  </setHeader>
  <setHeader headerName="errorException">
    <simple>${exception.stacktrace}</simple>
  </setHeader>
</onException>
</route>

<!--ruta para mover archivos respuesta de k9B3OUT a X8B3IN -->
<!-- Route9: k9B3OUT to k9B3IN (Vueltak9B3OUT) -->
<route id="X8-Z9-connect-Z9B3OUTX8B3IN-route3">
  <from
uri="master:X8-Z9-connect-Z9B3OUTX8B3IN-route3:{{input.route.Z9B3OUT}}"/>
  <log loggingLevel="INFO"
message="[X8-Z9-connect][Z9B3OUTX8B3IN-route3] Fichero ${file:name}.
Moviendo a X8B3IN..."/>
  <to uri="{{output.sftpRoute.Z9B3OUT}}"/>
  <log loggingLevel="INFO"
message="[X8-Z9-connect][Z9B3OUTX8B3IN-route3] Fichero ${file:name}. Movido a
X8B3IN."/>
  <onException redeliveryPolicyRef="Redeliveries">
  <exception>java.lang.Exception</exception>
  <log loggingLevel="ERROR" message="[X8-Z9-connect]
[Z9B3OUTX8B3IN-route3] ERROR: ${exception.stacktrace}"/>
  <setHeader headerName="routeMailSubject">
    <simple>Z9B3OUT-X8B3IN</simple>
  </setHeader>
  <setHeader headerName="description">
    <simple>Error processing file ${file:name} from k9B3OUT to X8B3IN
</simple>
  </setHeader>
  <setHeader headerName="errorException">
    <simple>${exception.stacktrace}</simple>
```

7. Proves contra ChatGPT d'OpenAI

Consulta

a partir de estos 3 xmls:

mover_simple_de_ORIGEN_a_DESTINO.xml

```
<!-- Mover simple de ${ORIGEN} a ${DESTINO}-->
```

```
<xml>
```

```
<source>${ORIGEN}</source>
```

```
<target>${DESTINO}</target>
```

```
</xml>
```

mover_doble_de_ORIGEN_a_DESTINO.xml

```
<!-- Mover doble de ${ORIGEN} a ${DESTINO}-->
```

```
<xml>
```

```
<!-- Ida de ${ORIGEN} a ${DESTINO}-->
```

```
<source>${ORIGEN}</source>
```

```
<target>${DESTINO}</target>
```

```
<!-- Vuelta de ${DESTINO} a ${ORIGEN}-->
```

```
<source>${DESTINO}</source>
```

```
<target>${ORIGEN}</target>
```

```
</xml> mover_triple_de_ORIGEN_a_DESTINO.xml <!-- Mover triple de ${ORIGEN}A
```

```
a ${DESTINO}A y de ${ORIGEN}B a ${DESTINO}B--> <xml> <!-- Ida de
```

```
${ORIGEN}A a ${DESTINO}A--> <source>${ORIGEN}A</source>
```

```
<target>${DESTINO}A</target> <!-- Vuelta de ${DESTINO}A a ${ORIGEN}A-->
```

```
<source>${DESTINO}A</source> <target>${ORIGEN}A</target> <!-- Ida de
```

```
${ORIGEN}B a ${DESTINO}B--> <source>${ORIGEN}B</source>
```

```
<target>${DESTINO}B</target> <!-- Vuelta de ${DESTINO}B a ${ORIGEN}B-->
```

```
<source>${DESTINO}B</source> <target>${ORIGEN}B</target> </xml>
```

¿puedes generar un xml nuevo tipo mover_doble_de_ORIGEN_a_DESTINO.xml pero substituyendo las palabras ORIGEN por ABCDF y DESTINO por WTUVX?

Resposta

¡Claro! Aquí tienes el XML generado, llamado

"mover_doble_de_ABCDF_a_WTUVX.xml":

```
```xml
```

```
<!-- Mover doble de ABCDF a WTUVX-->
```

```
<xml>
```

```
<!-- Ida de ABCDF a WTUVX-->
```

```
<source>ABCDF</source>
```

```
<target>WTUVX</target>
```

```
<!-- Vuelta de WTUVX a ABCDF-->
```

```
<source>WTUVX</source>
```

```
<target>ABCDF</target>
```

```
</xml>
```

...

El XML nuevo es una versión modificada de "mover\_doble\_de\_ORIGEN\_a\_DESTINO.xml", donde se han reemplazado todas las instancias de "ORIGEN" por "ABCDF" y todas las instancias de "DESTINO" por "WTUVX". Espero que esto sea lo que estabas buscando. Si tienes alguna otra pregunta, ¡no dudes en preguntar!

### Reconsulta

¿puedes hacer lo mismo para el de tipo triple?

### Resposta

¡Por supuesto! Aquí tienes el XML generado, llamado "mover\_triple\_de\_ABCDF\_a\_WTUVX.xml":

```
```xml
<!-- Mover triple de ABCDFA a WTUVXA y de ABCDFB a WTUVXB-->
<xml>
  <!-- Ida de ABCDFA a WTUVXA-->
  <source>ABCDFA</source>
  <target>WTUVXA</target>
  <!-- Vuelta de WTUVXA a ABCDFA-->
  <source>WTUVXA</source>
  <target>ABCDFA</target>
  <!-- Ida de ABCDFB a WTUVXB-->
  <source>ABCDFB</source>
  <target>WTUVXB</target>
  <!-- Vuelta de WTUVXB a ABCDFB-->
  <source>WTUVXB</source>
  <target>ABCDFB</target>
</xml>
```
```

El XML nuevo es una versión modificada de "mover\_triple\_de\_ORIGEN\_a\_DESTINO.xml", donde se han reemplazado todas las instancias de "ORIGEN" por "ABCDF" y todas las instancias de "DESTINO" por "WTUVX". Espero que esto sea lo que estabas buscando. Si tienes alguna otra pregunta, ¡no dudes en preguntar!

### Consulta

Donades les tres aplicacions següents: <!--"Descripció aplicació 1 amb funcionalitat 1 (paràmetre A), funcionalitat 2 (paràmetres B i C) i funcionalitat 3" --> <Aplicació 1> <Funcionalitat 1 "paràmetre A"> <Funcionalitat 2 "paràmetres B" "paràmetre C"> <Funcionalitat 3> <!--"Descripció aplicació 2

amb funcionalitat 1 (paràmetre A) i funcionalitat 3"--> <Aplicació 2>  
<Funcionalitat 1 "paràmetre A"> <Funcionalitat 3> <!--"Descripció aplicació 3  
amb funcionalitat 4"--> <Aplicació 3> <Funcionalitat 4> Podries generar una  
nova aplicació a partir de la següent descripció? "Descripció aplicació 6 amb  
funcionalitat 2 (paràmetres B i C) i funcionalitat 4"

ChatGPT

Sí, basant-me en la descripció proporcionada, puc generar una nova aplicació amb les funcionalitats 2 i 4. Anomenarem aquesta nova aplicació "Aplicació 6". A continuació, t'ofereixo una descripció de l'Aplicació 6:

<Aplicació 6>

- Funcionalitat 2: Aquesta funcionalitat permet als usuaris utilitzar els paràmetres B i C per dur a terme una acció específica. Els usuaris poden personalitzar els valors dels paràmetres B i C segons les seves necessitats i l'aplicació respondrà en conseqüència.
- Funcionalitat 4: Aquesta funcionalitat ofereix una característica única a l'Aplicació 6. Permet als usuaris realitzar una tasca o acció específica que no està relacionada amb els paràmetres anteriors. Aquesta funcionalitat pot variar en funció del propòsit de l'aplicació i pot ser personalitzada per satisfer les necessitats dels usuaris.

L'Aplicació 6 combina les funcionalitats 2 i 4 per proporcionar als usuaris una experiència única i versàtil. Pots personalitzar els paràmetres B i C per a la funcionalitat 2, i també utilitzar la funcionalitat 4 per realitzar tasques addicionals dins de l'aplicació.

Reconsulta

No és correcte, la resposta hauria de seguir una estructura com la següent:  
<!--"Descripció aplicació 4 amb funcionalitat 1 (paràmetre A), funcionalitat 2  
(paràmetres B i C) i funcionalitat 3"--> <Aplicació 4> <Funcionalitat 1  
"paràmetre A"> <Funcionalitat 2 "paràmetre B" "paràmetre C"> <Funcionalitat  
3>

ChatGPT

Em disculpo per la confusió. A partir de la descripció proporcionada, generarem una nova aplicació anomenada "Aplicació 6" amb les funcionalitats 2 (paràmetres B i C) i 4. A continuació tens la descripció de l'Aplicació 6:

<Aplicació 6>

- Funcionalitat 2: Aquesta funcionalitat permet als usuaris utilitzar els paràmetres B i C per dur a terme una acció específica. Els usuaris poden personalitzar els valors dels paràmetres B i C segons les seves necessitats i l'aplicació respondrà en conseqüència.
- Funcionalitat 4: Aquesta funcionalitat ofereix una característica única a l'Aplicació 6. Permet als usuaris realitzar una tasca o acció específica que és independent dels paràmetres anteriors. Aquesta funcionalitat

pot variar en funció del propòsit de l'aplicació i pot ser personalitzada per satisfer les necessitats dels usuaris.

L'Aplicació 6 combina les funcionalitats 2 i 4 per proporcionar als usuaris una experiència versàtil. Els usuaris poden utilitzar els paràmetres B i C per controlar l'acció realitzada per la funcionalitat 2, i també poden aprofitar la funcionalitat 4 per dur a terme altres tasques o accions dintre de l'aplicació. Espero que aquesta nova descripció s'ajusti a les teves expectatives.

#### Reconsulta

No siguis creatiu, ceneix-te a respondre amb l'estructura bàsica de l'exemple ChatGPT

Disculpa per la confusió anterior. A partir de la descripció proporcionada, generem una nova aplicació anomenada "Aplicació 6" amb les funcionalitats 2 (paràmetres B i C) i 4. A continuació tens la descripció de l'Aplicació 6:

<!--"Descripció aplicació 6 amb funcionalitat 2 (paràmetres B i C) i funcionalitat 4"-->

<Aplicació 6>

<Funcionalitat 2 "paràmetres B i C">

<Funcionalitat 4>

L'Aplicació 6 té una funcionalitat 2 que permet als usuaris utilitzar els paràmetres B i C per dur a terme una acció específica. Els usuaris poden personalitzar els valors d'aquests paràmetres segons les seves necessitats. A més, l'aplicació també ofereix una funcionalitat 4 que permet als usuaris realitzar una tasca o acció específica diferent de les altres funcionalitats mencionades.

Espero que aquesta resposta s'ajusti a l'estructura que estaves buscant. Si tens alguna altra pregunta, estic aquí per ajudar-te.

#### Reconsulta

Perfecte, ara repetim amb l'entrada: "Descripció aplicació 7 amb funcionalitat 1 (paràmetre PISCINA), amb funcionalitat 2 (paràmetres CASA i FINESTRA), funcionalitat 3 i funcionalitat 4"

#### ChatGPT

Entesos. Basant-me en la nova descripció proporcionada, generaré una nova aplicació anomenada "Aplicació 7" amb les funcionalitats 1 (paràmetre PISCINA), 2 (paràmetres CASA i FINESTRA), 3 i 4. A continuació tens la descripció de l'Aplicació 7:

<!--"Descripció aplicació 7 amb funcionalitat 1 (paràmetre PISCINA), amb funcionalitat 2 (paràmetres CASA i FINESTRA), funcionalitat 3 i funcionalitat

4"-->

<Aplicació 7>

<Funcionalitat 1 "paràmetre PISCINA">

<Funcionalitat 2 "paràmetres CASA i FINESTRA">

<Funcionalitat 3>

<Funcionalitat 4>

L'Aplicació 7 ofereix diverses funcionalitats als seus usuaris. La funcionalitat 1 permet als usuaris utilitzar el paràmetre PISCINA per dur a terme una acció específica relacionada amb piscines. Els usuaris poden personalitzar el valor d'aquest paràmetre segons les seves necessitats.

A més, l'aplicació inclou la funcionalitat 2, que utilitza els paràmetres CASA i FINESTRA per realitzar una tasca o acció específica relacionada amb les cases i les finestres. Els usuaris poden proporcionar els valors d'aquests paràmetres per personalitzar l'acció.

A més de les funcionalitats anteriors, l'Aplicació 7 també ofereix la funcionalitat 3 i la funcionalitat 4, que permeten als usuaris realitzar altres tasques o accions específiques proporcionades per l'aplicació.

Aquesta és la descripció de l'Aplicació 7 basada en la informació proporcionada. Si tens alguna altra pregunta o necessites més aclariments, estic aquí per ajudar-te.